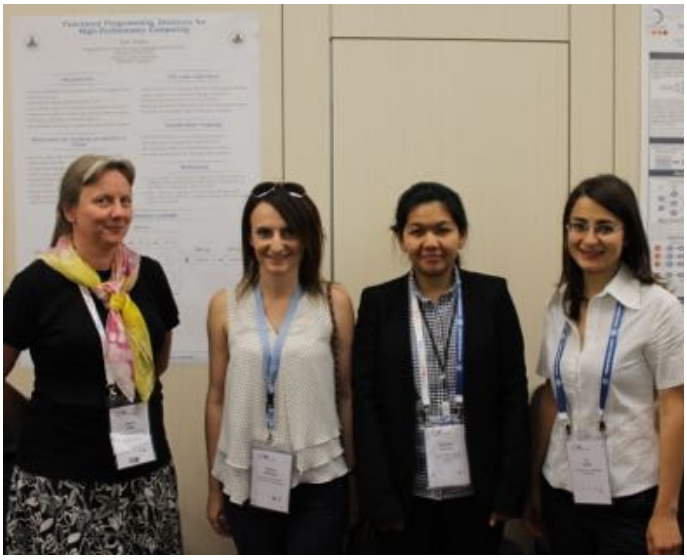
Diversifying the HPC community



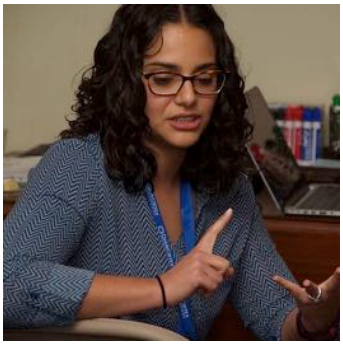
WHPC

WOMEN IN HIGH PERFORMANCE COMPUTING

Raising the profile of women at SC16

WHPC was created with the vision to educate and collaborate with the HPC community and encourage participation by providing knowledge, fellowship, and support to women and the organizations that employ them.

Women in HPC at SC16
Sunday 13 November 2016



welcome

Toni Collis, Director, WHPC

Welcome to the Women in High Performance Computing (WHPC) workshop at SC16. This book is a compilation of posters showcasing the work done by women working day-to-day with HPC. The WHPC workshop at SC16 is the fifth international workshop providing women with a platform to showcase their work, and highlights the contribution women make and how this is essential to the future of the HPC industry.

As we move towards a world where IT and technology dominate more and more of society's activities, with HPC programming methods and supercomputers increasingly used across the globe, it has never been more important to ensure that the HPC workforce represents the entire human population, not just a privileged few. By providing a diverse HPC workforce society is then best placed to represent the needs of everyone, irrespective of their gender, ethnicity, race or religion. This book highlights the importance of women in the world of HPC and their increasing contributions to supercomputing, HPC applications and technology development.

For the first time the SC16 Women in HPC workshop has offered mentoring to all of the poster presenters at this year's workshop. We hope that the experience of this workshop and the relationships that develop between authors and mentors provide valuable experience in the pursuit of an HPC career and encourages all to continue towards a more diverse and inclusive supercomputing community in the future.

Women in High Performance Computing: SC16 Posters

Gladys K Andino Baustista

Engaging women in HPC at Purdue University

We seek to improve the representation of women in HPC at Purdue University by highlighting the efforts of women HPC practitioners. In addition, we intend to describe a new initiative under development at Purdue. We will discuss three main thrusts of the planned effort and what it will accomplish in the near future.

1. Faculty activities to promote diversity in HPC: Dr. Vetria Byrd is the founder and organizer of the biennial Broadening Participation in Visualization (BPViz) Workshop co-funded by CRA-W/CDC and the NSF (Award No. 1419415). Dr. Byrd is developing a summer research experience for undergraduates (REU) to build visualization competencies in undergraduates. Her initiatives will engage undergraduate women and members of underrepresented groups in the field of visualization in HPC.
2. Women from domain sciences and HPC expertise in Research Support: Currently, the research computing team at Purdue includes three female computational scientists. They provide computing expertise to students, staff and faculty using HPC, particularly in bioinformatics applications, big data, learning as well as data visualization.
3. Female student engagements in HPC: Led by Purdue faculty and our female computational scientists, we are currently developing a new initiative to promote HPC for women and minorities. We intend to create a women-in-HPC networking group. The planned activities include regular meetings to discuss HPC-related issues and progress, and to promote and provide funding for women and minorities to attend conferences like Grace Hopper celebration and SC16. These efforts will broaden access to HPC for women and minorities in sciences.

Gladys holds a position as a senior scientific applications analyst in Information Technology at Purdue University. Her role is to provide computing expertise to students, staff and faculty using HPC, particularly those performing research that requires the use of bioinformatics software, analysis, and pipelines. In addition, she provides training and educational workshops about using computing for research campus wide. Her PhD research involved examining gene expression and viral concentration in Varroa mites, one of the principal causes of honey bee hive collapses worldwide. An entomologist by training, her doctoral work required her to learn computational bioinformatics and high-performance computing techniques basically from the ground up. After graduating, she jumped at the opportunity to share what she had learned, and to keep learning more, as a life sciences specialist for ITaP Research Computing.

Engaging Women in HPC at Purdue University

Gladys K. Andino[1*], Boyu Zhang[1], Preston Smith[1] and Vetria L. Byrd[2]
[1]ITaP Research Computing
[2]Computer Graphics Technology, Polytechnic Institute

Women in HPC at Purdue

PURPOSE

We seek to improve the representation of women in HPC at Purdue University by highlighting the efforts of women HPC practitioners. In addition, we intend to describe a new initiative under development at Purdue. We will discuss three main thrusts of the planned effort and what it will accomplish in the near future.

1. Faculty activities to promote diversity in HPC

❖ Dr. *Vetria L. Byrd* is the founder and organizer of the biennial Broadening Participation in Visualization (BPViz) Workshop co-funded by CRA-W/CDC and the NSF (Award No. 1419415).

- Dr. Byrd is developing a summer research experience for undergraduates (REU) program to build visualization competencies in undergraduates

Fig.1 Faculty Advisor of the WHPC at Purdue

- Her initiatives will engage undergraduate women and members of underrepresented groups in the field of visualization in HPC

2. Women from domain sciences and HPC expertise in Research Support

❖ Dr. *Boyu Zhang* a data scientist, provides software and algorithmic support to faculty members in solving Big Data problems.

- She builds, deploys, and maintain big data analytical frameworks
- Migrates a researcher's sound recording analysis workflow from a sequential process to a parallel process on HPC clusters
- Helps with performance testing on Big Data software used in statistics

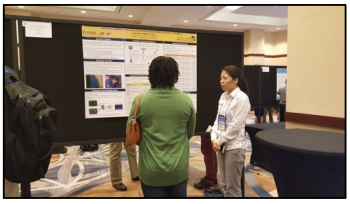


Fig.2 BioSounds as BigData: presented at XSEDE16

2. Women from domain sciences and HPC expertise in Research Support

❖ Dr. *Tsai-wei Wu*, has a background in Electrical Engineering (Optics).

- She supports scientific computation of large system in extreme scale, and three-dimensional data visualization and volume rendering to HPC users at Purdue
- In addition, she was selected as the XSEDE 2016 campus champion fellow, her research will focus on Two-dimensional (2D) semiconductors and their heterostructures

Fig.3 Tsai-wei Selected as XSEDE16 Campus champion

❖ Dr. *Gladys Andino*, provides computing expertise to HPC users, particularly those performing research that requires the use of bioinformatics software, analysis, and pipelines.

- She holds collaborations with some Faculty and research groups at Purdue (Chemogenomics studies, and genomics studies on *Varroa* mites)
- She provides training and educational workshops in topics such as Unix and HPC, campus-wide

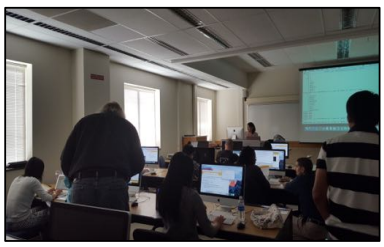


Fig.4 Unix workshops at Purdue

3. Female students, staff and faculty engagements in HPC

❖ To raise awareness of women in technology fields, ITaP Research Computing created a group "WHPC at Purdue".

Goals for this group are:

- To promote awareness of women in HPC in the University.
- Provide a supportive community for all women in HPC.
- Share opportunities and resources for scholarships, awards, and personal development.
- Promote participation in major conferences such as:
  - Grace Hopper Celebration of Women in Computing
  - SuperComputing conference
  - ACM Richard Tapia Celebration of Diversity in Computing

❖ The first WHPC meeting was held October 6, 2016 at Purdue University.

- There were a total of 25 attendees (22 females and 3 males)
  - 5 Faculty
  - 14 Staff
  - 5 Graduate students
  - 1 Undergraduate

Fig.5 WHPC at Purdue - First Callout

- Activities we plan for the future
  - Invited speakers to share science expertise, personal experiences and wisdom on careers in HPC
  - Mentor/protégé relationship building
    - ✓ Purdue Campus
    - ✓ Other Universities
  - Monthly brown bag lunches

Neelofer Banglawala

Bespoke bone modelling with VOX-FE

VOX-FE is a voxel-based finite element software suite developed jointly by the University of Hull's Medical & Biological Engineering group and the Edinburgh Parallel Computing Centre. VOX-FE comprises a front-end GUI and back-end solver, allowing easy visualisation and manipulation of detailed and complex bone models, and the investigation of how such models deform when subject to various forces. As part of an on-going Hull-EPCC collaboration funded by the ARCHER eCSE programme [1], VOX-FE2 now boasts a sophisticated PARAVIEW-based graphical user interface that allows the complex loading regimes present in biomechanical analyses to be readily applied to the model geometry. The resultant 3D stress and strain patterns can also be easily visualized. Furthermore, the VOX-FE solver has been entirely replaced by a new scalable, PETSc-based solver developed on ARCHER. Load balancing functionality has been added to the solver using ParMETIS, a fast scalable graph-partitioning library.

This poster details the solver-side developments that have led to VOX-FE3, including overall improvement in the solver's runtime, scalability, ability to handle larger models (~200 Million elements) and load balancing functionality. VOX-FE3 is available as open-source software.

With a background in theoretical physics (University of Cambridge), Neelofer obtained her PhD at the Institute for Condensed Matter and Complex Systems (University of Edinburgh) looking at stochastic processes in evolutionary biology. She then went on to work as a biophysical modeller at the British Antarctic Survey (Cambridge), where, despite her best attempts, she never made it to Antarctica. Her adventure in HPC began when in 2014 she joined the Edinburgh Parallel Computing Centre (EPCC). She is currently an Applications Developer and is a member of the team that runs ARCHER, the UK National Supercomputing Service.

Bespoke bone modelling with VOX-FE

Neelofer Banglawala[(1),*]
Iain Bethune[(1)], Richard Holbrey[(2)], Michael J. Fagan[(2)]
(1) EPCC, The University of Edinburgh, (2) The University of Hull

Modelling bones: why and how?

Bone is in a continual state of flux through ossification (growth) and resorption (loss). Bone loss begins at age 35, with > 30% bone lost by age 70. Bone has a complex geometry and is made up of different materials e.g. hard outer cortical shell, spongey inner trabeculae and soft marrow.

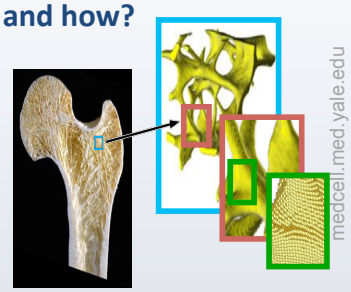


Figure 1. Bone model made up of small 3d finite elements (voxels).

How does bone respond to forces and constraints e.g. an implant? How does bone formation respond to stress and strain? We can explore these questions with *in silico* experiments on bone models. The complex geometry of bone can be modelled as consisting of millions of small cubic elements known as voxels.

VOX-FE : legacy solver

Developed jointly by Hull [1] and EPCC, VOX-FE [2] is a voxel-based finite element software suite (solver & GUI) for the analysis of bone models. The solver outputs the final voxel displacements of a bone model subject to linear forces/ constraints. Written in C++/MPI, earlier versions of VOX-FE could:

- Manipulate/solve bone models of ~20x10^6 elements, max 4 material types
- Parallelise models along 1 dimension
- Scale up to 256 cores on ARCHER [3]

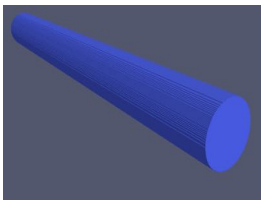


Figure 2. Dense cylinder "bone" model, used for comparing analytical solutions to simulation results.

But realistic bone modelling requires more complex models with several hundred million voxel elements with an arbitrary number of materials (e.g. surrounding tissue). Exploiting HPC resources becomes crucial.

Solver Development

To exploit large-scale HPC resources such as ARCHER, bone models need to be parallelised along all dimensions, and the solver must scale well to thousands of cores. Through two ARCHER Embedded CSE funded development projects, we replaced the legacy solver with an entirely new design that utilises the powerful functionality of the PETSc library [4]. This has the added benefit of expanding the range of solution algorithms available to the solver. We also improved the load balancing capabilities of VOX-FE by using the fast, parallel graph partitioning library ParMETIS [5]. The solver developments are summarised below:

Stage I – solver redesign using PETSc library

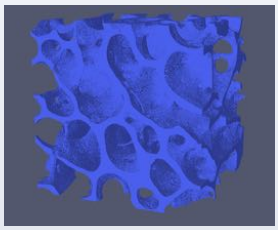
- New solver interface with PETSc functionality
- Redesign representation of bone model to support large, complex models with an arbitrary number of materials
- New and improved interface between solver and GUI

Stage II – load balancing using ParMETIS

- Representing the bone model as a graph of elements, ParMETIS then finds the optimal partition of the bone model across processes

Performance Improvements

The solver development has resulted in the latest release, VOX-FE3, which can manipulate and solve models of hundreds of millions of elements with an arbitrary number of materials. Furthermore, the solver scales well to thousands of cores, as shown in Figure 3.



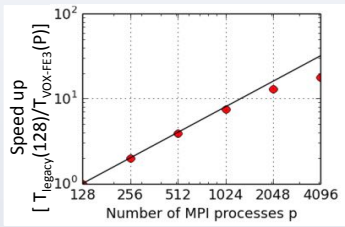
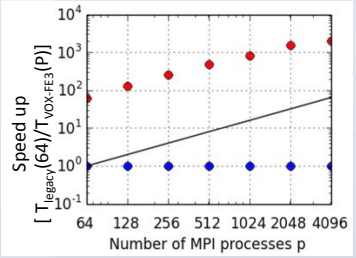


Figure 3. Right: Horse head trabeculae model (200x10^6 elements). Left: Strong scaling speedup of VOX-FE3 solver using horse head model. Solid line is ideal speedup.

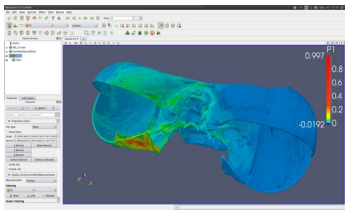


Better load balancing has resulted in less memory usage per process, improving the overall performance of the solver compared with the legacy solver

Figure 4. Strong scaling speedup of model loading times: legacy solver (blue), VOX-FE3 solver (red), ideal speedup (black).

GUI Development

A new ParaView [6] based GUI has replaced the legacy GUI, which was written in Borland C++ and was tied to the Windows OS. The new GUI can manipulate larger models and has additional features, such as being able to account for the effect of muscle tissue on bone.



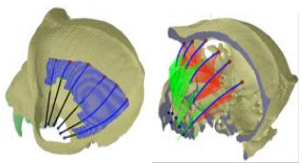


Figure 5. Left: screenshot of new ParaView GUI, visualising a dragonfly head. Right: example muscle wrapping implementation [7].

Further development: bone remodelling

Bone remodelling (growth/loss) can now be investigated by setting a threshold stress above/below which bone is added/lost when subject to differential stress and strain.

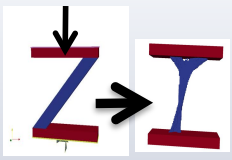


Figure 6. 'Z' bone model deforms into 'T' bone model when compressed from above.

References

[1] Biological & Medical Engineering Group, University of Hull
[2] VOX-FE is open source software: https://sourceforge.net/projects/vox-fe/
[3] ARCHER is the UK's National Supercomputing Service: http://www.archer.ac.uk
[4] PETSc: https://www.mcs.anl.gov/petsc/
[5] ParMETIS, Karypsis lab: http://www.karypsis.ac.uk
[6] ParaView: http://www.paraview.org/
[7] J Liu et al. 2012 Biomechanics and Modeling in Mechanobiology. 11:1 35-47.

* n.banglawala@epcc.ed.ac.uk

Hongmei Chi

Particle Swarm Optimization for High-dimensional Stochastic Problems

Inspired by the social behavior of the bird flocking or fish schooling, the particle swarm optimization (PSO) is a population based stochastic optimization method developed by Eberhart and Kennedy in 1995. In this poster, we investigate the effect of initializing the swarm with scrambled optimal Vander Corput sequence, which is a randomized quasirandom sequence. This ensures that we still have the uniformity properties of quasirandom sequences while preserving the stochastic behavior for particles in the swarm. Numerical experiments are conducted with benchmark objective functions with high dimensions to verify the convergence and effectiveness of the proposed initialization of PSO.

Dr. Hongmei Chi is an Associate Professor of Computer & Information and Sciences at Florida A&M University. She currently teaches graduate and undergraduate courses in cloud computing and datamining and researches in areas of parallel computing and applied security. Dr. Chi has published many articles related to research and education. Her web page is www.cis.famu.edu/~hchi

Particle Swarm Optimization for High-dimensional Stochastic Problems

Hongmei Chi[1] **& Kariyawasam Weerasinghe**[2]

[1]Florida A& M University and [2]Auburn University

hchi@cis.famu.edu — 1 (850) 412 7355

Abstract

Inspired by the social behavior of the bird flocking or fish schooling, the particle swarm optimization (PSO) is a population based stochastic optimization method developed by Eberhart and Kennedy in 1995. In this poster, we investigate the effect of initializing the swarm with scrambled optimal Vander Corput sequence, which is a random-ized quasirandom sequence. This ensures that we still have the uniformity properties of quasirandom sequences while preserving the stochastic behavior for particles in the swarm. Numerical experiments are conducted with benchmark objective functions with high dimensions to verify the convergence and effectiveness of the proposed initialization of PSO

Particle Swarm Optimization

- Each particle is searching for the optimum
- Each particle is moving and hence has a velocity.
- Each particle remembers the position it was in where it had its best result so far (its personal best)
- The particles in the swarm co-operate. A particle knows the fitnesses of those in its neighbourhood, and uses the position of the one with best fitness.

Figure 1: A bird swarm* *https://moenanishimura1.wordpress. com

Optimization Problem

Without loss of generality, consider a minimization problem in the D dimensional space. where $f \in \mathbf{C}$, $\mathbf{C}$ is the set of bounded, continuous functions and $f : \Re^D \to \Re$.

$$\begin{aligned} &Minimize : f(\vec{a}) \\ &Subject\ to : \\ &g_k(\vec{a}) \le 0 \qquad k = 1, 2, ...p \\ &h_m(\vec{a}) = 0 \qquad m = 1, 2, ...q \\ &a^j_{min} \le a^j \le a^j_{max} \qquad j = 1, 2, ...D \end{aligned} \tag{1}$$

where $\vec{a} = (a^1, a^2, ..., a^D)$ and p, q are number of inequality and equality constraints respectively.

Algorithm

$$\vec{v}_{t+1} = w\vec{v}_t + c_1 r_{1,t+1}(p\vec{best}_t - \vec{a}_t) + c_2 r_{2,t+1}(g\vec{best}_t - \vec{a}_t) \tag{2}$$

$$\vec{a}_{t+1} = \vec{a}_t + \vec{v}_{t+1} \tag{3}$$

- v is the particle velocity
- a is the particle position
- $pbest$ and $gbest$ are particle best position and global best position achieved so far
- c_1 factor is a constant called the cognitive(or personal or local) weight and c_2 factor is a constant called the social or global weight.It is assumed that c_1= c_2 = 1.49445
- w factor is called the inertia weight and is simply a constant which is equal to 0.73.
- r_1 and r_2 are random numbers between (0,1)

Convergence Analysis of GPSO for D dimensions

Proposition: Let $(\vec{X_n})$ be a sequence of random vectors defined on a sample space Ω, such that their components are square integrable random variables. Denote by (X^i_n) the sequence of random variables obtained by taking the i-th component of each random vector $(\vec{X_n})$. The sequence $(\vec{X_n})$ converges in mean-square to the random vector $\vec{X}$ if and only if X^i_n converges in mean-square to the random variable X^i (the i-th component of $\vec{X}$) for each $i = 1, 2, ..., k$.

- It has proved that for each $j = 1, 2, ..., D$, $(a^{k,j}_t)$ converges to $gbest^j$. so by above proposition it can be conclude that , $\vec{a^k}_t$ converges in mean-square to $g\vec{best}$ for each particle k, $k = 1, 2, ...M$.

Convergence to the Optimum

Let $f \in \mathbf{C}$ be the function to be minimized, and let $\vec{a^*} \in \Omega = \{searchspace\}$ be s.t. $f(\vec{a^*}) = min\{f(\vec{a}); \vec{a} \in \Omega\}$

- The distance between two adjacent particles in the swarm at the initial stage is either $\frac{1}{b^m}$ or $\frac{1}{b^{m+1}}$, where $m = \lfloor \log_b M \rfloor$, M= no.of particles.
- Let $a^j_0, b^j_0 \in \Omega$ be two adjacent particles at the initial stage, then we have, $|a^j_0 - b^j_0| \le \frac{1}{b^m}$.
- If $a^* = a^k_t$ for some k then $g\vec{best} = a^*$ .Then f achieves its minimum value.
- If $a^* \ne a^k_t$ for any k, then $|a^* - g\vec{best}| \le \frac{1}{2b^m}$ and Since f is continuous, $|f(a^*) - f(g\vec{best})| < \epsilon$. In this case f approaches its minimum value within some error of ϵ.

Numerical Results-Bohachevsky function

- $f_1 = a_1^2 + 2a_2^2 - 0.3cos(3\pi a_1) - 0.4cos(4\pi a_2) + 0.7$.
- It is a bowl shaped function.
- Global minimum: $f(a^*) = 0$ at $a^* = (0, 0)$.

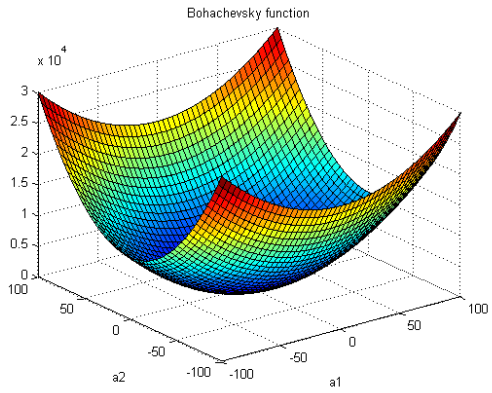
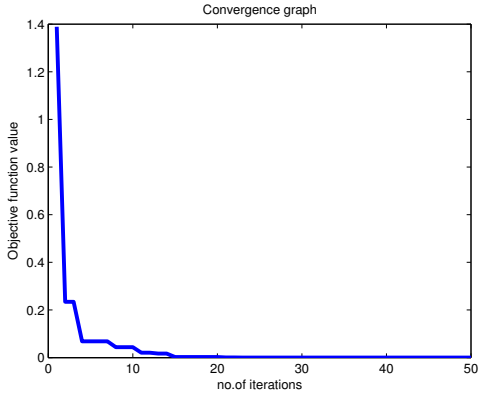


Figure 2: Bohachevsky function

Convergence of Objective function



Sharda Dixit

Automated Empirical Tuning of Performance and Power Consumption using region (CPU, Memory, I/O) driven DVFS for HPC Scientific Workloads

Power aware computing has now become a necessity in high performance computing (HPC) environment. The rapid increase in processing requirements leads to an increase in the number of processor cores, which in turn increases energy consumption substantially. The large amount of heat generated by such a system also increases the maintenance and operational cost. Hence, new solutions are needed for improving energy efficiency while maintaining the performance. This poster presents a method for saving energy in the execution of scientific workload (in HPC systems) by fine tuning the power and performance using Dynamic Voltage and Frequency Scaling (DVFS) technique.

The execution of each workload can be categorized on the basis of the number of CPU cycles and the I/O cycles consumed. Based on this categorization, decomposed (CPU-bound and non-CPU bound) statistics are collected and an empirical input format is constructed. This empirical format is passed to the DVFS module that generates a set of DVFS values (one for each sample). These DVFS settings are applied to the system and the output is compared with default settings to determine maximum possible power savings within the tolerance of performance degradation (specific to each application).

Reducing power introduces performance degradation, so a trade-off is made between power and performance to meet the requirement of performance with minimal cost of power. This model, applied on CPU bound and Non-CPU bound (Memory bound and I/O bound) workloads, significantly reduces power consumption with controlled degradation of performance.

Sharda Dixit is associated with Centre for Development of Advanced Computing (C-DAC) since last 10 years and leads the Power Optimization activity for High Performance Computing Systems. Her current research interests include power & performance optimization, profiling of scientific workloads and power aware software framework for future extreme scale HPC systems. She also specialises in Cryptography and Network Security. She received her Master's degree in Computer Applications in 2004 from National Institute of Technology (NIT), India.

Automated Empirical Tuning of Performance and Power Consumption using region (CPU, Memory, I/O) driven DVFS for HPC Scientific Workloads

Sharda Dixit, Anil Gupta, Amarjeet Sharma, Wasundhara Paliwal, Soham Ghosh, Sweta Anmulwar
Centre for Development of Advanced Computing (C-DAC), India
{sharda, anilg, amarjeets, wasundhara, sohamg, asweta} @cdac.in

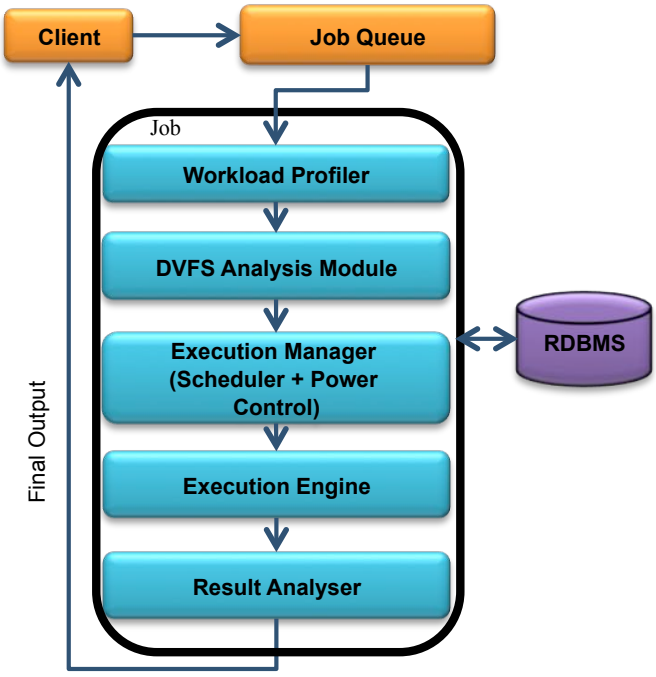
OVERVIEW

- A framework for optimizing power consumption on HPC scientific workloads
- Fine-tuning of operational voltage/frequency values with performance threshold
- Region (CPU, Memory, I/O) driven workload decomposition

MOTIVATION

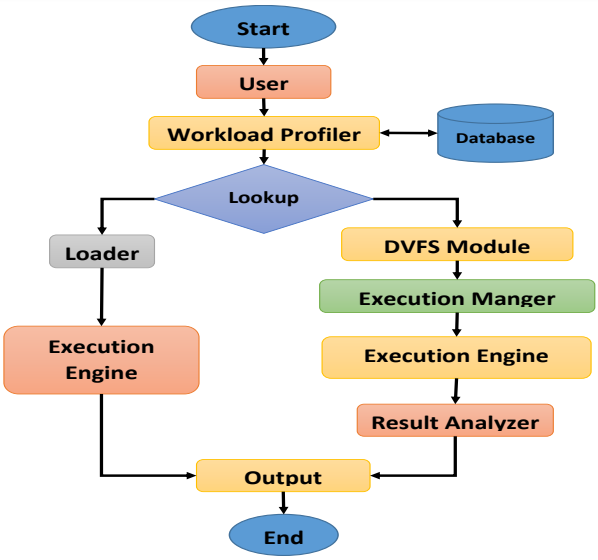
- Performance scaling is increasingly dominated by the power constraints
- Trade-offs to be made between performance and power/ energy
- These trade-offs are driven by application workloads
- Execution of workloads need to be power aware

PROPOSED FRAMEWORK



- **Job Queue:** This module schedules the user's jobs in a queue.
- **Workload Profiler:** This collects necessary and sufficient inputs for DVFS Decision Modules.
- **DVFS (Dynamic Voltage & Frequency Scaling) Analysis Module:** It generates optimal operational voltage and frequency values for workload.
- **Execution Manager:** It applies the operational parameters to execution unit and ensures that appropriate settings are configured before execution of the process/sub-process.
- **Execution Engine:** It executes the scheduled jobs with given DVFS parameters and obtain multiple results.
- **Result Analyzer:** It selects the optimal result for a given job among multiple set of results collected.

APPROACH



- The application workload is decomposed into CPU intensive and non-CPU (Memory and I/O) intensive regions
- Based on the decomposed statistics collected, an empirical input format is constructed
- A set of DVFS parameters is generated for each input sample
- The set of DVFS parameters are applied during the execution of the workload
- The result values along with the output with default DVFS settings is then compared to get the degree of power saving
- The optimal voltage and frequency values are stored for future reference for the execution of the same workload

CONCLUSIONS

- The proposed solution provides an optimal execution environment for scientific workloads on HPC clusters
- It generates multiple sets of results by fine-tuning DVFS parameters for each process/sub-process and based on their performance threshold, select the optimal one
- It gives finer control of operational frequency and voltage during the execution of workload leading to significant power savings

REFERENCES

[1] *Ryan Elmore et al., "An Analysis of Application Power and Schedule Composition in a High Performance Computing Environment", Technical Report, NREL/TP-2C00-65392, January 2011.*

[2] *Hayk Shoukourian et al., "Power Variation Aware Configuration Adviser for Scalable HPC Schedulers", 2015 International Conference on High-Performance Computing & Simulation (HPCS), 20-24 July 2015.*

[3] *Torsten Wilde et al., "Taking Advantage of Node Power Variation in Homogenous HPC Systems to Save Energy", ISC High Performance 2015, LNCS 9137, pp. 376–393, 2015.*

Lydia Duncan

Array Initialization Improvements in Chapel

Chapel is a programming language developed at Cray Inc. to improve the productivity of parallel computing. Combining the low-level control over a globally partitioned address space with the ability to abstract common iteration patterns into a reusable form, among other features, Chapel has a lot to offer HPC. However, there are still many improvements to be made before it can truly be considered production-ready.

For instance, Chapel's variables are all initialized at declaration time. This is fine for primitive types but in the case of large data structures, such as complex arrays, one can imagine wanting more fine-grained control. If the programmer knew the array's contents would be overwritten before the first read, they might want to avoid paying the cost of initialization. Additionally, while Chapel's arrays are initialized in parallel today, the implementation is less than ideal – the type is defined through a combination of library code and special case compiler hooks, making it more difficult to maintain and optimize. My work as part of the initializers subgroup has been on a more thorough design for constructors, one that aids the array implementation naturally.

Chapel's arrays should be of particular interest to the HPC community – the distribution libraries provide common data layouts across multiple nodes to optimize communication, and can be used as a template for writing alternative arrangements. Real application needs would further motivate our improvements, permitting Chapel to grow into its potential as a language and allow programmers to thrive at exascale.

Lydia has been a Software Engineer at Cray Inc. on the Chapel programming language for the past three years, after completing a summer internship with the team. She graduated as Salutatorian from Chief Sealth International High School in 2009, and completed her undergraduate degree in Computer Science cum laude from the University of Washington in June 2013. She participated in the Argonne Training Program for Extreme Scale Computing during the summer of 2014, and co-presented Chapel's half-day tutorial at SC last year with her teammate Michael Ferguson. She also gave a short tech talk at Women Techmakers in Seattle this past January on Chapel as a language, and the Chapel developers' conference (CHIUW) in June on her modifications to Chapel's import statement.

Array Initialization Improvements in Chapel

Lydia Duncan

```
var dom = {0..100, 0..100};
// The indices for an array

var A: [dom] int;
// Declaring an array of ints
```

Problem: Chapel variables initialized by default.

Solution: Use keyword “noinit” to allocate space but not initialize contents.

Problem: noinit originally left every part of a type uninitialized. However, arrays wanted some of their underlying structure initialized, no matter if the contents stored were initialized or given a value later.

```
var A: [dom] int = noinit;
// Declaring an array of ints
```

Solution: allow type designers to define what noinit means on their type through special support in their initializers. Then arrays can declare which parts must be initialized and which can be left alone!

Problem: distributing arrays and using them in parallel is clunky, error-prone and heavy weight.

Solution: Chapel arrays can be declared distributed as easily as using a different set of indices. Parallel operations are easy with promotion!

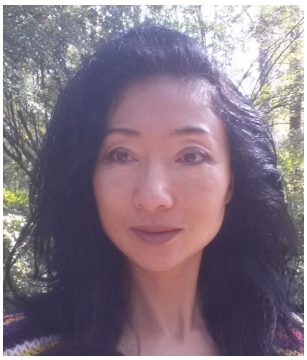
```
var dist = dom dmapped
        Block (boundingBox=dom);
// Taking advantage of multi-node
// features

var B: [dist] int = noinit;
// Declaring a distributed array
```

For more information on Chapel, its initializer story, and its multi node arrays, visit http://chapel.cray.com

Wei P Feinstein

Accelerating protein functional annotation with Intel Xeon Phi coprocessors



Drug development is routinely streamlined using computational approaches to improve hit identification and lead selection, to enhance bioavailability, and to reduce toxicity. As a mounting body of genomic knowledge has been accumulated in the past decade, great opportunities arise for pharmaceutical research. However, because processing this large volume of data demands unprecedented computing resources, the incorporation of HPC systems into drug discovery and development becomes challenging.

In this study, we describe the development and benchmarking of a parallel version of eFindSite, a structural bioinformatics algorithm for the identification of drug-binding sites in proteins and molecular fingerprint-based virtual screening. Parallelizing the structure alignment calculations using pragma-based OpenMP enables the desired performance improvements, scaling well with the number of computing cores. With minimal modifications, a complex, hybrid C++/Fortran77 code was successfully ported to a heterogeneous architecture, yielding significant speedups.

Compared to a serial version, the parallel code runs 11.8 times faster on the processor and 10.1 faster with Intel® Xeon Phi coprocessors. When both resources are utilized simultaneously, the speedup is 17.6; essentially, only 2.1 hours instead of 36.8 hours is needed to identify ligand-binding sites for 501 target proteins. In the near future, we plan to further parallelize the code for structure alignment using more than one thread to process each protein. However, since many small loops are used in this part of the code, achieving noticeably better performance may be difficult. eFindSite is an open source tool and accessible either by web service or standalone at http://brylinski.cct.lsu.edu.

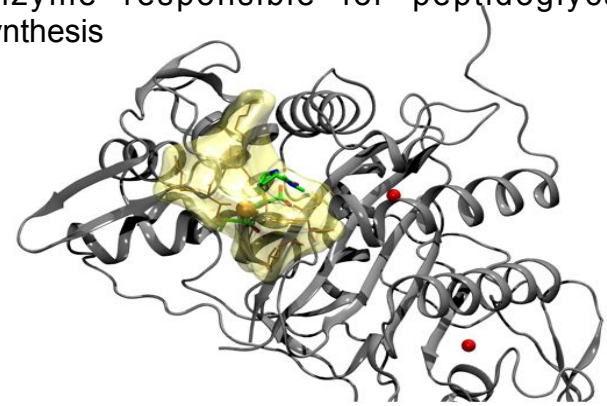
I currently work in the HPC user services division of Louisianan State University. I am also a computational biologist interested in developing algorithms for applications in drug discovery. Before I joined the current group about 18 months ago, I worked as a postdoc in the field of computational biology at LSU. In addition, I co-led the TESC (Technologies for Extreme Scale Computing) team, where a group of interdisciplinary graduate students at LSU collaborated to improve the performance of various domain science projects using hardware accelerators. Based on my exposure to HPC and subsequent understanding of the pivotal role of HPC in my own research as well my struggle to make things work in the HPC world, I made a career change decision. From a computational biologist using HPC, I am now on the other side of the isle of HPC to assist researchers in maximizing the utilization of HPC resources in their scientific endeavors.

Accelerating protein functional annotation with Intel Xeon Phi coprocessors

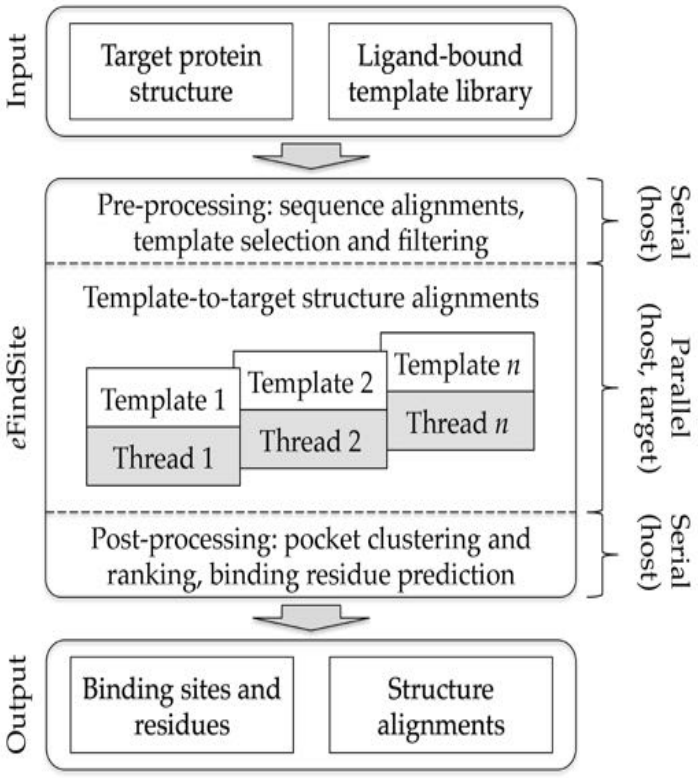
Wei P. Feinstein and Michal Brylinski
Louisiana State University

What is *e*FindSite

*e*FindSite is a software package to accurately identify ligand-binding sites and binding residues across large datasets of protein targets using weakly homologous templates. The image below illustrates drug docking sites of a target protein (Pseudomonas aeruginosa, PBP3 (PDB-ID: 3pbr chain A), a critical enzyme responsible for peptidoglycan synthesis

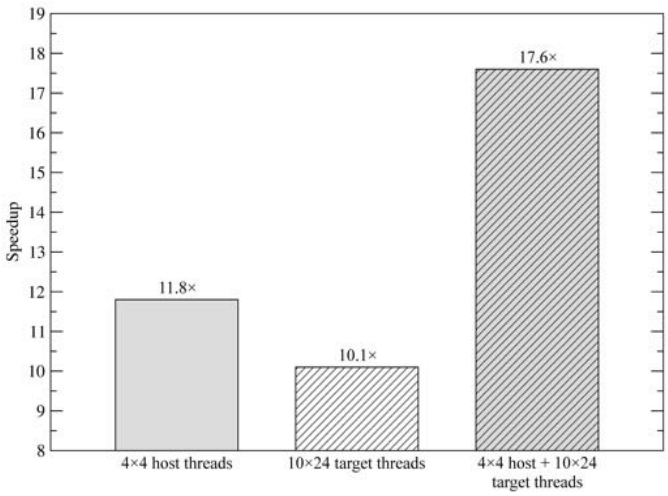


Workflow of *e*FindSite

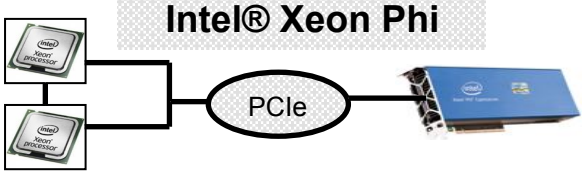


Performance Comparison

Data Sets	50-100 templates	100-150 templates	150-200 templates	200-250 templates
200-300 residues	50	50	50	50
300-400 residues	50	50	50	50
400-500 residues	34	32	23	12



Intel® Xeon Phi



Conclusions and Future Study

- Easy porting to MIC and short development time
- Satisfactory speedups
- Spaghetti codes OK
- Further improvements require low level optimization
- Xeon Phi offers an inexpensive extension to CPU nodes to provide additional computing power

Rosa Filgueira

dispel4py - A Python toolkit for enabling the automatic portability of scientific applications among HPC architectures

Scientific communities have access to a large variety of computing resources. For testing scientific applications local resources are usually selected, however for larger runs, distributed resources is the most frequent choice. Successful use of these technologies requires a lot of additional machinery whose use is not straightforward for non-experts, since different parallel frameworks should be used depending on the type of memory where applications are run. This implies, that for achieving the best applications' performance, users have to change their codes depending on the features of HPC architecture selected. Therefore, new advanced software libraries, tools and interfaces are needed to empower scientists to invent and improve their methods and models, which allow them to work more effectively as they create, refine and use their scientific methods in a scalable way.

This works presents a new Python toolkit for scientists, called dispel4py [1] to provide them automatic and transparent portability of their data intensive and high-performance applications among HPC-architectures. It provides an enactment engine that automatically maps and deploys abstract workflows onto multiple parallel platforms, including Apache Storm, MPI, Multiprocessing, as well as a Sequential mapping for development and small applications. Recently, the dispel4py team and the Pegasus [2] team have collaborated to develop an integrated and complete approach, called Asterism [3] for running scientific applications across multiple heterogeneous systems without users having to manage the data distribution across systems; co-place and schedule their methods with computing resources; and store and transfer large/small volumes of data.

dispel4py has been evaluated [4] using the *seismic ambient noise cross-correlation* application, a common data-intensive analysis used by many seismologists. We used 1000 seismic stations as input data (150 MB) performing 499500 cross-correlations as output data (39GB).

1. dispel4py source code: https://github.com/dispel4py/dispel4py
2. Ewa Deelman, Karan Vahi, Mats Rynge, etc. Pegasus in the Cloud: Science Automation through Workflow Technologies. IEEE Internet Computing, volume 20, pp.: 70-76, 2016.
3. Asterism source code: https://github.com/dispel4py/pegasus_dispel4py
4. Rosa Filgueira, Amrey Krause, Malcolm Atkinson, Iraklis Klampanos, Alexander Moreno. dispel4py: A Python Framework for Data-Intensive Scientific Computing. International Journal High Performance Applications (IJHPCA), volume 8, pp.: 165–414, 2016

Rosa Filgueira has recently joined the British Geological Survey (BGS) as a Senior Data Scientist to work with a variety of H2020 projects. From October 2011 to October 2016, she worked as a Senior Research Associate at the Data Intensive Research (DIR) Group of the University Edinburgh. Previously, she was working as a Research and Teaching Assistant at the Computer Architecture Group of University Carlos III Madrid. Between these two positions, she had spent an extended research visit as a research visitor in the Data Intensive Research group in Edinburgh funded by two HPC-Europa2 grants. Recently, she performed a research visit at University of Southern California (USC), in the Information Sciences Institute (ISI), funded by the PECE SISCA award. Her research expertise is on improving the HPC applications' scalability and performance and on exploring how to make data-intensive methods more accessible, having contributed to several European (e.g. VERCE and ENVRIplus projects) and national (e.g. EFFORT) projects in hazard forecasting and parallel processing.

Personal website: http://www.rosafilgueira.com/

dispel4py: A Python toolkit to enable automatic portability among HPC architectures

Rosa Filgueira (1), Rafael Ferreira da Silva (2), Amrey Krause (3), Ewa Deelman (2) and Malcolm Atkinson (4)

(1)) British Geological Survey –Lyell Centre, UK (2) University of Southern California – ISI, US (3) University of Edinburgh – EPCC, UK (4) University of Edinburgh –DiR, UK

Automation

Automates pipeline executions
Stream-based model
Portability among HPC resources

Workflow Composition

Python toolkit
Easy-to-use
Groupings (all-to-all, all-to-one, one-to-all)

Automatic mappings

Multiprocessing (shared memory)
Distributed memory (MPI)
Distributed Real-time (Apache Storm)

Optimization

Automatic parallelization
Multiple streams (in/out)
Avoids I/O

Processing Elements (PEs)

PE is the **basic** computational unit
PEs are **connected** by streams
Python for describing PEs & connections
PEs consume/produce **any number and types** of streams
PEs run **concurrently** and in **homogeneous** resources
Graph → workflow topology → defined by users
Instance → PE executable copy running in a process
→ PE translated (by dispel4py) to one or more instances
Grouping → communication pattern among instances

dispel4py [1,2] workflows can be developed on **local machines**. And run them at **scale** on wide range of **HPC resources without** need to **adapt** them

Case study: Seismic ambient noise cross-correlation

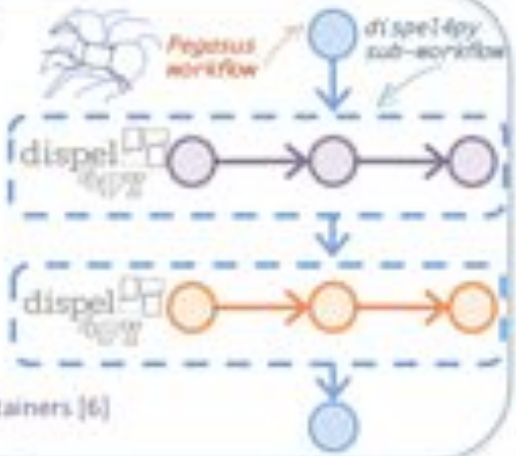
Preprocesses (Phase1) and cross-correlates (Phase2) traces from seismic stations using IRIS database

VERCE project [3] - workflows evaluated using multiple **HPC** resources and **mappings**

Asterism: hybrid workflows

dispel4py and **Pegasus** [4] teams collaborated to develop **Asterism** [5], for running data-intensive codes across **heterogeneous** resources

- Automatic parallelization & stream based executions: **dispel4py**
- Data movement & coordination: **Pegasus**
- Execution environment: Docker Containers [6]



Testing Asterism – Seismic ambient noise cross-correlation

Conclusions

dispel4py: Python library for streaming and data-intensive processing

- Users express their computational activities
- Same workflow executed in several parallel systems
- Easy to use, open, and encourage sharing the methods & applications

Asterism: Python framework to run data-intensive applications across multiple heterogeneous resources

References and Online material

[1] Rosa Filgueira, *et all. dispel4py: A Python Framework for Data-Intensive Scientific Computing*. IJHPCA, 2016
[2] dispel4py code: https://github.com/dispel4py/dispel4py
[3] Malcolm Atkinson, *et all. VERCE delivers a productive e-Science environment for seismology research*. 11th IEEE eScience, 2015
[4] Ewa Deelman, *et all. Pegasus in the Cloud: Science Automation through Workflow Technologies*. IEEE Internet Computing, 2016
[5] Asterism code: https://github.com/dispel4py/pegasus_dispel4py
[6] Research Object: https://seek.rightfield.org.uk/investigation/2

Meghan Fisher

Simulating Volcanic Eruptions on Early Mars

Current 1D models for explosive volcanic plumes on early Mars are based on empirical terrestrial models shown to overestimate the maximum plume height by tens of kilometers. Previous studies have emphasized the need for a physics based simulator to constrain the height of early Martian eruption plumes. We present a 3D Navier-Stokes simulation for early Mars that allows simulation of volcanic eruptions for early Mars atmospheric compositions, particularly the turbulent processes not accounted for in past empirical Martian plume models. The model is based on the terrestrial Navier-Stokes MPI-based simulator Active Tracer High-resolution Atmospheric Model (ATHAM), replacing terrestrial planetary and atmospheric conditions without altering the underlying physics of the model. Since the actual atmospheric conditions of early Mars are not definitively known, we used various proposed atmospheric compositions and surface pressures to simulate possible atmospheric profiles. Unlike 1D empirical models, the Martian ATHAM (M-ATHAM) simulations require HPC resources. Simulations were run on the Idaho National Laboratory's supercomputer Falcon; individual simulations were run on 81 processors and took between one and two days to complete. Without distributed computing, it would be impossible to run M-ATHAM at a fine enough spatial resolution to observe multiple turbulent eddy scales and prevent overestimation or underestimation of plume height. The 3D simulations produced by M-ATHAM also allow eruption visualization, which was not possible with the current models. We found that overall plume height varied significantly with atmospheric composition and that current empirical models overestimate maximum plume heights slightly less than previously thought.

I am a fifth year geoscience Ph.D. candidate at Idaho State University, where I study the physics of explosive volcanic plumes using experimental analogues and numerical simulations. I am developing a Navier-Stokes simulator for Martian volcanic plumes. As an Idaho Space Grant Consortium Fellow this fall, I am simulating volcanic plume collapse on early Mars to determine what conditions lead to collapse and how far the resulting pyroclastic flows would travel. Previously, I have developed remote sensing techniques to determine the velocity fields and density distributions of volcanic plumes using webcam video. I took my first parallel computing course my first year in graduate school and later attended the International HPC Summer School on HPC Challenges in Computational Sciences. When I graduate, I plan to continue to study computational geosciences in either government labs or private industry.

Simulating Volcanic Eruptions on Early Mars

Meghan A. Fisher (fishmegh@isu.edu)[1],
Shannon Kobs Nawotniak (kobsshan@isu.edu)[1]
Department of Geoscience, Idaho State University, Pocatello, ID

1. A 3D Navier Stokes model is necessary to resolve issues with integral plume models

Integral plume rise models for early Mars predict the maximum plume height of ~63 km (Glaze and Bologa, 2002). However, these models do not account for turbulent mixing and estimate entrainment using the Morton et al. (1956) entrainment hypothesis, which can result in discrepancies between actual plume height and simulated plumes. Glaze and Bologa (2002) concluded that to more accurately determine maximum plume height, a Navier Stokes model is necessary. Plume height directly affects deposit sizes. Since fall deposits can appear similar to fluvial deposits on satellite imagery, it is vital to constrain the size of the deposits when searching for water on Mars.

2. M-ATHAM is an add-on module for ATHAM with some modifications to make it Mars compliant.

Active Tracer High-resolution Atmospheric Model (ATHAM; Oberhuber et al., 1998) is a terrestrial 3D Navier-Stokes model that utilizes MPI distributed commuting. M-ATHAM simulates the temperature and pressure profiles and replaces terrestrial coefficients with early Martian values. We made modifications to the microphysics module, which describes how heat transfer from the water in the eruption effects the plume, so that thermal conductivity and diffusion coefficients are consistent with Martian formulations.

3. Using distributed computing, M-ATHAM produces both positive and negative plumes

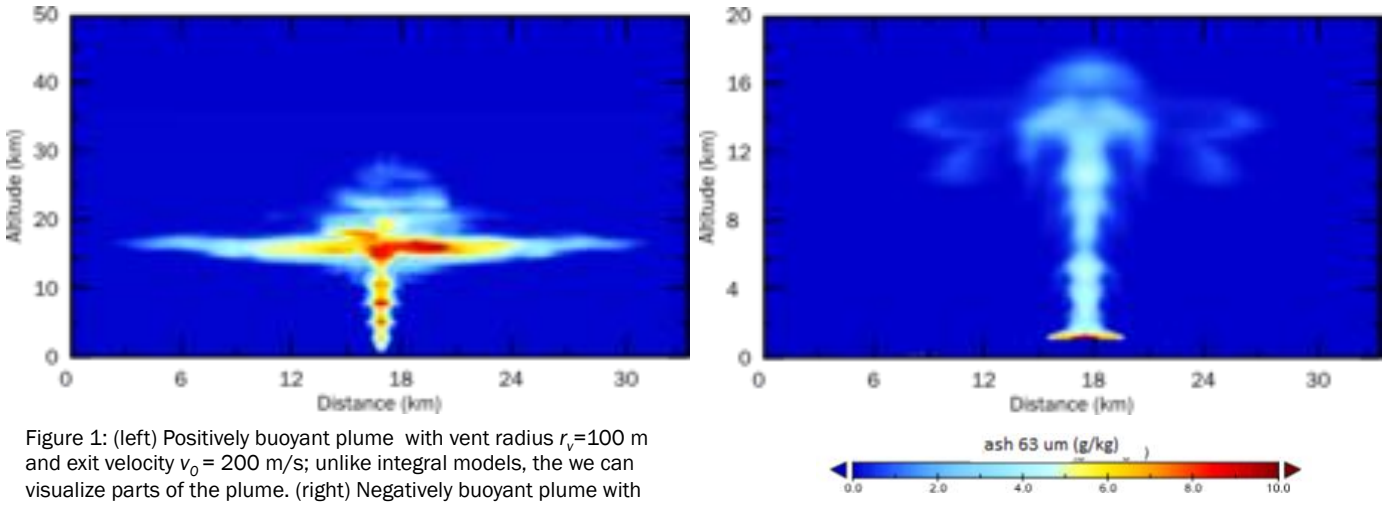


Figure 1: (left) Positively buoyant plume with vent radius r_v=100 m and exit velocity v_0 = 200 m/s; unlike integral models, the we can visualize parts of the plume. (right) Negatively buoyant plume with r_v=100 m and v_0 = 800 m/s. The plumes collapse d since the plume discharge rate is too big to entrain enough air to achieve positive buoyancy. Both plumes were erupted into 100% CO2 atmosphere.

M-ATHAM's tallest plume is 10% taller than the tallest plumes produced by valid integral models. These plumes are approximately 35% taller than the maximum terrestrial plume height. Volcanic plumes on early Mars rise higher than terrestrial plumes with the same mass discharge rates. However, early Martian plumes collapse at lower mass discharge rates than terrestrial eruptions.

M-ATHAM provides a method to visualize plumes in order to study the spatial-temporal movement of rise and collapse.

4. Future Work

All simulated plumes were erupted into atmospheres with no ambient wind. Wind can act as a stabilizing force, resulting in plumes rising buoyantly instead of collapsing. We are simulating eruptions with ambient wind to better understand under what conditions plumes collapse.

Acknowledgements

Computational time was provided by Idaho National Laboratory. Special thanks to Suniti Karunatillake at LSU for advice with model development and Diana Boyack of the ISU Digital Mapping Lab for technical support. Model development is funded by a NASA's Idaho Space Grant Consortium Research Initiation Grant.

Glaze, L. S., & Baloga, S. M. (2002). Volcanic plume heights on Mars: Limits of validity for convective models. *Journal of Geophysical Research: Planets*, *107*(E10).
Morton, B. R., Taylor, G., & Turner, J. S. (1956, January). Turbulent gravitational convection from maintained and instantaneous sources. In*Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* (Vol. 234, No. 1196, pp. 1-23). The Royal Society.
Oberhuber, J. M., Herzog, M., Graf, H. F., & Schwanke, K. (1998). Volcanic plume simulation on large scales. *Journal of Volcanology and Geothermal Research*, *87*(1), 29-53.

Maria Juliana Garzón Vargas

High performance embedded computing platform for emergency vehicle transportation

The effort private health care institutions and government make to cover Emergency Medical Services in most cases is acceptable but constantly changing due to quality, a never-ending process that demands continuous improvement updates. The goal of the EMS system is to minimize further out-of-hospital complications, stabilize and transport patients on a fraction of seconds. However, the overall image that emerges from real life in developing countries is negative, as it has some serious restrictions: requires real time results, accurate technology, and both directly depends on the effective use of appropriate equipment and supplies set up by standards and protocols. Over the past years mobile embedded systems have emerged in this field as an integral solution showing major improvement on specific and advanced needs. The aim of this study is to make an improvement on the quality of patient's transport on computational aspects, to achieve this we are going to identify and define the software architecture that seizes the hardware capability of an embedded system with added value, a very specific need only high performance embedded computing has solved to these days. Accordingly, the outcome we will get from this research is a high-performance, low-energy computing and more importantly a low-cost reliable extensible platform, where future work challenges are a wide variety of services including noise reduction on electromagnetic signals, GPS localization of the vehicles, finding the closest medical center to the point where the emergency took place, and visual transmission of radiologic images.

Maria Garzón, 24, is a Senior Computer Engineering Student at the Industrial University of Santander. Her current research focuses on proposing the design of a tailored software architecture that allows the development of real time medical emergency services applications on a mobile embedded HPC platform. She's been working along with the UIS High Performance and Scientific Computing Center (SC3-UIS) for a year. In her spare time she enjoys playing soccer, spending time with her family and friends and making them laugh.

High performance embedded computing platform for emergency vehicle transportation

Maria Juliana Garzón Vargas, Gabriel Pedraza Ferreira, Carlos J. Barrios Hernández,
High Performance and Scientific Computing Center
Universidad Industrial de Santander, Bucaramanga, Colombia.

Objectives

- Identify the IT needs for emergency vehicles
- Propose an extensible software architecture
- Develop the platform's functional prototype

Background and Motivation

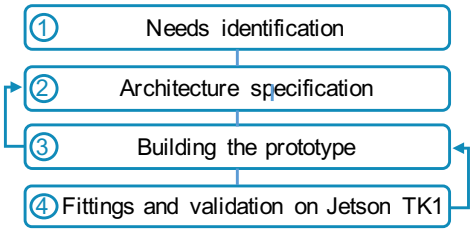
Basic life support in terms of Emergency Medical Services is the action of taking patients from the place where the event occurred to the closest medical center. This directly involves pre-hospital healthcare, which at the same time relies on a set of IT technologies that aim to improve patient care, being this the very first contact of patients with the health care system.

However, the technological support previously mentioned suggests a high cost for all EMS' entities, not only with the purchase of it but the continuing technical assistance this devices require, which creates a gap in the service as it is not available worldwide, as a consequence, a low-cost alternative solution is required.

All in all, it's appropriate to propose that this kind of support can be justified by the assessment and diagnosis of the patient's state along the way with the transmission of pertinent information to the medical center. Through a platform that improves existing features with optimal use of resources this goal will be achieved.

Methods

The software development methodology implemented to introduce the platform is an adaptation of the evolutive prototyping methodology. This research is currently working in phase fourth of the model.[1]



EMS worldwide

To understand the first phase and how the global emergency medical service systems work nowadays, we focus on three targets:

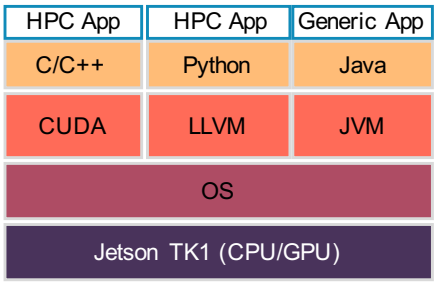
- EU: CEN 1789:2007 [2]
- USA: KKK-A-1822F[3], NFPA 1917 [4]
- COL: NTC 3729–2007 [5]

Hardware Platform NVIDIA Jetson TK1

We have chosen the embedded hardware architecture NVIDIA Jetson Tegra Kepler 1, since it provides the right performance features and requirements for this matter; it's positioned as the first supercomputer for embedded systems and is chosen for the development of high-quality visual applications against a considerable level of reliability. [6]

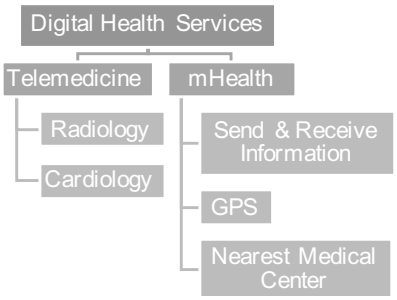
Software Architecture

So far, the outcome is the model of a software architecture composed of three layers: **1. Hardware Layer**, according to the Jetson TK1 hardware **2. Software Layer**, composed of Middlewares (Low Level Virtual Machine and Java Virtual Machine), Operating System (Ubuntu 14.04) and Device Drivers **3. Application Layer**, where the main goal is running three medicine applications: two HPC apps and one generic, within its respectively libraries (i.e. for CUDA: cuFFT, Arrayfire, OpenCV). [7]



Further work

Applications developed on the last layer of the platform from telemedicine to additional services as GPS localization and transmission of relevant information are thought to be the future work for this research.



References

[1] Technische Universität München, Department of Informatics, Robotics and Embedded Systems. Website: www6.in.tum.de/Main/ResearchFtos
[2] AENOR. Website: www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0045721#.V_aaR4_hDct
[3] U.S. General Services Administration, Federal Specification for the Star-of-Life Ambulance. Website: www.ok.gov/health2/documents/KKK-A-1822F%20%2007.01.2007.pdf
[4] National Fire Protection Association, Proposed Draft of NFPA 1917, Standard of Automotive Ambulances 2013 Edition. Website: www.nfpa.org/Assets/files/AboutTheCodes/1917/NFPA1917Draft.pdf
[5] NTC 3729-2007. Website: repository.unac.edu.co/jspui/bitstream/11254/160/20/Norma%20t%C3%A9cnica%20Colombiana%203729%20ambulancias
[6] NVIDIA Jetson TK1. Embedded Systems (2014). Website: www.nvidia.com/object/jetson-tk1-embedded-dev-kit.html
[7] Noergaard, T. (2005), Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers, Elsevier Editorial.

Universidad Industrial de Santander — CONSTRUIMOS FUTURO
Super Computación y Cálculo Científico UIS
NVIDIA GPU RESEARCH CENTER
NVIDIA GPU EDUCATION CENTER

Patricia Grubel

Performance Characterization of HPX: A Task-based Runtime System on the Xeon PhiTM Knights Landing (KNL)

One class of models aimed towards exascale computation is the task-based parallel computational model. Task-based execution models and their implementations aim to support parallelism through massive multi-threading where an application is split into numerous tasks that execute concurrently. In task-based systems, scheduling tasks onto resources can incur large overheads that vary with the underlying hardware. In this work, the goal is to measure performance and overheads incurred when running parallel benchmarks using the HPX parallel runtime system on the newest many core processor from Intel® , the Xeon Phi™ Knights Landing (KNL). HPX is a runtime system that employs asynchronous fine-grained tasks and asynchronous communication for parallel and distributed applications. The performance studies give an understanding of task scheduling overheads and inflation of task duration due to parallelization on the underlying hardware. The knowledge gained can be applied to understanding loss of efficiency in parallel and distributed applications due to both task scheduling and overheads caused by parallelization on the underlying hardware. These studies are performed on Cori, the newest supercomputer system at the National Energy Research Scientific Computing Center (NERSC) a division of Berkeley Lab.

Patricia Grubel earned her Ph.D in Electrical and Computer Engineering from New Mexico State University (NMSU) with a specialty in computer architecture on August 4, 2016. She achieved this after 13 years of professional engineering experience working as a research analyst and computer systems engineer for government and industry. After taking a family break, she decided to return to engineering by pursuing her doctorate at NMSU. Her research at NMSU included performance analysis and dynamic adaptation of parallel applications using HPX, an asynchronous task-based runtime system. She is interested in continuing her research in understanding and improving performance of HPC applications.

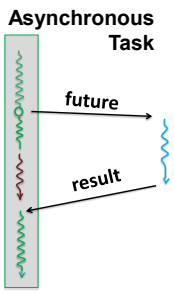
Performance Characterization of HPX - A Task-based Runtime System on the Xeon Phi™ Knights Landing

Patricia Grubel[1,2], Bryce Lelbach[2,3], Hartmut Kaiser[2,4]

[1]NMSU, Electrical and Computer Engineering, [2]STE||AR Group, [3]Berkeley Lab, [4]LSU Center for Computation and Technology

HPX – Task- based C++ Runtime

HPX is a general purpose C++ task-based runtime system that oversubscribes millions of asynchronous tasks onto a constrained number of physical threads or cores on multi-core, many-core, and heterogeneous systems. The goal of asynchronous task-based runtime systems is to ensure all processors are kept busy doing useful work. This is accomplished using futures where a value(s) required by one task, not immediately, can generate another to compute the required value(s) and can be scheduled to run on another core or hardware thread.



Intel® Xeon™ Phi Knights Landing (KNL)

- 64 "Silvermont" cores @1.4 GHz on single socket
- 4 hardware threads per core (1 per core for this study)
- 2 512 bit processing units
- 32KB L1 Cache Instruction & data each
- 1MB L2 Cache
- Memory -16 GB MCDRAM configured as L3 Cache,
  - 96 GB DDR4

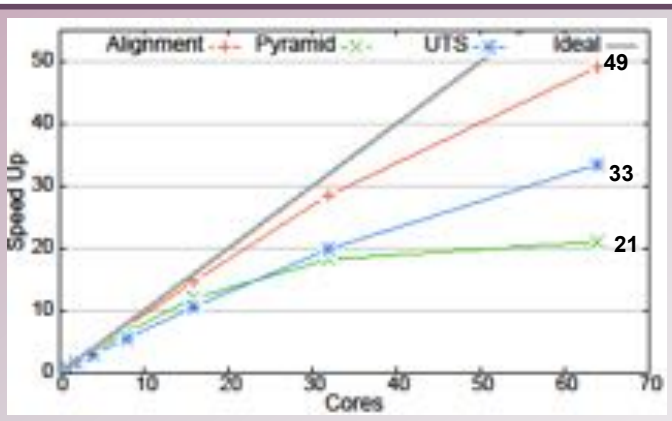
Note: This preliminary study uses 1 thread per core.

Overhead Categories

- Task Management Overheads
- Work Time Inflation

Task Management Overheads (TMO) include creation, deletion, and scheduling of tasks. On KNL these costs are on the order of microseconds per task and can be a significant percentage of execution time for fine grained tasks. UTS has the largest TMO with fine-grained tasks averaging 7.6 µsec. Alignment with coarse-grained tasks averaging 7.5 msec has the smallest TMO.

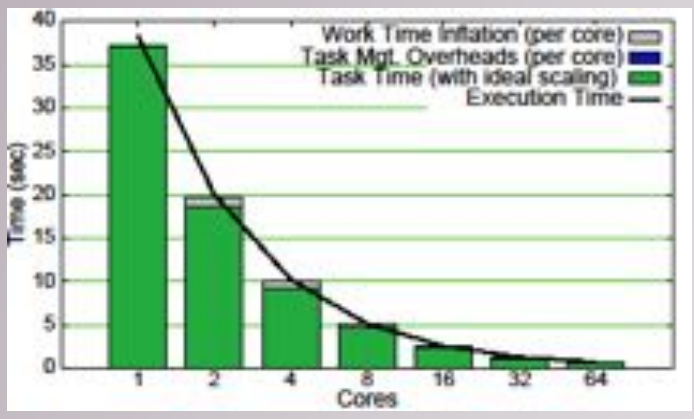
Work Time Inflation is the increase of the duration of the task caused by parallelization on the underlying hardware, and can be caused by cache misses, non-uniform memory and memory interconnect latencies, cache coherency, false sharing, and or memory bandwidth saturation. Pyramids and UTS have larger inflation of tasks due to data dependencies. Task management overheads and work time inflation are presented per core for comparison to parallel execution time.



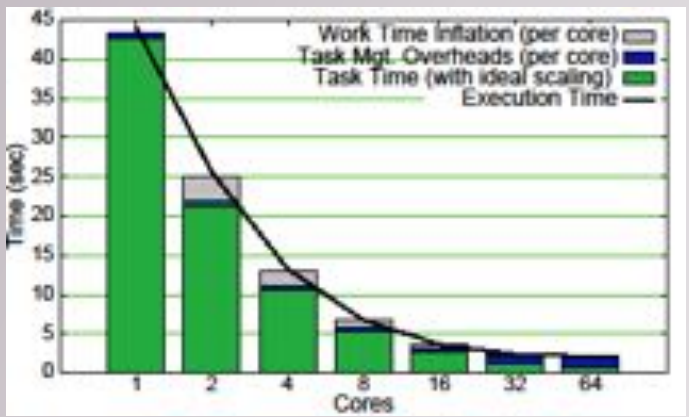
Speedup = $T_1 \div T_n$

Inncabs Benchmark Suite

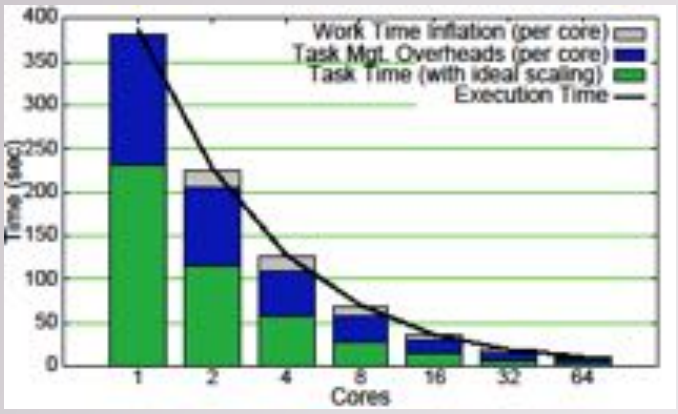
- C++ easily ported std::async → hpx::async
- Variety of Structures - Loop, Recursive or Mutex
- Variety of Task Granularity (Task Duration)



Alignment – aligns protein sequences, loop structure, coarse grain, average task 7.5 ms. Speedup 49 on 64 cores



Pyramids – 2D stencil solver, recursive balanced structure, median grain, average task 380 µs, overheads medium, Larger WTI due to data dependencies. Speedup is 21.



UTS – Unbalanced Tree Search, recursive unbalanced, fine grain, average task 7.6 µs, overheads large, speedup 33.

National Energy Research Scientific Computing Center

We appreciate the support from NERSC for use of the KNL early access white box nodes for these preliminary results. During this study NERSC was implementing Phase II of Cori, the newest supercomputer. Phase II brings on board 9304 KNL nodes with 68 cores adding 29.7 PFLOPS peak theoretical performance.

http://stellar-group.org

Hanlin He

SuperLU Pilot Libraries on KNL Machine

The future extreme scale systems will greatly impact the existing scientific and engineering applications. Porting those applications to the upcoming extreme scale systems is challenging, because of massive thread-level parallelism and heterogeneous hardware. In this poster, we present our experience of porting SuperLU on Cori, a pre-exascale, production supercomputer at Lawrence Berkley National Lab. SuperLU is a mission-critical numerical library widely used by many scientific and engineering applications. Cori is based on a new generation of Intel Xeon Phil accelerator with 64 cores and heterogeneous memory. The heterogeneous memory on Cori has a combination of on-package high bandwidth memory and off-package traditional memory. Given such platform with massive number of threads and the memory heterogeneity, how to leverage those thread-level parallelism and high bandwidth memory is challenging.

We perform detailed performance analysis on SuperLU and reveal its performance problems on Cori. Those problems include load imbalance, highly memory intensive accesses, and poor data locality. We introduce a couple of solutions to refactor SuperLU and improve its performance, including traditional loop unrolling and loop collapsing. We also reveal that the ratio between MPI tasks and OpenMP threads have a big impact on performance of SuperLU. We introduce performance models to predict the optimal ratio. Furthermore, given the limited capacity of high bandwidth memory, how to decide data placement between it and the traditional memory is a challenge. We decide data placement based on algorithm knowledge and performance profiling. We hope to explore intelligent data placement solutions in our future work.

Hanlin He is a first-year PhD student in EECS of University of California, Merced (UC Merced). She works on high performance computing, particularly focusing on performance optimization and modeling for scientific applications on large-scale parallel systems. Hanlin won outstanding undergraduate student award in the class of 2016 at UC Merced. She was a student intern in the Lawrence Berkeley National Lab at the summer of 2016, working with computational scientists on optimizing the performance of SuperLU on the next generation of supercomputers. Hanlin will also be a student volunteer in the prestigious Supercomputer Conference 2016. When she was an undergraduate student, she was actively involved in the high performance computing research and had a research paper submitted to SC'16 co-authored with other students.

SuperLU Pilot Libraries On KNL Machine

Hanlin He[1], Dong Li[2], Xiaoye (Sherry) Li[2]

[1]University Of California, Merced, [2]Lawrence Berkeley National Laboratory

Abstract

Knights Landing (KNL), the new architectures are markedly different from the existing ones, with manycore organizations of large amount of parallelism. The goal of this SuperLU pilot project is to ensure its performance on this new machines.

Background

KNL: Second generation of Many Integrated Core (MIC) architecture product from Intel, a newer product than the first generation Knights Corner (KNC).

SuperLU is a general purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations on high performance machines.

Architecture View On KNL

•**64 Cores** @ 1.3GHz, **4 hyper-threads/core** (256 threads in total)

•Up to 16 GB **MCDRAM** on-chip memory ~480 GB/s peak BW

•Up to 384 GB of **DDR off-chip** memory ~90 GB/s peak BW

• MCDRAM has 5x of DDR memory bandwidth

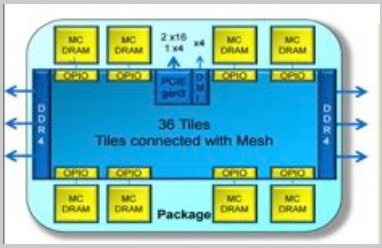


Figure 1: **Knights Landing** Overview
Photo Credit: nersc.gov

Research Challenge

The challenge of porting to any new architecture is gaining an understanding of the architectural bottlenecks that may be exposed to an application code.

Question:

Where is the performance bottleneck?
&&
What are some optimization strategies?

Materials & Methods

Intel **VTune Amplifier XE** is a performance analysis tool that enables you to find serial and parallel code bottleneck, it can help us identify opportunities for improvement.

It can perform a General Exploration analysis to understand how efficiently your code is utilizing the architecture.

•Cache misses
•Vectorization usage
•Hotspots …

Performance Analysis

Set up environment variables:

Test 1:
export OMP_NUM_THREADS=64
export KMP_PLACE_THREADS=64c,1t
With **1 MPI task**

mpirun -np **1** numactl -m **0** ./exe

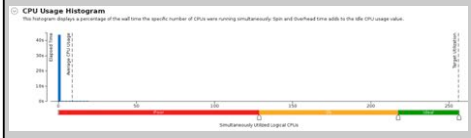


Figure 2: CPU Usage histogram from Vtune

Main problem with load imbalance! (A lot of threads are idling)

This is a Load Imbalance Problem.
Potential solution: We can use **more MPI tasks** and **less** OpenMP **threads**

Test 2:
export OMP_NUM_THREADS=8
export KMP_PLACE_THREADS=**8c,1t**

With **8 MPI Tasks (process):**
mpirun -np **8** numactl -m 0 ./exe

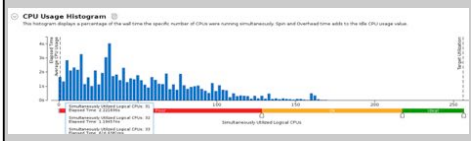


Figure 3: Improved CPU Usage

Or using more hyper-threading, it will also help to balance the workload.

!But this way won't address the fundamental OpenMP threading problems.

Major Bottleneck

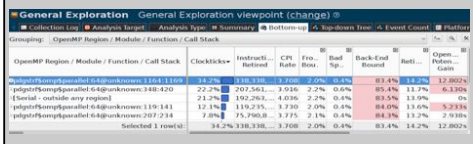


Figure 4: Bottom-Up view from VTune

Comes from **OpenMP regions**.

There are many small OpenMP regions and implicit barriers at the end of those OpenMP regions which causes execution time wasted on synchronizations.

One possible solution is to combine those small OpenMP regions, but this solution may not be always possible.

Future Works

We need a predictive model that can describes the behavior of the system, then we can easily predict the optimal ratio (between mpi & openMP) to achieve the best performance.

Furthermore, we need to know how to control data locality on those NUMA domains has impacts on memory-level parallelism, which in turn impact performance.

Acknowledgement

I would like to thank the Department of Energy's Workforce Development of Teachers and Scientists as well as Workforce Development & Education at Berkeley Lab.

Zahra Khatami

HPX Data Prefetching Iterator

The massive increase of on-node parallelism increases the complexity of memory hierarchies. Data prefetching is one of the methods used to reduce the memory accesses latency by calling the data required for the next step into the cache. In this research, we proposed the cache prefetching iterator used in the parallel algorithms in HPX to aid prefetching data of the next iteration step, that not only reduces the memory accesses latency, but also relaxes the global barrier, which results in better parallel performance. HPX has uniform higher-level API, which is fully generic and acts as an extensible framework for parallelizing applications. As a result, the HPX prefetching iterator is developed in such a way that it works with any data type and is non-intrusively usable with all parallel algorithms. The *future* construct in HPX, which is a computational result that will become available at a later time, is used extensively in the proposed prefetching method in order to have the asynchronous function execution while prefetching data. Moreover, the distance between two prefetching operations is determined based on the length of cache line, which results in increasing the effectiveness of the prefetcher and decreasing the relative cost. The performance evaluation of the HPX prefetching feature shows the scalability improvement by an average of 20-30%.

I am currently a third year Computer Engineering PhD student studying at Louisiana State University. I have completed several projects in the field of software development, parallel computing and high performance computing during past years. It has led to my attention that this firm is the right place to challenge myself through critical thinking, creativity and innovativeness. I have worked with PGX Group at Software Lab located at Oracle for developing PGX.Dist, which is a fast, parallel, in-memory graph analytic framework. Also, I have worked with Stellar Group at Center for Computation and Technology located at Louisiana State University for developing HPX, which is a general purpose C++ runtime system for parallel and distributed applications of any scale. Working in the field of high performance computing has been my career goal and I always believe that as a woman I can reach to it.

HPX Data Prefetching Iterator

Zahra Khatami, Hartmut Kaiser, and J. Ramanujam
Center for Computation and Technology, Louisiana State University, The STE||AR Group, http://stellar-group.org

Abstract

The major challenge : the difficulty of improving application scalability with conventional techniques.

One of the solutions : prefetching data before its actual access is executed.

The generic prefetching scheme proposed in HPX, which results in:

✓ improving the parallel performance by leveraging the abstraction capabilities,

✓utilizing asynchronous task-based execution flow,

✓exploiting execution policies for the fine-grained control.

Results

```
auto ctx = hpx::parallel::make_prefetcher_context(
  loop_range.begin(), loop_range.end(),
  prefetch_distance_factor,
  container_1, container_2, ..., container_n);

hpx::parallel::for_each(policy,
        ctx.begin(), ctx.end(),
        [&](std::size_t i)
        {
           container_1[i] = ...;
           container_2[i] = ...;
            .
            .
            .
           container_n[i] = ...;
        });
```

Figure 2: The prefetching method used in for_each

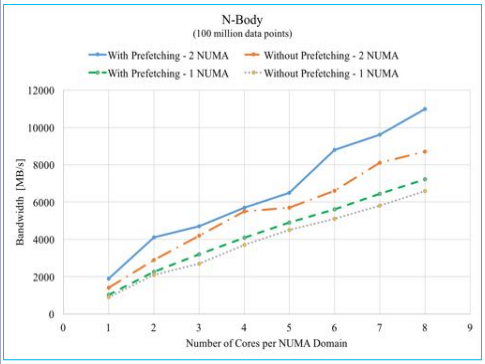


Figure 3: The data transfer rate of for_each with the standard random access iterator versus prefetching iterator

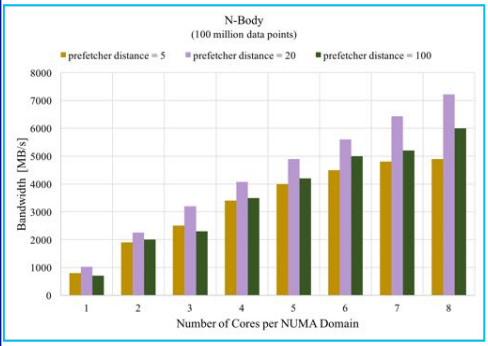


Figure 4: 1 NUMA Domain-The data transfer rate

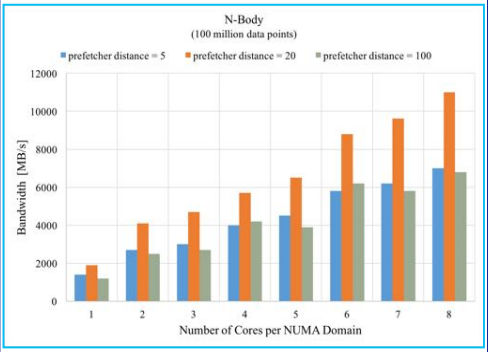


Figure 5: 2 NUMA Domain-The data transfer rate

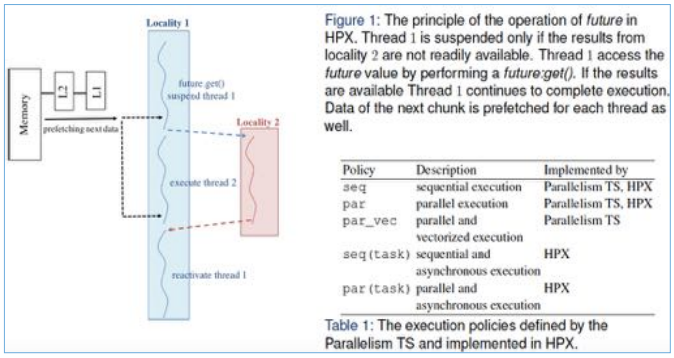


Figure 1: The principle of the operation of *future* in HPX. Thread 1 is suspended only if the results from locality 2 are not readily available. Thread 1 access the *future* value by performing a *future:get()*. If the results are available Thread 1 continues to complete execution. Data of the next chunk is prefetched for each thread as well.

Policy	Description	Implemented by
seq	sequential execution	Parallelism TS, HPX
par	parallel execution	Parallelism TS, HPX
par_vec	parallel and vectorized execution	Parallelism TS
seq(task)	sequential and asynchronous execution	HPX
par(task)	parallel and asynchronous execution	HPX

Table 1: The execution policies defined by the Parallelism TS and implemented in HPX.

HPX

➢parallel C++ runtime system

➢enables fine-grained task parallelism

➢fully generic higher-level API

➢extensible framework for parallelizing application

Introduction

Data prefetching methods:

•Hardware prefetching method: predicting the future cache misses by using the past access pattern with considering the data stream.

•Software prefetching method: prefetching data before the execution of its actual access by using the prefetch directives into the code.

•Thread based prefetching method: executing code in the prefetcher thread context and bringing the data of the next cache line into the shared cache before the main thread accesses it:

- ✓ Precomputing the load addresses accurately.
- ✓ Following more complex pattern compared to the other methods.

However, scaling can be degrade with

Thread based prefetching: Cache misses, Global barriers

and Resource competition.

The cache prefetcher used in HPX aids prefetching that

✓reduces the memory accesses latency, and

✓inhibits the global barrier.

✓*for_each* helps creating sufficient parallelism by determining the number of the iterations to run on each HPX thread.

✓HPX threads makes the invocation of the loop asynchronous, while the data of all containers within the loop of the next step is prefetched in each iteration.

✓HPX is able to prefetch data in sequential or in parallel with applying an execution policy.

✓HPX prefetcher works with any data type of the containers and even if each container has different data type.

Prefetching Iterator Implemented in HPX

•***for_each*** is one of the HPX parallel algorithms used to evaluate the proposed prefetching method.

•Data of the next iteration step is prefetched in the cache memory with the prefetching iterator called in each iteration within the ***for_each*** .

•HPX combines prefetching method with the asynchronous task execution by providing a new *future* instance representing the result of the function execution (Figure 1).

•The program execution is divided into several chunks within ***for_each*** (Figure 2) and its iterator is developed to prefetch the data of the next chunk size in either sequential or in parallel.

•The prefetching iterator is initialized in ***make_prefetcher_context*** and it executes with ***ctx.begin(). ctx*** is the struct that references to all container in the

•The distance between each two prefetching operations is computed based on the value of ***prefetch_distance_factor***, which is the factor of the length of the cache line.

Experimental Results

In an N-Body problem, there are N particles moving under the influence of the gravitational attraction.

Prefetching iterator increases bandwidth vs. standard random access iterator by 30% on average using two NUMA domains with 8 threads each (figure 3).

The results of the performance of the prefetching iterator with different ***prefetch_distance_factor*** are shown in figure 4 and 5 for 1 and 2 NUMA domains respectively:

•For the large distance, data prefetching cannot improve the parallel performance.

•Very small prefetcher distances make more data to be prefetched,
which become more expensive and dominate the gains from prefetching.

We would like to thank Antoine Tran Tan and Adrian Serio from Louisiana State University for the invaluable and helpful suggestions to improve the quality of the research. This works was supported by the NSF award 1447831.

Jiajia Li

Model-driven Sparse CP Decomposition for High-Order Tensors

Tensor analysis provides a powerful set of methods to analyze and extract knowledge from very sparse, high-order data in various applications, such as machine learning, social network analytics, healthcare analytics, neuroscience, and image processing to name a few. This work optimizes the classical CANDECOMP/PARAFAC decomposition (CPD) method by using memoization technique to increase data reuse. We propose an adaptive tensor memoization algorithm to accelerate the most expensive computational core of CPD, the Matricized Tensor Times Khatri-Rao Product (MTTKRP) sequence. Considering diverse sparse tensor features and architecture characteristics, a novel model-driven framework is constructed to help determine the optimal algorithm and to do trade-off between time and space. Compared to previous work, our approach achieves up to 10X speedup on up to 85th-order real sparse tensors. Our algorithm also shows near constant scalability with respect to the tensor order, while using acceptable storage space.

Jiajia Li is a third-year Ph.D. candidate in Computational Science & Engineering at Georgia Institute of Technology. She works in Professor Richard Vuduc's "HPC Garage" group as a graduate research assistant. Her current research focuses on optimizing tensor algorithms such as tensor decomposition and tensor computational kernels (e.g. tensor-times-matrix multiply (TTM) and metricized tensor times Khatri-Rao product (MTTKRP)) from both algorithm and architecture aspects on various platforms.

In the past, she received a Ph.D. degree from Institute of Computing Technology at Chinese Academy of Sciences. Her research was about optimizing and auto-tuning irregular algorithms, including sparse matrix vector multiplication, algebraic multi-grid, and dynamic programming on parallel architectures. Before this, she received her B.S. in Information and Computing Science from Dalian University of Technology.

Model-driven Sparse CP Decomposition for High-Order Tensors

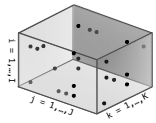
Jiajia Li, Ioakeim Perros, Jimeng Sun, Richard Vuduc
Georgia Institute of Technology

Background

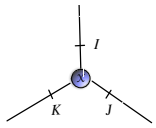
Tensor decomposition is a set of unsupervised methods to analyze and extract knowledge from tensors, which is widely used in healthcare analytics, image processing, machine learning, and social network analytics.

Sparse tensors

Many real-world tensors are hyper-sparse and have specific features. To discover useful knowledge, efficient sparse algorithms are critical to performance and scalability.



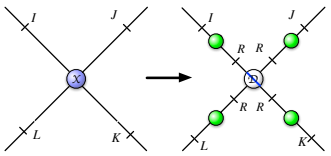
A 3rd-order sparse tensor



Tensor network diagram

CPD

CANDECOMP/PARAFAC decomposition (CPD) approximates an input tensor as a sum of component rank-one tensors, given the number of desired components. CPD is scalable in time and space compared to other low-rank methods.



CPD on a 4th-order tensor

Contributions

- This work proposes an adaptive and efficient algorithm for the computational kernel, Matricized Tensor Times Khatri-Rao Product operation (MTTKRP) of the classical CANDECOMP/PARAFAC decomposition (CPD), by minimizing redundant computations.
- We also build a model-driven procedure to determine the adaptable algorithmic parameters for different input sparse tensors and enable the trade-off between time and space based on the user need and memory resource.
- Our adaptive algorithm shows up to 10x speedup compared to state of the art on real-world high order tensors. Our method also shows near constant scalability with respect to the tensor order, while using acceptable storage space.

Algorithms

- Decouple an MTTKRP operation into a TTM and a TMHR.

Algorithm 3 Memorized Full MTTKRP algorithm (memFM).

Input: A fourth-order sparse tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K \times L}$, dense factor matrices $\mathbf{B} \in \mathbb{R}^{J \times R}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$, $\mathbf{D} \in \mathbb{R}^{L \times R}$, the product order p_o;
Output: Updated dense factor matrix $\tilde{\mathbf{A}} \in R^{I \times R}$;

▷ $\tilde{\mathbf{A}} \leftarrow \mathcal{X}_{(1)}(\mathbf{D} \odot \mathbf{C} \odot \mathbf{B})$

// Compute as $p_o = [4, 3, 2]$.
1: Save $\mathcal{Y}^{(1)}$; $\mathcal{Y}^{(1)} \leftarrow \mathcal{X} \times_4 \mathbf{D}$ — TTM
2: Save $\mathcal{Y}^{(2)}$; $\mathcal{Y}^{(2)} = \text{TMHR}\ (\mathcal{Y}^{(1)}, \mathbf{C})$ — TMHR
3: $\tilde{\mathbf{A}} = \text{TMHR}\ (\mathcal{Y}^{(2)}, \mathbf{B})$ **return** $\tilde{\mathbf{A}}$;

1st MTTKRP $\tilde{\mathbf{A}} \leftarrow \mathcal{X}_{(1)}(\mathbf{D} \odot \mathbf{C} \odot \mathbf{B}) \Leftrightarrow \tilde{A} \leftarrow \mathcal{X} \bullet D \blacksquare C \blacksquare B$, $\mathcal{Y}^{(1)}$, $\mathcal{Y}^{(2)}$

Algorithms

- Design a simple memoization algorithm for MTTKRP sequence.

$\tilde{\mathbf{A}} \leftarrow \mathcal{X}_{(1)}(\mathbf{D} \odot \mathbf{C} \odot \mathbf{B}) \Leftrightarrow \tilde{A} \leftarrow \mathcal{X} \bullet D \blacksquare C \blacksquare B$

Reuse degree: 2 — $\mathcal{Y}^{(1)}$, $\mathcal{Y}^{(2)}$

$\tilde{\mathbf{B}} \leftarrow \mathcal{X}_{(2)}(\mathbf{D} \odot \mathbf{C} \odot \tilde{\mathbf{A}}) \Leftrightarrow \tilde{B} \leftarrow \mathcal{Y}^{(2)} \blacksquare \tilde{A}$

$\tilde{\mathbf{C}} \leftarrow \mathcal{X}_{(3)}(\mathbf{D} \odot \tilde{\mathbf{B}} \odot \tilde{\mathbf{A}}) \Leftrightarrow \tilde{C} \leftarrow \mathcal{Y}^{(1)} \blacksquare \tilde{B} \blacksquare \tilde{A}$

$\tilde{\mathbf{D}} \leftarrow \mathcal{X}_{(4)}(\tilde{\mathbf{C}} \odot \tilde{\mathbf{B}} \odot \tilde{\mathbf{A}}) \Leftrightarrow \tilde{D} \leftarrow \mathcal{X} \bullet \tilde{C} \blacksquare \tilde{B} \blacksquare \tilde{A}$

One Group

- Design an **adaptive** memoization algorithm for MTTKRP sequence, to flexibly choose number of memoized MTTKRPs and reuse degrees.

$\tilde{\mathbf{A}} \leftarrow \mathcal{X}_{(1)}(\mathbf{D} \odot \mathbf{C} \odot \mathbf{B}) \Leftrightarrow \tilde{A} \leftarrow \mathcal{X} \bullet D \blacksquare C \blacksquare B$

Reuse degree: 1 — $\mathcal{Y}^{(2)}$

$\tilde{\mathbf{B}} \leftarrow \mathcal{X}_{(2)}(\mathbf{D} \odot \mathbf{C} \odot \tilde{\mathbf{A}}) \Leftrightarrow \tilde{B} \leftarrow \mathcal{Y}^{(2)} \blacksquare \tilde{A}$

Group 1

$\tilde{\mathbf{C}} \leftarrow \mathcal{X}_{(3)}(\tilde{\mathbf{B}} \odot \tilde{\mathbf{A}} \odot \mathbf{D}) \Leftrightarrow \tilde{C} \leftarrow \mathcal{X} \bullet \tilde{B} \blacksquare \tilde{A} \blacksquare D$

Reuse degree: 1 — $\mathcal{Z}^{(2)}$

$\tilde{\mathbf{D}} \leftarrow \mathcal{X}_{(4)}(\tilde{\mathbf{B}} \odot \tilde{\mathbf{A}} \odot \tilde{\mathbf{C}}) \Leftrightarrow \tilde{D} \leftarrow \mathcal{Z}^{(2)} \blacksquare \tilde{C}$

Group 2

- Decrease the operation number.
- Probably increase the storage space.

Tradeoff

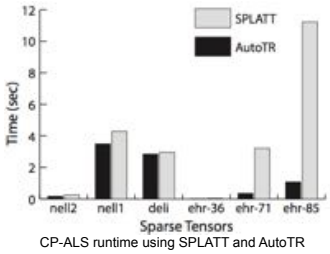
- Build a mode-driven auto-tuner to accurately predict the algorithm parameters for sparse tensors.

Results

We test our algorithm on Intel Xeon E7-4820 platform using 16 threads.

Dataset	Order	Max Mode Size	NNZ	Density
nell2	3	30K	77M	1.3e-05
nell1	3	25M	144M	3.1e-13
deli	3	17M	140M	6.1e-12
ehr-36	36	19	11K	4.7e-26
ehr-71	71	21	221K	1.4e-55
ehr-85	85	21	920K	7.9e-68

Sparse tensor description



CP-ALS runtime using SPLATT and AutoTR

Conclusion

Our optimization to increase the reuse of MTTKRP sequence is beneficial for high-order tensors. We use acceptable space to trade for higher performance.

References

B. W. Bader, T. G. Kolda et al., "Matlab tensor toolbox (Version 2.6)," Available online, February 2015.

S. Smith, N. Ravindran, N. Sidiropoulos, and G. Karypis, "Splatt: Efficient and parallel sparse tensor-matrix mul- tiplication," in Proceedings of the 29th IEEE Interna- tional Parallel & Distributed Processing Symposium, ser. IPDPS, 2015.

Acknowledgement: This project is supported by the U.S. National Science Foundation (NSF) Award Number 1533768.

Fang Liu

Building a Research Data Science Platform from Industrial Machines

Data Science research has a long history in academia which spans from large-scale data management, to data mining and data analysis using technologies from database management systems (DBMS's). While traditional HPC offers tools on leveraging existing technologies with data processing needs, the large volume of data and the speed of data generation poses significant challenges. Using the Hadoop platform and tools built on top of it drew immense attraction from academia after it gained tremendous success in industry.

Georgia Tech (GT) received a donation of 200 compute nodes from Yahoo. Turning these industry retired machines into a research platform poses unique challenges, such as: nontrivial hardware design decisions, configuration tool choices, node integration into existing HPC infrastructure, partitioning resource to meet different application needs, software stack choices, etc.

Currently, we have 40 nodes up and running, in which 25 nodes run as a Hadoop/Spark cluster, 12 nodes run as a HBase/OpenTSDB cluster, the others run as service nodes. We performed a number of successful tests, with Spark Machine Learning algorithms using a 177GB image dataset, Spark DataFrame/GraphFrame with a Wikipedia dataset, Hadoop/MapReduce word count on a 300G dataset. The OpenTSDB cluster is for real-time time series data ingestion and analysis for sensor data. We are still working on bring up more nodes in the next couple of months. We share our first-hand experience gained in our journey, which we believe will benefit and inspire other academic institutions.

Dr. Fang (Cherry) Liu is a Research Scientist at Partnership for Advanced Computing Environment (PACE) center in Georgia Institute of Technology, and she also holds the Adjunct Associate Professor title from school of Computational Science and Engineering (CSE). With joint position, she actively provides expert scientific computing consulting service, educates campus HPC community as well as does collaboration research with faculties from multiple departments.

Before joining Georgia Tech, she was an assistant scientist at mathematics and computational science division at Department of Energy (USDOE) Ames Laboratory, where she closely worked with world-class domain scientists from physics, chemistry and fusion energy on providing HPC solutions.

Dr. Liu holds a Ph.D. degree from Indiana University at Bloomington in Computer Science, her broad interests lay on parallel/distributed scientific computing, software interface design, multi-physics code coupling, data management and provenance, big data infrastructure design and implementation.

Building a Research Data Science Platform from Industrial Machines

Georgia Tech

Fang (Cherry) Liu
Partnership for an Advanced Computing Environment (PACE)

Fu Shen
School of Computer Science

Paras Jain
School of Computer Science

Duen Horng Chau
School of Computational Science and Engineering

Turning industry machines into a high-performance research data science platform based on Hadoop facilitates computing cycle reuse!

Motivations and Goal

- **Motivations:**
  - Free cycles from 200 compute nodes donated by Yahoo
  - Allow deep understanding on Hadoop ecosystem from ground-up building experience
  - Give more freedom to try out up-to-date software to bring more research value in which existing cloud solutions won't provide.
- **Goal:** Build a research data science platform (DSP) based on the Hadoop platform

Existing Tools Comparison

- Hadoop distributions like Hortonworks and Cloudera have drawbacks for a research DSP because:
  - vendor code less compatible with configuration tools
  - infrequent update schedules
  - limited library selection in enterprise releases
  - harder to debug proprietary libraries without fee-based consulting
- Amazon Web Services (on EC2 or EMR), Azure (Data Lake products) and Google Cloud Platform
  - Amazon EMR uses S3 which has higher access latency
  - Amazon EC2 requires system administrator knowledge about software installation
  - Azure uses the Hortonworks distribution
  - Google Cloud Platform's Dataproc is new and does not have significant adoption yet

Hardware and Software Configuration

The cluster physically located at School of Computational Science and Engineering

- 200 compute nodes run Red Hat Enterprise Linux 6.7
  - 2x 4-core Intel Xeon CPUs (2.5GHz)
  - 24GB memory
  - Service nodes use RAID 1 mirroring (2x 1TB)
  - DataNodes use separate 500GB data and OS disks
- Two clusters are built:
  - 24-node Hadoop (2.7.2), Spark (1.6.1) running on YARN
  - 12-node Hbase (1.1.5), OpenTSDB (2.2.0) cluster

Challenges

- **Performance:** how to get most performance from existing hardware?
- **Maintenance:** how to make maintenance minimally intrusive?
- **Sustainability:** how to enable horizontal scalability to more compute nodes in future?

Testing and Evaluation

- Several test cases to verify DSP runs as expected
  - Logistic regression from Spark ML
  - Wordcount from MapReduce
  - Spark GraphFrame test
- Four test data sets from 1MB to 300GB (see Table 1)
- Hadoop Wordcount runtime was examined across Java heap parameters (see Table 2)
- Spark ML Logistic regression runtime was examined across several cluster configurations (see Table 3)

IDs	Name	Size
ds2	HIGGS.csv	7.5GB
ds3	wiki-Vote.txt	1.0MB
ds4	wikipedia	300GB

Table 1: Sizes of test data sets

map.memory.mb	4096	2048	1560	2560
map.java.opts.-Xmx (MB)	3686	1843	1400	2304
reduce.memory.mb	5120	2048	2048	2560
reduce.java.opts.-Xmx (MB)	4608	1843	1843	2304
Runtime (hours)	**2.18**	**1.41**	**1.66**	**2.11**

Table 2: Hadoop Wordcount runtime across Java heap configurations running on ds4 data set

driver-memory	8G	6G	8G	8G	10G	8G
executor-memory	4G	4G	4G	8G	8G	4G
num-executors	8	4	4	8	8	8
executor-core	4	8	8	4	4	8
Runtime (mins)	**27**	**38**	**41**	**49**	**80**	**23**

Table 3: Spark ML Linear Regression runtime across executor configurations on ds1

Conclusions

- We created 40-node multi-clusters using donated Yahoo compute nodes over 2 months
- Performance, maintenance and sustainability goals established
- Test suites cover tasks Georgia Tech researchers plan to perform on these clusters
- **Future Work:** bringing up more nodes, further performance tuning, investigate high-throughput inter-cluster communication

Acknowledgements

We would like to thank Brian MacLeod, Andre McNeil, Dan Zhou, Paul Manno, Neil Bright, Mehmet Belgin, Daniel Forsyth, Deborah Davis, Meera Kamath, Josephine Palencia, Blake Fleischer and Michael Brandon for their help in the project

Oana Marin

Lossy Data Compression in a highly scalable Computational Fluid Dynamics code

Even the most scalable and performant applications which are data intensive can be severely hindered on supercomputers by the low speed of the I/O, despite many current strategies such as asynchronous or parallel I/O. To this end we developed a lossy data compression algorithm which can reduce, e.g. for visualization, even the most challenging data fields down to 3% of their original size. Nonetheless, our compression strategy can be used also for resilience, or more general post processing purposes, by imposing an a priori accuracy threshold. This is achieved via the a priori error estimator presently derived which we subsequently tested on highly turbulent fields of up to 10 billion grid points. The compression is performed on the fly by truncating the data at the compute node level and bitwise encoding it at I/O node level in order to accelerate the I/O speed. The implementation at compute node is embarrassingly parallel requiring absolutely no communication between data packages, and also exhibits excellent per MPI rank efficiency via tensor product data representation and performant matrix-matrix product implementation.

Oana Marin is a postdoctoral researcher at Argonne National Laboratory, currently working on development of highly scalable algorithms using spectral element methods. In 2013 she joined the group of the Gordon Bell winning code Nek5000, where she gained experience on applications ranging from nuclear engineering, multiphase flows to turbulence modeling and magneto-hydrodynamics. She holds a PhD in Applied Mathematics from the Royal Institute of Technology, Stockholm, Sweden, where she specialized in developing numerical methods for boundary integral equations.

Lossy Data Compression in a highly scalable Computational Fluid Dynamics code

O. Marin, M. Schanen, P. Fischer

Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY
Office of Science

Lossy data compression – controllable data truncation

- Lower memory footprint on disk
- Faster IO speeds
- Truncate the over resolved parts of the solution
- Perform calculations on only relevant parts of data
- Find accurate error threshold

Spectral element data layout

Velocity represented over each element

$$u(x)|_{\Omega_e} = \sum_{i,j,k} u^e_{ijk}\phi_i(x)\phi_j(y)\phi_k(z)$$

And over the entire mesh

$$u(x)|_{\Omega} = \sum_{i=1}^{N^3 E} u^g \phi_{gi}, \quad \phi_{gi} \begin{cases} 1, & g = e \\ 0, & g \neq e \end{cases}$$

Discrete Cosine Transform

Transform from real space to DCT space given by

$$\hat{u}(k_1,\ldots,k_d) = c_{k_1}\cdots c_{k_d}\sum_{n_1=0}^{N}\cdots\sum_{n_d=0}^{N} u(n_1,\ldots,n_d)\cos\left[\frac{\pi}{2N}(2n_1+1)k_1\right]\cdots\cos\left[\frac{\pi}{2N}(2n_d+1)k_d\right], \quad k_j = 0,\ldots,N,\ c_{k_j} = \begin{cases}\sqrt{1/N}, & k_j = 0 \\ \sqrt{2/N}, & 0 < k_j \leq N\end{cases}$$

Properties of the DCT transform.

1. Orthogonality: $T^T T = T T^T = I$, where I is the identity matrix.
2. $T^{-1} = T^T$: thus the inverse DCT transform IDCT is the transpose of the DCT.
3. Energy preservation: $u^T u = (Tu)^T (Tu) = u^T T^T T u = u^T u$.
4. Energy compactness.
5. Separability: the three-dimensional DCT (let us define it $T^3(u)$) is a sequence of one-dimensional DCT transforms: that is, $T^3(u) = (T \otimes T \otimes T)u$.

A priori error evaluation

Domain Ω consists of elements Ω_e: $\Omega = \cup_e \Omega_e$

Define the norm for curvilinear meshes $\|u\| = \sqrt{\frac{\int_\Omega u^2\, d\Omega}{\int_\Omega d\Omega}}$

From $\|u\|^2_{L_2} = \sum_e \|u_e\|^2_{L_2} = \frac{|\Omega|}{N^3}\sum_{e=1}^{E}\frac{\|u_e\|^2_{L_2}}{V_e} \to \|u\|^2_{L_2}$

To obtain a global error $\|\tilde{u}\| \leq \epsilon$, is sufficient to demand $\sqrt{\frac{\sum_e \|u_e - \tilde{u}_e\|^2_{L_2}}{|\Omega|}} \leq \epsilon$

The discrete L_2 norm is $\|u\|_{L_2} = \sqrt{\int_\Omega u^T u\, d\Omega} = \sqrt{u^T B u}$ (B mass matrix)

In DCT space $\hat{u} = Tu$ the norm is the same as in real space

$\|\hat{u}\|_{L_2} = \sqrt{\int_\Omega \hat{u}^T \hat{u}\, d\Omega} = \hat{u}^T B \hat{u} = u^T T^T B T u = u^T B u$

Huffman Encoding

On each element Ω_e of size $M = N^d$ (d - spatial dimension). Assume each entry $e \mapsto \underbrace{[b_n, \ldots, b_0 . b_{-1}, \ldots, b_{-m}]}_{n+1 \text{ bits}}$

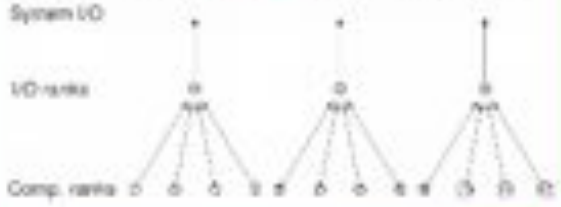
Truncate M by setting K entries to 0.

Compression ratio = $\frac{M}{K}$

Perform bitwise encoding h

$$h([b_n,\ldots,b_0 . b_{-1}\ldots]) = \begin{cases} \underbrace{0}_{1 \text{ bit}}, & \text{if } b_n = \cdots = b_{n-l+1} = 0 \\ \underbrace{[1, b_n, \ldots, b_{n-l+1}, s]_{\text{enc}}}_{n-l+1 \text{ bits}} \end{cases}$$

Encoding performed at I/O node level prior to write to disk



Algorithm

procedure TRANSFORM
  for $k = 1, N_d$ **do**
    Evaluate $T^k u$
    $u = T^k u$
  end for
end procedure
procedure COMPRESS
  if IOnode **then**
    huff_encode(u)
  end if
end procedure

Consider $u \underbrace{\to}_{\Delta x^d V} Tu (= \hat{u})$ a 1d transform

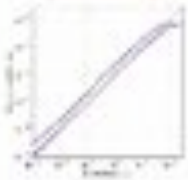
then in 3d $w = (T_x \otimes T_y \otimes T_z)u = T_z u T_y^T T_x^T$

reduce evaluations from $O(N^{2d})$ to $O(N^{d+1})$

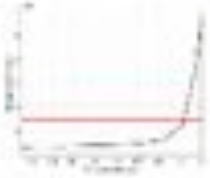
in dimension d.

Truncation in DCT space based on a priori error estimator.

A posteriori error matches up to billions of gridpoints imposed treshold (dashed red).



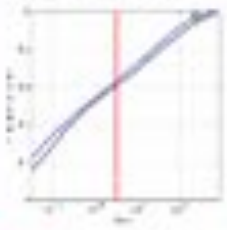
IO speed at IO node, red line uncompressed, dashed red IO bandwidth



Results

Compression performed on 3.5 billion gridpoints, flow past an airplane wing simulation [2].

A truncation of 97% of the original data is sufficient for reliable visualizations.



Possible to achieve better compression than by single precision.

Error estimator gives an accurate handle on compression within error bounds.

Tip of the wing reconstructed

After 99% truncation

After 97% truncation

Resilience issues addressed easily using the error estimator.

Full control over errors incurred via compression.

[2] S.M. Hosseini et al. Direct numerical simulation of the flow around a wing section at moderate Reynolds number. Int. J. Heat & Fluid Flow, 2016.

Bhavani S Nanjundaiah

High Performance Big Data Analytics using Spectrum Scale

Apache Hadoop is used to process quintillion bytes of data that is created every day which can be structured, semi-structured or unstructured data. It uses HDFS (Hadoop Distributed File System) as a standard way of storing these data and MapReduce to process the data stored in HDFS. Though HDFS is widely used file system to store Big data, there are other file systems which offer an enterprise-class alternative to HDFS. IBM® Spectrum Scale, formerly IBM General Parallel File System (IBM GPFS) is one such file system which has several advantages over HDFS. Spectrum Scale is a POSIX-compliant, high-performing and scalable file system. It is widely used in HPC applications worldwide for its features. In 2009, Spectrum Scale was extended to work seamlessly in the Hadoop ecosystem and is available through a feature called File Placement Optimizer (FPO). Storing application's Hadoop data using FPO allows to gain advanced functions and high I/O performance required for many big data operations. FPO provides Hadoop compatibility extensions to replace HDFS in a Hadoop ecosystem, with no changes required to Hadoop applications.

In this poster we compare the features of Spectrum Scale versus HDFS. We also talk about the **Spectrum Scale Transparency Connector** which emulates the HDFS environment. It implements HDFS Namenode and HDFS Datanodes which allows Hadoop clients to interact with Spectrum Scale file system using *hdfs dfs* system command.

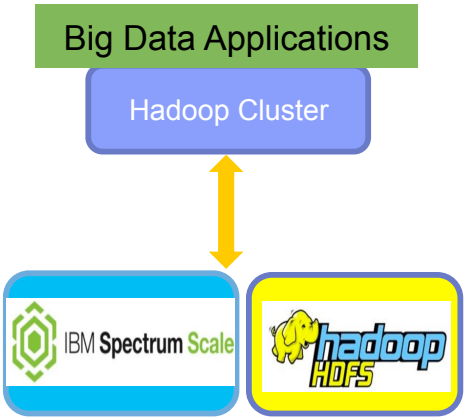
This poster also includes the integration of Spectrum Scale file system with Apache Ambari. Apache Ambari is a tool used to deploy, manage and monitor Hadoop clusters. The poster shows how IBM Integration modules helps in seamlessly integrating the SS file system along with the HDFS Service. Integration module also provide functionality to unintegrated Spectrum Scale file system and switch back to HDFS which allows user to compare the advantages of Spectrum Scale filesystem versus HDFS.

Bhavani S Nanjundaiah is working with HPC software development particularly IBM MPI library. From last seven years she has worked on multiple aspects of HPC like designing solution using IBM Technical Computing products for Big Data and Life Sciences applications, development of network management and monitoring software. She has multiple patents in the post silicon processor validation domain. Her areas of interest are to understand and address the gap between hardware, mid-tier libraries like MPI and how libraries like MPI can benefit the applications.

High Performance Big Data Analytics using IBM® Spectrum Scale™

Bhavani S Nanjundaiah Deepak K Jha

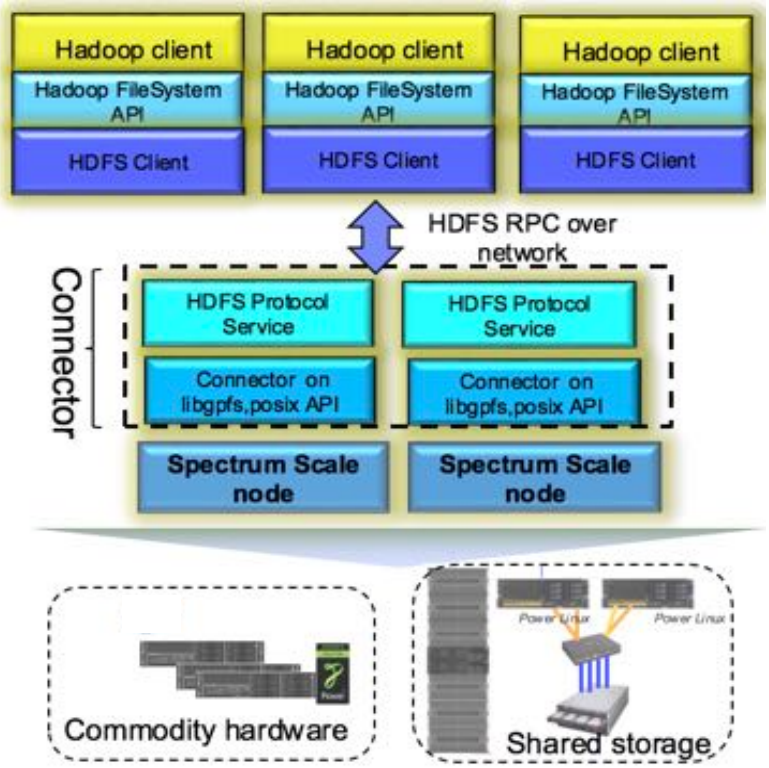
IBM Spectrum Scale for Big Data Applications - Key advantages



- IBM Spectrum Scale (formerly GPFS) is a high-performance clustered file system
- All Big Data applications run seamlessly with Spectrum Scale without needing any changes
- Supports both **Shared Everything** and **Shared-nothing** architecture
- Spectrum Scale complies **POSIX** industry standard, enabling data ingest and export through NFS, SMB and Object protocols
- Unlike HDFS, Spectrum Scale does not require proprietary non standard tools for **ingestion** and export of data
- Spectrum Scale allows random reads and writes to a file while allowing applications to do **In-Place analytics**
- Spectrum Scale overcomes the performance limitations associated with HDFS dependency on the underlying native OS file system
- Spectrum Scale provides **enterprise level** features like encryption, ILM, DR, GPFS Native RAID

HDFS Protocol Service Design (Transparency Connector)

- Transparency Connector has proprietary implementation of HDFS NameNode and DataNode to support Spectrum Scale
- Transparency NameNode is lightweight as well as **Stateless** and as the data is striped on to the disks and can be recovered in case of failures
- Support workloads that have hard coded HDFS dependencies
- Full Kerberos support in Hadoop ecosystem
- Federate multiple Spectrum Scale clusters
- Supports multiple Hadoop clusters on the same filesystem for better workload isolation
- Leverage HDFS client cache for better performance
- No need to install Spectrum Scale clients on all nodes



IBM Spectrum Scale Integration with Apache Ambari

Ambari

Framework to provision, manage and monitor Hadoop Cluster

IBM Open Platform with Apache Hadoop

- Spectrum Scale is added as a new service on existing IOP-HDFS cluster
- IOP-Hadoop cluster can be deployed on new or existing Spectrum Scale file system
- Spectrum Scale can be unintegrated to flip back to native HDFS
- Data will not be moved back and forth between HDFS & Spectrum Scale
- Alerts and Monitoring of the Hadoop cluster work seamlessly
- Supports upgrades for Spectrum Scale and Connector

Lena Oden

Towards efficient usage of heterogeneous memory architectures

The increasing gap between processor performance and memory bandwidth has led to the development of several novel memory technologies. Heterogeneous memory systems with multiple layers of memory balance the requirements in both bandwidth and capacity by providing layers with larger but slower and less but faster memory. These will be provided by upcoming computing machinery, such as Intel's Knights Landing (KNL) platform that has up to 16GB high-bandwidth on-package memory and additional 384GB off-package DDR4 memory. However, the optimal usage of these heterogeneous architectures presents new challenges for HPC developers.

The goal of my work is to provide a programming model and runtime system that allows developers the efficient, effective, and transparent use of these heterogeneous memory architectures.

One main focus of the framework is the online and offline optimization of data object distribution across the multiple layers, as previous work suggests a significant improvement in performance if highly accesses objects are allocated in fast memory.

High bandwidth memory is a limited resource, and therefor it will not be possible to allocate all highly accesses structures in high bandwidth memory.

To get the best performance in these cases, we developed an asynchronous prefetching library that allows continuous and non-continuous prefetching of objects in lower-bandwidth memory to a fast-memory cache. Our case studies have shown that prefetching can help to increase the performance.

Concluding, the framework will support developers to better harness the potential of the upcoming heterogeneous memory architectures.

Lena Oden is postdoctoral associate at Argonne National Labatory, USA, in the Mathematics and Computer Science division. She is part of MPICH development team in the Programming Models and Runtime systems group and responsible for the UCX support inside MPICH and the usage of heterogeneous memory. Lena also works on programming models and runtime support for heterogeneous memory architectures.

Before she came to Argonne, she received her PhD in Computer Science from the University of Heidelberg in Germany for her work on communication models between distributed GPUs. She received a PhD fellowship from the Fraunhofer association and worked at the Fraunhofer Institute for industrial Mathematics in Germany. She published several papers about GPU communication and is also part of program committees of several conferences and workshops and has reviewed papers for different journals for HPC and parallel computing.

Towards efficient usage of heterogeneous memory architectures

Lena Oden, Pavan Balaji

Heterogeneous memory architectures

The increasing gap between processor performance and memory bandwidth has led to the development of several novel memory technologies. The requirements increase not only in bandwidth but also in capacity of the underlying memory system.
A single memory technology is unlikely to be able to fulfill both requirements – bandwidth and capacity. Instead, heterogeneous memory systems with multiple layers of memory balance the two requirements, by providing layers with larger but slower bandwidth and less but faster memory.

A new challenge

From an application developer's perspective, efficient usage of heterogeneous memory is a fresh challenge, since it introduces a new level of optimization. An optimized application will benefit from both the high bandwidth and the high capacity of these systems.
The goal of my work is to provide a programming model and runtime system that makes the usage of heterogeneous memory management as simple and efficient as possible.

Intel KNL memory models

Recently Intel released the Knights Landing (KNL) platform with two layers of memory: up to 16 GB of high-bandwidth on-package memory (MCDRAM) and an additional 384 GB off-package DDR4 memory.

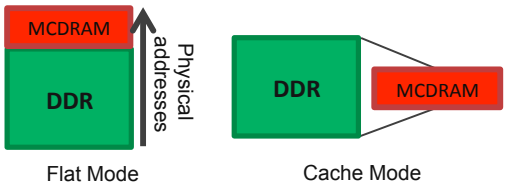


Figure 1: MCDRAM usage modes of the KNL

The KNL provides three modes for the usage of the MCDRAM. In the flat mode, the MCDRAM memory is seen as normal memory and lies in the same address space as does the DDR memory
In the cache mode, the usage of the MCDRAM is transparent to the user. It is used as a direct mapped hardware cache.

Object distribution

Figure 2 shows the performance for a 2D 5-point stencil of the size 26kX26k with different memory usage models
Allocating the grids in MCDRAM shows significantly better performance than allocating them in DDR memory.

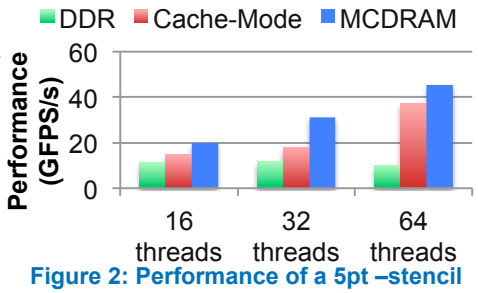


Figure 2: Performance of a 5pt –stencil on a heterogeneous memory system

Because of the lower bandwidth of the DDR memory, the application is not scalable for more than 16 threads if the MCDRAM is not used.
Using MCDRAM directly also results to a better performance than using the cache mode.

```
#pragma hbw_mem A
#pragma omp parallel
       for(i = 1, k = 1; i<size_y; i++,k++) {
            for(j = 1; j <size_x; j++) {
                B[ind(i,j)] = 0.5 * A[ind(k,j)]
                . . . }}}
```

Figure 4a: Instrumented Code for HBM usage

Source-to-source compiler

```
for(l = 1; l< size_y; l+=CHUNK) {
    prefetcher->start(offset, fetch_size);
    cache = (double*) prefetcher->sync();
#pragma omp parallel
       for(i = 1, k = 1; i<l+CHUNK; i++,k++) {
            for(j = 1; j <size_x; j++) {
                B[ind(i,j)] = 0.5 * cache[ind(k,j)]
                . . . }}}
```

Figure 4b: Stencil with explicit prefetch instructions

Asynchronous prefetching to MCDRAM

Allocating all heavily accessed objects in the fastest memory layer is not always possible. For a grid size of 46kX46k, for example, neither the input field nor the output field of a stencil completely fits into MCDRAM. In this case, access to the lower bandwidth of the DDR memory may drastically decrease the performance.
Prefetching and migration of data between the memory layers can improve the performance in these cases.

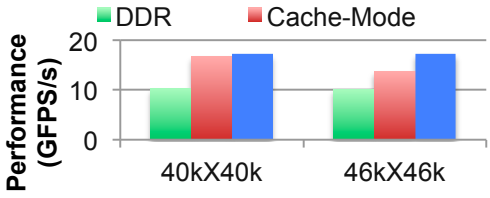


Figure 3: Performance of a 5pt-stencil with size > MCDRAM size

We developed a prefetching library that allows asynchronous prefetching from DDR memory to MCDRAM.

The data are asynchronously moved between DDR memory and a software-managed cache in MCDRAM. The prefetching can be overlapped with computation.
For large grid sizes, where the data no longer completely fit into MCDRAM memory, this approach outperforms even the cache mode. Figure 3 shows the results for two problem sizes. For a 40kX40 grid size, one grid fits into MCDRAM

Programming model and runtime support for heterogeneous memory

Our case study has shown that explicit allocation and prefetching in a heterogeneous memory system can bring performance benefits. However, currently a lot of manual optimization is required. The simplified example in Figure 4b shows how the prefetching is added to the computation. The goal of our future work is to simplify the usage of this support.

Step 1: Use of pragmas to mark highly accessed objects
In a first step, we want to modify the code in a way that allows marking heavily accessed objects with pragmas, as shown in Figure 4a. The compiler then can automatically add instructions to enable allocation of objects in high-bandwidth memory and – if required – prefetching. At runtime, the underlying runtime system has to decide where to allocate an object and add prefetching if required.

Step 2: Automatic optimization with profiling
The *pragma* approach still requires manual instrumentation of the code. The programmer needs to know which objects are highly accessed and best allocated in MCDRAM/HBW memory. This information can be gathered from profiling tools. The profiling data can be used to automatically create code with optimal data distribution.

Oluwabamise T Oluwaseyi

HPC advancement to other fields

Deciding to attend school in the United States of America is a great opportunity and privilege to me mostly because of my field of specialization. For the past seven years, I have dreamt of studying computer, but I didn't know how extensive the field could be. Working with the SWOSU NASA research team has benefited me a lot; even as a freshman, I have come to know about High Performance Computing due to my work with them, and it's been great to know how versatile this field could be. I first was having difficulty trying to assimilate the definition of HPC on the internet until I found this layman's definition of High Performance Computing as any computational activity requiring more than a single computer to perform a task. Well, with further research, I realized that High Performing Computing is not limited to computer science or engineering alone; it is multifaceted. It has been used at the University of Pittsburgh for cancer research, for the estimation of global climate change, for institution technology development etc.

Then I asked myself this question: "why are there few women interested in this field?" In one of my readings, I found out that feeling isolated is a common frustration among women in technology especially when technology leaders are extremely against women for the few women in the field. Another interesting reason that I counted as false is because women have historically chosen Lower-Paying yet fulfilling jobs like teaching and journalism, whereas their male counterparts choose high-paying career like computer science and engineering.

I remember one time a friend asked me what my major was, and I told her I was studying computer science; then she looked at me in a way I couldn't understand and said, "You, computer science?" I said, "Yes, what's the problem with that?" She said, "You don't look like someone in that field of study," and I started wondering if those studying computer science have a special look or if I look too dumb to study computer science. Back at home also in Nigeria, people have tried to convince me to go for medicine instead of computer science, asking me if I have seen any girl who wants to go for computer science, but I tell them I am that girl.

Meeting Dr. Jeremy Evert also gave me the courage to stand firm on computer science because no matter what I do, he will always encourage me to do better. I have decided to make it a mandate to me not only to utilize and work with HPC in the United State, but also to take it over to Africa.

Oluwaseyi Oluwabamise is a freshman at Southwestern Oklahoma State University. She is a computer science major, and her minor is Entrepreneurship. She is working with the SWOSU NASA Research team this summer, even though she is a freshman and has not taken any computer science classes yet. After graduating from her secondary school in 2013, she went for a three-month computer training course, but didn't get enough training because she had to study for her SAT and TOEFL exam. She has been trying her best working with the team with the help of her professor, Dr. Jeremy Evert, and the rest of her colleagues who has given her full support in the work she is doing which has led her to know more about High Performing Computing.

On the Development of a Sustainable Curriculum for Undergraduate Research in High Performance Computing at Southwestern Oklahoma State University

Oluwaseyi Oluwabamise(oluwabamiseo@student.swosu.edu), Dr. Karen Sweeney, Dr. Amanda Evert, and Dr. Jeremy Evert

SWOSU Senior Computer Science senior Adriel Fillippini presenting NASA research project to Incoming Freshmen during a campus orientation tour. The goal of these activities is to allow the experienced upperclassmen to hand off their projects and share their excitement with students at the beginning of their program.

SWOSU students learn HPC as part of the Computer Science curriculum. The joy of HPC and the rewards of problem solving excite students about STEM related opportunities. Available curriculum materials for HPC are limited. Most of the text and literature has been generated by career academics with advanced degrees, and with a focus on instructing a population working on advanced degrees themselves. This type of reference does not always serve the needs of undergraduate students who have limited backgrounds in programming and their own area of study. To alleviate this problem, SWOSU students are heavily involved in curriculum development.

The process begins by congratulating the students on their expertise. They understand better than anyone in the room what they do and don't know. They are then provided tools to collaborate on the development of their own documentation. All work is completed within Github repositories, including programming source code, working notes in markdown, and final research chapters for publication in LaTeX. Students collaborate through Slack. Students use Scrum for project management.

This research was supported in part by an Oklahoma NASA EPSCoR Space Grant Consortium Research Initiation Grant. The federal grant/cooperative agreement number is NNX15AK42A.

Students were provided $36K in wages over the Summer of 2016 to begin work on their curriculum. They were asked to develop a validation study of previous findings by NASA Goddard Space Flight Center researcher, Dr. Charles Ichoku. Students were asked to verify the accuracy of NASA satellite data against measurements taken by ground based sensors measuring earth atmosphere aerosols. They documented what they needed to learn and how they learned it in order to complete the research.

Students were asked to contribute to the project any way that excited them. Criminal Justice Major Blessing Abiodun led the literature investigation. Charles Sleeper wrote a chapter on how to build a Raspberry Pi cluster. Computer Science Freshman Oluwaseyi Oluwabamise joined the team later in the summer and was asked to test and evaluate the curriculum that had been developed by other students. The thought was that if a college freshman, new to the team, could not follow the work, the work would not have value to future freshmen.

Dr. Henry J. Neeman, Assistant Vice President, Information and Technology, Research Strategic Advisor, and Director of the OU Supercomputing Center for Education and Research providing a HPC tour to six SWOSU students during the 2016 XSEDE Summer HPC Bootcamp. In the center is Charles Sleeper, a Native American Student, and second generation college student, plans to use tribal funds and his skills in HPC to get through graduate school. Behind Charels is Dean Phares, a USAF veteran and FAA Air Traffic Controller, who included HPC on his application for a National Defense contractor where he now works. To the right of Charles is Jordyn Hartzell, who in Spring 2016 led her team to a second place finish in the First Oklahoma HPC Competition.

Maria Andrea Pimiento Ojeda

Processing and Visualization in Embedded Architectures of High Performance Computing

Large data processing and detailed visualization of the results are often required in scientific applications, however, the quantity of data used in research applications is increasing exponentially, making these tasks slow and difficult to handle in personal computers.

An alternative to that situation is to use remote supercomputers, but in some cases they end up being underused, and they are also expensive to use, both on terms of energy efficiency and financial costs. Using embedded architectures suitable to high performance computing such as NVIDIA JETSON which is an embedded development platform from Nvidia; it features high-performance, low-energy computing and computer vision, with the advantage of having a small size and being portable.

This technology can be used to develop scientific applications for different purposes, and in different places, such as in a laboratory, in the countryside, in a vehicle, among others.

This poster will discuss a work in progress, integrating embedded systems, image processing, and high performance computing. The objective is to propose an embedded architecture for high performance computing for image processing and visualization, and then implement a prototype to speed up the image processing algorithm called Extended Depth of Field (EDF), using massively parallel processing on GPU with CUDA on a Nvidia Jetson TK1; this device will be coupled to a Scanning Electron Microscope, to process the acquired images.

Maria Andrea Pimiento is a senior of Computer Engineering at Universidad Industrial de Santander (UIS) in Colombia, and she did an academic exchange at Universidad Nacional Autónoma de México (UNAM) in Mexico with the Banco Santander scholarship. She is currently working in the High Performance and Scientific Computing Center (SC3-UIS) research group.

Her research interests include parallel programming, especifically massively parallel processing on GPU, and power optimization. Thus she is doing her graduation project titled "Processing and Visualization in Embedded Architectures of High Performance Computing" using CUDA on an embedded system.

Processing and Visualization in Embedded Architectures of High Performance Computing

Maria Andrea Pimiento Ojeda (ing.mariapimiento@gmail.com)
Advisors: C. J. Barrios Hernández (SC3-UIS, COL), A. Plata Gómez (GOTS-UIS, COL)
Mentor: S. Chandrasekaran (UDEL, USA)
1. High Performance and Scientific Computing Center (SC3-UIS)
2. Grupo de Investigación en Óptica y Tratamiento de Señales (GOTS-UIS)
Universidad Industrial de Santander (UIS)
Bucaramanga, Colombia

MOTIVATION

- The quantity of data used in research applications is increasing exponentially.
- Some embedded architectures feature high-performance, low-energy computing, and computer vision capability, for low price.
- This technology can be used to develop diverse applications such as: healthcare (in battlefields and/or hospitals), autonomous vehicles, etc.

SYSTEM COMPONENTS

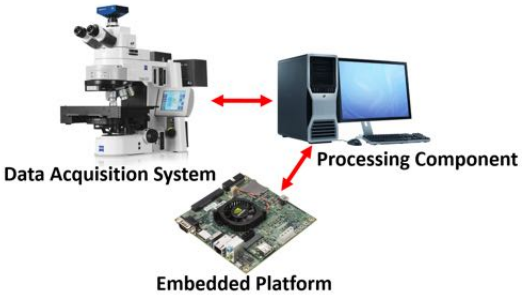


Figure 1. System Components

Description of the components of the proposed architecture and their implementation:

1. Processing System

1.1. Embedded platform

HPC development platform for massively parallel processing on GPU, with low power consumption and small size. Implemented using a NVIDIA Jetson TK1 which has the Tegra K1 SOC and the Linux4Tegra OS.

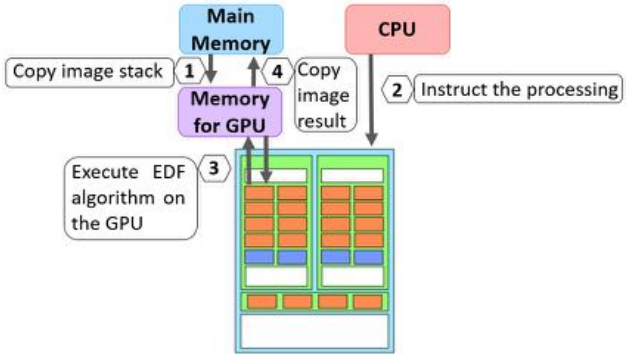


Figure 2. Execution Flow

1.2. Processing Component (Optional)

If the system has a dedicated processing component, the embedded platform will perform only the coprocessing. If not, the embedded platform will perform all the processing. Implemented using a Personal Computer with Windows XP Service Pack 2 and AxioVision Microscopy Software.

2. Visualization System

Software that allows 3D visualization and a coupled screen to the processing system. Implemented using ParaView Software and the PC screen.

3. Data Acquisition System

System responsible for acquiring the data and send them to the processing system, it can be of any type and include or not preprocessing. Implemented using an Electronic Microscope Carl Zeiss.

4. Algorithm

The implementation of the algorithm must be massively parallelizable. Implemented using the Extended Depth of Field (EDF) method which obtains a 3D image focused at all planes.

Extended Depth of Field (EDF) process:

- Obtain the position in which achieved focus for each of the images, and color intensity in that position.
- Make the topographical image and the focused image, respectively.
- Render the 3D image from the above images.

It will be implemented in CUDA with C language.

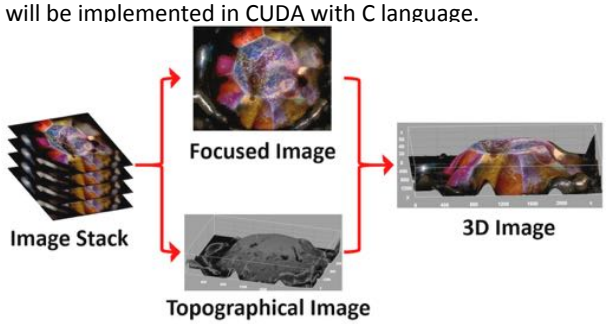


Figure 3. Extended Depth of Field (EDF) process

5. Communication

Communication protocol between system components. It has no specific requirements, but seeks to be as fast as possible. Implemented using a TCP Socket to connect the CUDA implementation of the EDF algorithm, with the image acquisition software AxioVision that handles the microscope from the PC.

FUTURE WORK

- Implement the proposed algorithm on NVIDIA Jetson TK1.
- Evaluate system performance: Energy efficiency and time costs.

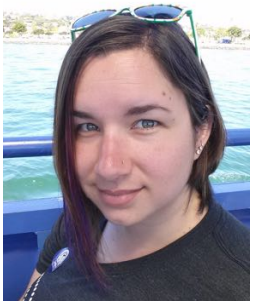
More: http://forge.sc3.uis.edu.co/redmine/projects/project-mpimiento
SC3-UIS: http://www.sc3.uis.edu.co/

REFERENCES

[1] Ben-Eliezer Eyal, Zalevsky Zeev, Marom Emanuel, Konforti Naim. All-Optical Extended Depth of Field, IOP Publishing Ltd 2003.
[2] Bischof, C., Brückner, M., Gibbon, P., Joubert, G.R., Lippert, T., Mohr, B. y Peters, F. Parallel Computing: Architectures, Algorithms, and Applications. IOS Press, 2008.
[3] Hernández, Mónica. Implementación Del Método De Profundidad De Campo Extendida En Arquitecturas Paralelas. Revista e-Colabora, 2012.
[4] Li, Qing y Yao, Caroline. Real-Time Concepts for Embedded Systems. Elsevier, 2003.
[5] Raj Kamal. Embedded systems: architecture, programming and desing. Tataq McGraw-Hill, 2007.

Caitlin Ross

Performance Analysis and Visualization of Dragonfly Network Simulations



Parallel discrete-event simulation (PDES) is an important tool in the co-design of extreme scale systems because PDES can provide a cost-effective way to evaluate designs of HPC systems. The CODES simulation framework provides various HPC network, workload and storage models and is built on top of the ROSS PDES framework. ROSS uses an optimistic event scheduling protocol, where events are processed without global synchronization of the processing elements. When an event is processed out of timestamp order, the system must be rolled back to a previous state so the events can be re-executed in the correct order. There are many factors that can affect the rollback behavior, and ultimately the performance, of a simulation, but this needs to be better studied in order to make simulations more efficient. We have instrumented ROSS to collect detailed data about the simulation engine in order to gain insight to the rollback behavior of our simulations. The instrumentation must be done carefully in order to minimize the perturbation of the simulation, since perturbing the simulation can drastically change the rollback behavior and introduce an additional data collection overhead. In this work, we use our instrumented ROSS discrete-event simulator to gain insight into the performance of a 5K-node dragonfly network topology model provided by the CODES simulation framework. We perform a scaling study that compares instrumented ROSS to non-instrumented ROSS in order to determine the amount of perturbation when running at different simulation scales. We also provide visualizations from the resulting data of running the dragonfly network simulation with instrumented ROSS.

Caitlin Ross is a third year PhD student in Computer Science at Rensselaer Polytechnic Institute. She received her B.S. in Computer Science from The University of North Carolina at Greensboro in 2014. Her current research is in performance analysis of parallel discrete-event simulations. Caitlin also participates in activities that focus on diversity in Computer Science. She is the Vice Chair of RPI's ACM-W chapter and has helped to organize various events for students, including a women-focused hackathon

Performance Analysis and Visualization of Dragonfly Network Simulations

Caitlin Ross
Christopher D Carothers
Rensselaer Polytechnic Institute

Misbah Mubarak
Robert Ross
Argonne National Laboratory

Jianping Kelvin Li
Kwan-Liu Ma
University of California, Davis

I. Introduction

- ROSS: optimistic parallel discrete-event simulation (PDES) framework [1]
- CODES: built on ROSS and provides models for HPC network topologies, storage systems, and workloads [2]
- In optimistic PDES, each logical process (LP) processes events without frequent global synchronization among the processing elements (PEs)
  - Causality violations must be rolled back
  - Rollbacks happen on kernel process (KP) basis
  - Global synch happens by the global virtual time (GVT) computation; lowest timestamp of unprocessed events
- Goal: To better understand rollback behavior and its effect on performance of the simulation engine

II. ROSS Instrumentation

- 3 modes of instrumentation:
  - GVT-based: sample metrics immediately after GVT computation
  - Real time sampling: sample metrics at user specified real time intervals
  - Event tracing: collect data about each event
- Can choose to collect data at PE, KP, or LP granularities for GVT-based and real time sampling
- Store sample data in buffer, dump to file just after GVT computation if buffer almost full

III. Experiments

- All simulation runs performed with CODES dragonfly network topology model [3]
  - 6 terminals per router (876 total)
  - 12 routers per group (5,256 total)
  - 6 global channels
  - Synthetic workload with uniform random traffic, adaptive routing
- Sims performed on following systems:
  - Mastiff: SMP machine with 4 AMD procs w/ 16 cores each
  - Blue Gene/Q
- Parameters:
  - PEs/MPI ranks: 16, 32, 64
  - Num KPs: 16, 64, 246
  - Batch: 2, 8, 16
  - GVT-interval: 2, 8, 16, 32, 64, 256

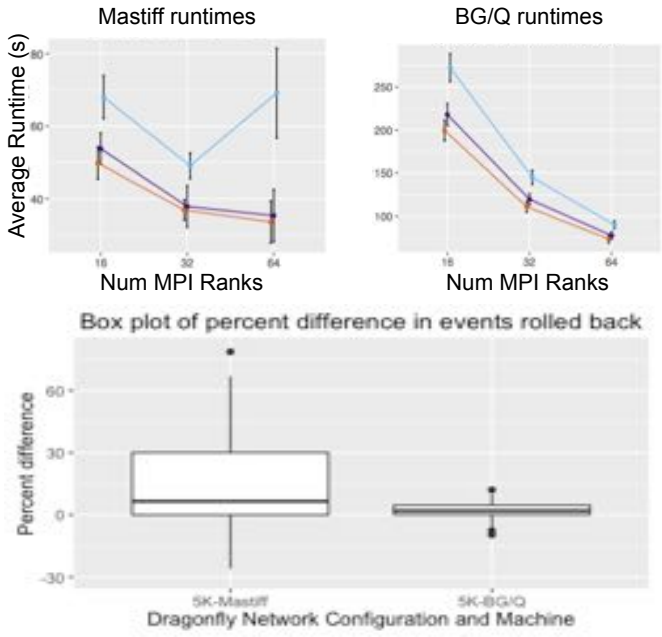
References

[1] ROSS: Rensselaer's Optimistic Simulation System. http://github.com/carothersc/ROSS
[2] CODES: Enabling co-design of Exascale Storage Architectures and distributed data-intensive science facilities, http://www.mcs.anl.gov/research/projects/codes/
[3] M. Mubarak, C. D. Carothers, R. Ross, and P. Carns, "Modeling a million-node dragonfly network using massively parallel discrete-event simulation," in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*, 2012, pp. 366–376.
[4] C. Ross, C.D. Carothers, M. Mubarak, P. Carns, R. Ross, J.K. Li, K. Ma, "Visual data-analytics of large-scale parallel discrete-event simulations," in *Proceedings of the 7th International Workshop on Performance Modeling, Benchmarking, and Simulation of High Performance Computing Systems*, 2016.

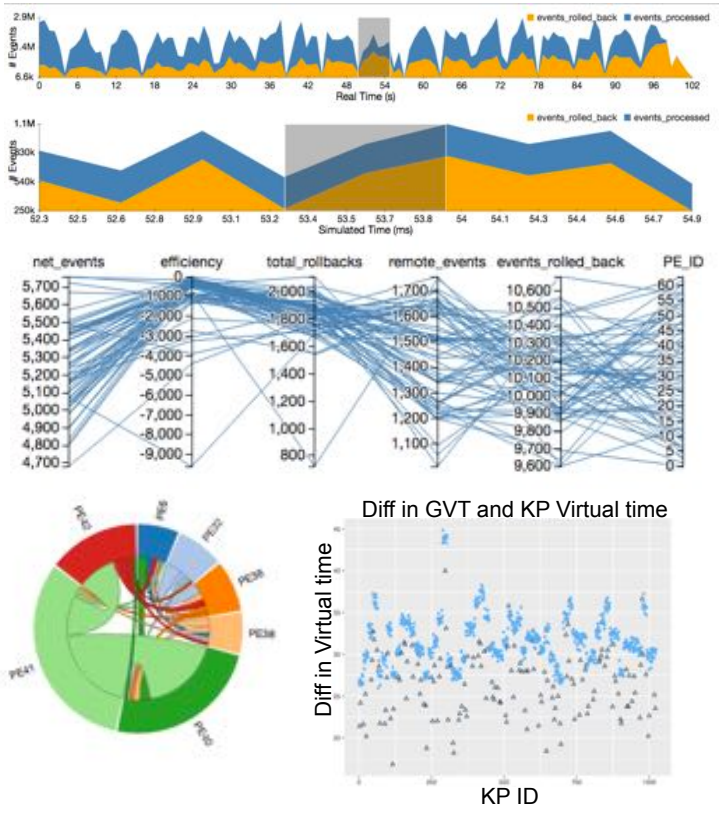
This material was based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computer Research (ASCR), under contract DE- AC02-06CH11357 and DE-SC0014917.

Rensselaer Argonne NATIONAL LABORATORY UCDAVIS

IV. Perturbation Analysis



V. Performance Visualizations



VI. Conclusion & Future Work

- Visual analysis tool allows for viewing data at different granularities and points in time
- Able to assess communication patterns of PEs
- Perturbation of sims on BG/Q was within reasonable threshold in most cases
- Future work:
  - Examine larger-scale models and models with poor rollback efficiency
  - Data reduction, especially for event tracing
  - Further refine visualization tool

Louise Spellacy

Partial Inverses of Block Tridiagonal Non-Hermitian Matrices

The SMEAGOL electronic code uses a combination of density function theory (DFT) and Non-Equilibrium Green's Functions (NEGF) to study nanoscale electronic transport under the effect of an applied bias potential [1]. Inversion of a block tridiagonal non-Hermitian matrix is required to obtain the Green's function used by the SMEAGOL code. In many cases, only the block tridiagonal part of the inverse is needed. Currently the SMEAGOL code is limited by single node, multicore matrix inverses. The addition of parallel sparse matrix inverse functionality will allow significantly larger systems to be addressed.

The algorithm presented here is an extension of a previous work where a method for parallel inversion of Hermitian block tridiagonal matrices is detailed [2] and [3]. This method extends [2] and [3] to the non-Hermitian case. The tridiagonal blocks of the matrix are evenly distributed across p processes. The local blocks are used to form a "super matrix" on each process. These matrices are inverted locally and the local inverses are combined in a pairwise manner. There are log(p) combination steps. At each combination step, the updates to the global inverse are represented by updating twenty "matrix maps" on each process. The matrix maps are finally applied to the original local blocks to retrieve the block tridiagonal elements of the inverse. This extended algorithm requires the computation and communication of a greater number of matrix maps than the algorithm detailed in [3].

The pairwise algorithm has been implemented as a standalone program, written in Fortran and MPI. It has been tested on local clusters in the Trinity Centre for High Performance Computing. Comparisons with existing codes, MUMPS and ScaLAPACK, will be given.

1. SMEAGOL: Non-equilibrium Electronic Transport www.smeagol.tcd.ie
2. S. Cauley, M. Luisier, V. Balakrishnan, G. Klimeck, and C.-K. Koh. Distributed non-equilibrium Green's function algorithms for the simulation of nanoelectronic devices with scattering. Journal of Applied Physics, 110(4), 2011.
3. S. Cauley, J. Jain, C.-K. Koh, and V. Balakrishnan. A scalable distributed method for quantum-scale device simulation. Journal of Applied Physics, 101(12):123715, 2007.

Louise Spellacy has been at Trinity College Dublin since 2010. She is currently a research assistant in the Trinity Centre for High Performance Computing. She was previously a student, receiving a M.Sc. in High Performance Computing with distinction in 2014 and a B.A. in Mathematics in 2013. Her interests include parallel linear algebra algorithms and their implementation using MPI, and the profiling of parallel applications. Parallel profiling of MPI applications was the topic of her M.Sc. thesis. In addition to her research, she prepares and delivers courses on numerical computing and parallel tools.

Partial Inverses of Block Tridiagonal Non-Hermitian Matrices

Louise Spellacy[1] (spellal@tcd.ie) and Darach Golden[2]
[1]School of Mathematics, Trinity College, Dublin
[2]Research IT, Trinity College, Dublin

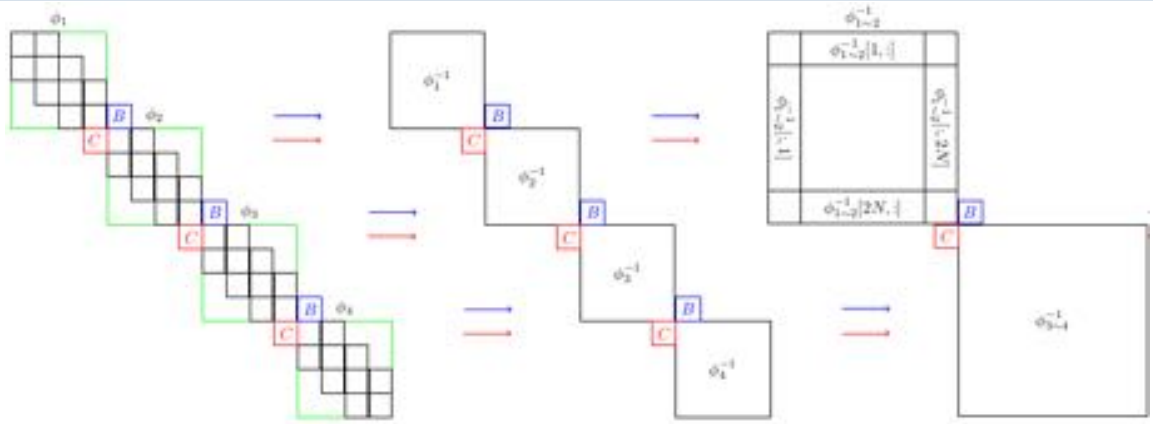


Figure 1: Pairwise Combination Steps on Four Processors

The Problem

- Square diagonal sub-blocks with complex elements, $A_i \in \mathbb{C}^{N_i \times N_i}$.
- Upper and lower sub-blocks, $B_i \in \mathbb{C}^{N_i \times N_{i+1}}$ and $C_i \in \mathbb{C}^{N_{i+1} \times N_i}$, respectively are complex.
- Matrix K is complex and square, $K \in \mathbb{C}^{N_B \times N_B}$, where $N_B = \sum_{i=1}^{b} N_i$.
- b is the number of diagonal sub-blocks.
- No assumption that the sub-blocks are Hermitian.

$$K = \begin{pmatrix} A_1 & -B_1 & & & & \\ -C_1 & A_2 & -B_2 & & & \\ & -C_2 & A_3 & -B_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -C_{b-2} & A_{b-1} & -B_{b-1} \\ & & & & -C_{b-1} & A_b \end{pmatrix}$$

Figure 2: The Problem Matrix

The Method

Pairwise algorithm (in yellow) is an extension of previous work where a method for parallel inversion of Hermitian block tridiagonal is detailed.

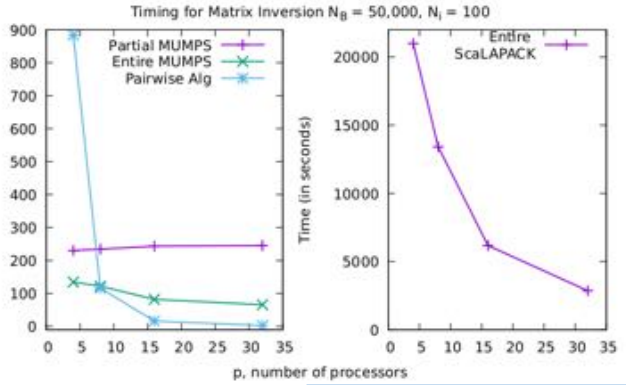
* $3b-2$ sub-blocks of K are evenly distributed across p processors.
* Block tridiagonal non-Hermitian super-matrix $\phi_i, 1 \leq i \leq p$, created from local sub-blocks on each processor.
* Remaining "bridge matrices", B_i and C_i, are made available on each processor.
* Inverse ${\phi_i}^{-1}$ of each super-matrix is calculated on each processor using standard inverse method (e.g. LAPACK).
* Pairwise combination steps begin concurrently.
* Two super-matrices ${\phi_i}^{-1}$ and ${\phi_m}^{-1}$ are combined, using bridge matrices B_i and C_i, to create ${\phi_{i\sim m}}^{-1}$.

- Generating the entire ${\phi_{i\sim m}}^{-1}$ at each step is memory and computationally expensive.
- Subsets of ${\phi_{i\sim m}}^{-1}$ (referred to as matrix maps) are stored instead, recording changes to the inverse at each combination step.

Time Complexity $\mathcal{O}\left((N_B/p)^3 + N_i \log p + ({N_i}^3 b)/p\right)$

Memory Complexity $\mathcal{O}\left((N_B/p)^2 + {N_i}^2 + ({N_i}^2 b)/p\right)$

Results and Conclusions



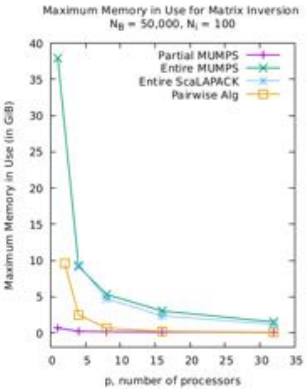


Figure 3: Time and Memory Consumption Results for $N_B = 50{,}000$ and $N_i = 100$.

- Pairwise algorithm is faster than partial MUMPS when number of processors is increased.
- Pairwise algorithm has similar memory usage to the partial MUMPS.
- Entire (Dense) ScaLAPACK and MUMPS have similar memory usage but MUMPS is faster than ScaLAPACK.
- Further investigation is required regarding changes in accuracy when sub-block sizes and number of processors increase.

P	MUMPS Vs Sca	MUMPS Vs Pair	Sca vs Pair
4	$10^{-11}/10^{-10}$	$10^{-10}/10^{-09}$	$10^{-10}/10^{-09}$
8	$10^{-12}/10^{-10}$	$10^{-10}/10^{-08}$	$10^{-10}/10^{-08}$
16	$10^{-11}/10^{-10}$	$10^{-09}/10^{-07}$	$10^{-09}/10^{-07}$
32	$10^{-12}/10^{-10}$	$10^{-08}/10^{-06}$	$10^{-08}/10^{-06}$

Table 1: Absolute and Relative Difference for $N_B = 50{,}000,\ N_i = 1000$

Sangeetha Banavathi Srinivasa

Smart Load Balancing of File Systems in HPC clusters

The load imbalance in Lustre file system comes mainly from the bursty I/O patterns of scientific applications and the complex structure of the storage system. This complex structure is due to the hierarchical nature of the system consisting of Metadata Server (MDS), Metadata Target (MDT), Lustre Networking (LNET), Object Storage Servers (OSS), and Object Storage Targets (OST). The current solution to load balance the system utilizes the access frequency of every layer in the storage system and selects the OST which lies in the least access frequency path. This is inefficient because of the lack of effective global optimization. Our approach to solve load imbalance in Lustre file system is to have a global mapper on MDS which gives the clients a list of OSTs based on the runtime statistics collected from OST, OSS, MDT, MDS and LNET. The global mapper will have an AI component that will help in learning from the statistics collected. This gives a global view of the complex system and helps in making an informed decision which solves the problem of load imbalance. We are evaluating our solution on an extreme-scale compute cluster, Titan, at the OakRidge Leadership Computing Facility (OCLF).

Sangeetha B. Srinivasa is a Graduate student at Virginia Tech(VT), Blacksburg, USA, as well as a Graduate Research Assistant(GRA) at Advanced Research Computing(ARC). She is pursuing research in HPC, Cloud Computing and Big Data. She is currently working on projects in collaboration with OakRidge National Lab, TN, USA and IBM, Ireland. As a GRA she assists students, faculty and staff in troubleshooting issues when using HPC resources of ARC. ARC provides advanced computational systems, large-scale data storage, visualization facilities, and software to support research in VT. As part of this endeavor she works on packaging, testing and deploying necessary software packages.

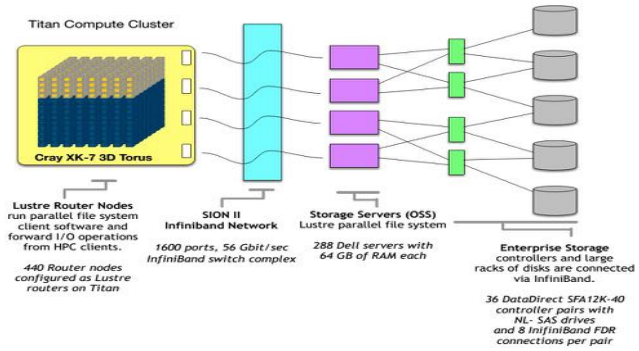
Prior to joining VT, Sangeetha has worked at Cisco Systems in India as a Software Engineer for two years. She was part of the Mobile Internet Technology Group (MITG). As part of this team she developed and helped maintain the 'Star-OS' infrastructure component NPU(Network Processors) drivers & microcode.

On Data-Driven I/O Load Balancing in Extreme-Scale Storage Systems

Sangeetha B. Srinivasa*, Arnab K Paul*, Arpit Goyal*, Feiyi Wang, Sarp Oral, Ali R Butt*
{bsangee, akpaul, arpitg, butta}@vt.edu, {fwang2, oralhs}@ornl.gov
Virginia Tech*, Oak Ridge National Laboratory

1. GOALS

- Distributed Monitoring Tool to collect File creation statistics from 18688 Titan Cluster nodes.
- Use AI for new file placement on Object Storage Targets(OSTs) to manage load balancing.

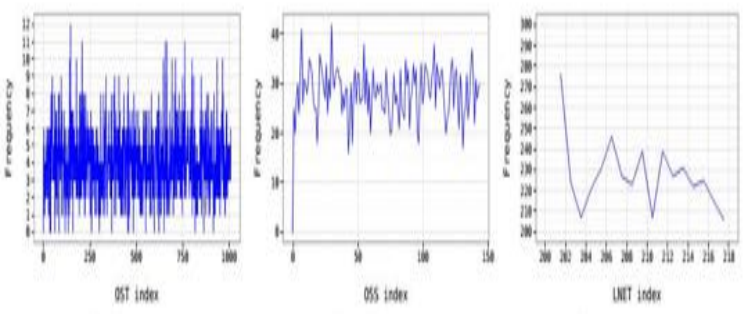


2. MOTIVATION

- Increasing number of scientific applications have bursty I/O patterns.
- The current system makes use of traditional Capacity based Round Robin approach and there is no Load Balancing mechanism as such.

3. PROBLEM

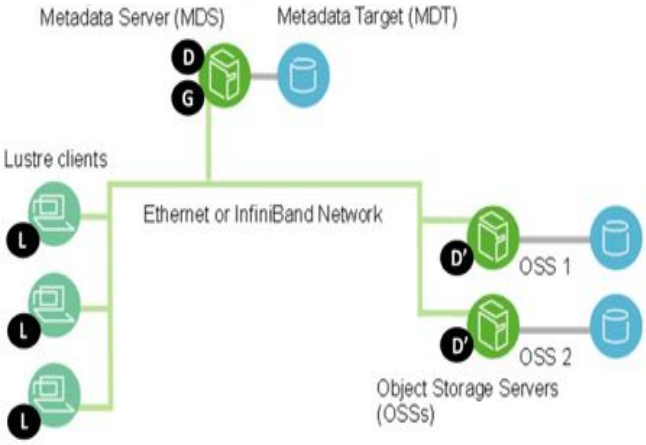
- Same set of OSTs are being used for placement of newly created files.
- Variability in terms of applications being run by clients and variability in OSTs.



(a) Default OST Placement (b) Default OSS Placement (c) Default LNET Placement

4. SOLUTION

Distributed Monitoring Tool
Collect runtime statistics from OST, OSS, MDS, MDT and LNET.



5. ONGOING WORK – LOAD BALANCING

The following approaches are being evaluated in order to come up with an optimal Load Balancing Model.

- **Hidden Markov Model**
  - Model the current behavior and predict the future usage of OSTs.
- **Linear Programming Model**
  - Ascertain the relationship between the various parameters collected by the Distributed Monitoring Tool and formulate a LP model.
- **Reinforcement Learning**
  - The model evolves with changing input requests based on a score function.

6. EVALUATION

- Simulator to model current Lustre behavior.
- Integration of our Library with MDS on Titan.

7. ACKNOWLEDGEMENT

This work is sponsored in part by the NSF under the grants: CNS-1405697, CNS-1422788, and CNS-1615411.

Daria Tarasova

Algorithm Development for Cloud-Based Quantitative Histological Image Analysis Tool

Histological evaluation of tissue sections is a key step in chemical and drug development and safety testing to assess potential adverse health effects in humans. These evaluations are performed by highly trained pathologist, typically through qualitative or semi-quantitative methods. The manual quantitation of features of interest is a difficult and time-consuming process subject to inter- and intra-observer variability. The Quantitative Histological Analysis Tool (QuHAnT) is a computational approach to quantitate histological features of interest in a high-throughput manner, with the goal of helping pathologists provide more accurate evaluations, more quickly. This project describes the development of the image analytics code (initially created in Matlab), and the optimization and integration of algorithms into Python and C++ for use on the high performance computing systems provided by Joyent. Due to the large amount of high resolution images pathologists analyze, the tool was structured to be pleasantly parallel in order to provide immediate results to the user. Joyent was used to leverage its ability to interact between the image analytics code in C++ and the user interface system in Node.js. With the analytics operating on one image as its input, the cloud-based system performs the analytics on multiple images, decreasing time spent per image and overall. *Funded by the Michigan Translational Research and Commercialization Program supported by the State of Michigan 21st Century Jobs Fund received through the Michigan Strategic Fund and administered by the Michigan Economic Development Corporation*

Daria Tarasova grew up in Holt, Michigan and is currently studying at Michigan State University to obtain her undergraduate degree in Computer Science with a specialization in Chemistry by December 2017. Working in the labs of Drs. Tim Zacharewski and Dirk Colbry, she focused on development of a digital pathology tool operating on a cloud computing system. Her research interests include computer vision and cloud computing. Prior to research, Daria was involved with Information Technology Empowerment Centers' 2020 Girls Program to encourage middle school girls to get involved in STEM activities by planning educational activities teaching basic engineering and computer science principles.

ALGORITHM DEVELOPMENT FOR HIGH THROUGHPUT QUANTITATIVE HISTOLOGICAL IMAGE ANALYSIS

Daria Tarasova[1], Rance Nault[2,4], Dirk Colbry[3], and Tim Zacharewski[2,4]

College of [1]Engineering, Departments of [2]Biochemistry & Molecular Biology, and [3]Computational Mathematics, Science and Engineering,
[4]Institute for Integrative Toxicology, Michigan State University

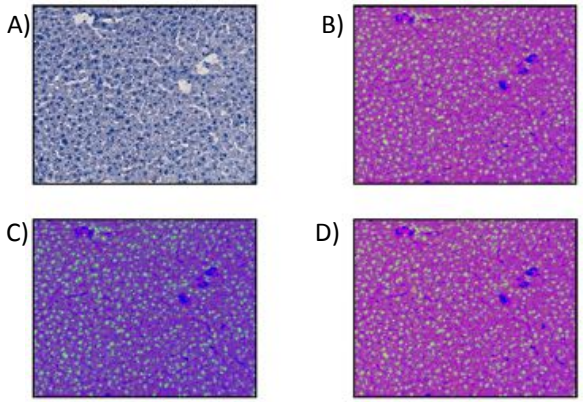
MICHIGAN STATE UNIVERSITY
Institute for Integrative Toxicology

INTRODUCTION AND OBJECTIVES

To assist pathologists with the quantitation of histological features, QuHAnT, a high throughput image analysis software was developed in Matlab to automate the process and improve reliability and reproducibility. However, using Matlab would have made it difficult to use QuHAnT on the cloud, and therefore was translated into C++/OpenCV to generate equivalent results as the original software was validated to the gold standard of the field. The C++/OpenCV version of QuHAnT consisted of 3 parts: (1) converting the input RGB image to an HSV color space, (2) threshold it to segment the features from the background and (3) calculate the area, centroid, and bounding box of each component.

The OpenCV functions that corresponded with these tasks did not satisfy these requirements, requiring the development of modified functions. Moreover to process and store thousands of images in a reasonable time-frame, the software was developed for deployment on cloud computing resources. The methods for algorithm development and testing on Joyent®-Manta cloud resources are shown below.

RGB TO HSV CONVERSION



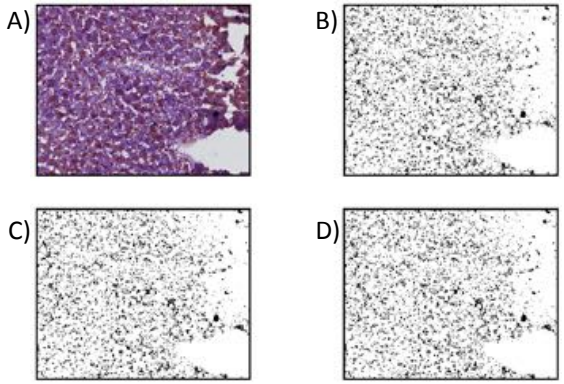
▲ **FIGURE 2. DIFFERENCES IN RGB-HSV CONVERSIONS**

A sample image (A) was converted from RGB color scale to the HSV color scale using the Matlab ***rgb2hsv*** function (B), OpenCV ***cvtColor*** function (C), and in-house converted QuHAnT algorithm (D). The converted algorithm for QuHAnT matched the Matlab algorithm while OpenCV did not match both visually and quantitatively.

Supported by the Michigan Translational Research and Commercialization Program (MTRAC)

Email: tarasov1@msu.edu
http://dbzach.fst.msu.edu

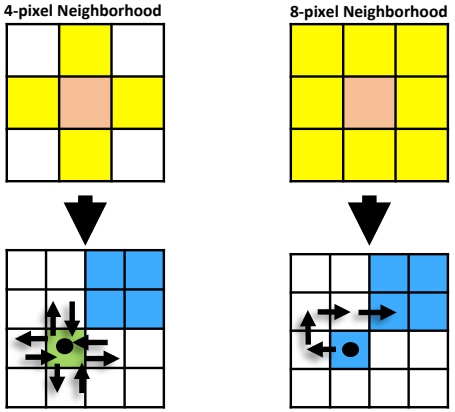
THRESHOLDING (FEATURE DETECTION) ALGORITHM



▲**FIGURE 3. DIFFERENCE IN THRESHOLDING ALGORITHM**

A sample image (A) was segmented into a binary image using the Matlab version of the code (B), OpenCV ***inRange*** function (C), and the C++ developed version of QuHAnT (D). The more specialized approach was needed for efficiency and to minimize the odds for inaccuracy as the images vary.

FEATURE IDENTIFICATION & QUANTITATION ALGORITHM



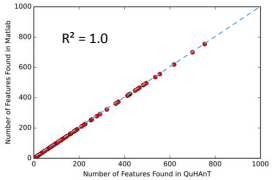
▲ **FIGURE 4. DIFFERENCES IN NEIGHBORHOODS**

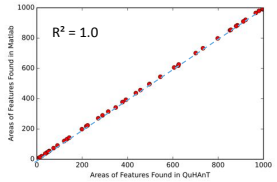
OpenCV uses a 4-neighbourhood approach while Matlab uses an 8-neighbourhood approach. This results in the consideration of the yellow pixels labeled around the current pixel being evaluated (orange). This results in considering each region of pixels as separate (green and blue) or a single connected region (blue).

VALIDATION OF QuHAnT C++ ALGORITHMS

▸ **FIGURE 5. FEATURES NUMBER IDENTIFICATION**

Correlation between the number of individual features identified in 133 sample images determined with validated Matlab and QuHAnT OpenCV/C++ algorithms. An ideal feature comparison would represent a 1 to 1 relationship indicating a perfect correlation between the validated and developed QuHAnT.

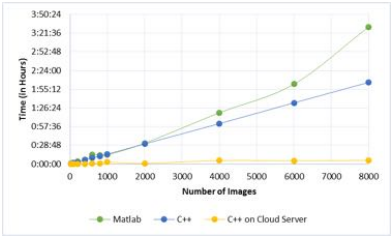




◂ **FIGURE 6. FEATURE AREA QUANTITATION**

Correlation between the total feature area identified in 133 sample images determined with validated Matlab and QuHAnT OpenCV/C++ algorithms

CLOUD PERFORMANCE



▲ **FIGURE 7. DIFFERENCE IN PERFORMANCE OF ALGORITHMS**

The throughput of QuHAnT analytic algorithms in Matlab on the Michigan State High Performance Computer, C++ on a local instance of Joyent, or distributed on Joyent's cloud resources were compared. The data shows that the C++ algorithm performed slightly better than Matlab. However, deploying the software in a pleasantly parallel manner dramatically reduced analysis time.

CONCLUSION

- The QuHAnT software was converted from Matlab to OpenCV/C++ which added flexibility and efficiency.
- Leveraging cloud resources has a much greater impact than refining the software and algorithms.

Jesmin Jahan Tithi

Cache-oblivious wavefront algorithms for dynamic programming problems: efficient scheduling with optimal cache performance and high parallelism

Wavefront algorithms are algorithms on grids where execution proceeds in a wavefront manner from the start to the end of the execution. Iterative-wavefront algorithms for evaluating dynamic programming (DP) recurrences exploit optimal parallelism but show poor cache-performance. Tiled (or blocked-loop) iterative wavefront algorithms achieve optimal cache-complexity and high parallelism but are cache-aware, and neither portable nor cache-adaptive. In contrast, standard cache-oblivious recursive divide-and-conquer (CORDAC) algorithms have optimal serial-cache-complexity but often have low parallelism due to artificial dependencies among subtasks. The cache-oblivious wavefront algorithms for DP problems are variants of CORDAC algorithms with reduced or no artificial-dependencies and hence, have better parallelism than the standard CORDAC algorithm.

We show how to transform a CORDAC algorithm to a recursive-wavefront algorithm to achieve optimal parallel-cache-complexity and high parallelism under state-of-the-art schedulers for fork-join programs (e.g., cilk™'s work-stealing scheduler). These cache-oblivious wavefront algorithms use closed-form formulas to compute at what execution time step each divide-and-conquer function/task must be launched in order to achieve high parallelism without losing in cache-performance. We present experimental performance and scalability results showing superiority of these new algorithms on standard multicore and manycore architectures compared to the standard CORDAC algorithms.

Jesmin Jahan Tithi is an HPC Software Architect at the Platform Architecture and Performance Team at Intel Corporation. Jesmin completed her Ph.D. on "Engineering high-performance parallel algorithms with applications to bioinformatics" at Stony Brook University, New York, USA in 2015 and Bachelors in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in 2009. Jesmin served as a lecturer at the Bangladesh University of Engineering and Technology which is the top engineering university in Bangladesh. During her Ph.D. research, she did successful internships at Intel Corporation, Google Inc., and Pacific Northwest National Laboratory. Her research interests are algorithm engineering, high-performance computing, heterogeneous programming models, bioinformatics, and machine learning algorithms.

Cache-oblivious wavefront algorithms for Dynamic Programming problems: probably efficient scheduling with optimal cache performance and high parallelism

Jesmin Jahan Tithi ¶ ¥, Pramod Ganapathi¶, Rezaul Chowdhury¶ and Yuan Tang*

¥Intel Corporation, ¶Dept. of Computer Science, Stony Brook University Stony Brook, New York, *School of Computer Science & Shanghai Key Laboratory of Intelligent Information Processing, Fudan University

What is Cache-oblivious Wavefront?

❑**Cache-oblivious Wavefront (Wave):** Variant of cache-oblivious recursive divide-and-conquer **(CORDAC)** with

- **Reduced/no artificial dependency** among subtasks
- Often with asymptotically **better parallelism**
- Do not use cache-parameter in algorithmic description **[unlike tiling/blocking]**
- Often **cache-efficient**

❑**Executes as if a wavefront is moving:** cells on grids are updated in a wavefront fashion

- **Examples:** stencil computations, **dynamic programming (DP)** algorithms

Benefits of Cache-oblivious Wavefront

Wave algorithms have order of magnitude better parallelism compared to the CORDAC algorithms

Algorithm	Longest common subsequence	Parathesis/ Matrix Chain Multiplication	Floyd-Warshall's All pair shortest path
Wave	510	1916.0	1404
CORDAC	18	23	148

Table 1: *Parallelism in Wave and CORDAC algorithms reported by cilkview™ scalability analyzer. Numbers show till how many cores a program should scale*

Key Results

- **On Multicores (16-24 core Xeon):** Wave algorithms are around 2× faster than CORDAC
- **On Manycores (287 core Knights Landing/KNL):** Wave algorithms are around 4-6× faster than CORDAC

Major contributions

Shows how to systematically transform CORDAC to COW

- Keep structure similar to the CORDAC for cache optimality
- Use analytically computed timing function to detect task readiness
- Can be scheduled using standard fork-join and a specialized hint-accepting scheduler
- No atomic-instructions/locks

Our algorithmic approach eliminates shortcomings of prior cache-oblivious wavefront (PPoPP 2015) algorithms

Example: in theory Wave has better span than CORDAC

LCS/Edit Distance	Span (lower is better)	Cache complexity
Cache-oblivious wavefront	$O(n \log n)$ optimal	$O\left(\frac{n^2}{BM}\right)$ optimal
Recursive divide-and-conquer (CORDAC)	$O(n^{\log_2 3})$	$O\left(\frac{n^2}{BM}\right)$ optimal

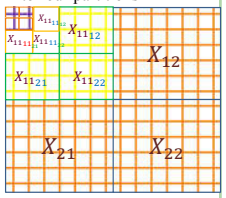
Table2: *Theoretical cache complexity and span of CORDAC vs Wave. B = block transfer size, M = cache size, n = input size,*

Recursive divide and conquer (CORDAC)

❑ Divide the grid/matrix recursively into four partitions

❑ Solve each partition recursively respecting the **dependency among partitions (i.e., cells on grid)**

❑ Keep dividing until each partition becomes small enough



CORDAC on grid/DP table

❑ Solve these small basecases iteratively

An example wavefront DP algorithm: Edit Distance

Edit Distance: Find minimum number of edits to convert string S to string T using Substitution, Delete, Insert edit operations.

In edit distance (ref: Introductions to Algorithm from MIT Press) value on cell (i, j) on a 2D grid depends on cell (i-1, j-1), (i, j-1) and (i-1, j) and string character at S[i] and T[j].

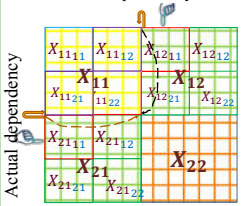
Represents a class of other problems: Longest Common Subsequence, SmithWaterman, and others

Source of sub-optimal parallelism in CORDAC

❑**Artificial dependencies among tasks** at several granularities

❑**Artificial dependencies increase the span** (critical path length of DAG), and **reduce parallelism**

Actual dependency



Edit distance

Execution order of **edit distance**:

X_{11}

$X_{12} \,||\, X_{21}$ *(parallel)*

X_{22}

Execution does not move to X_{12} or X_{21} until done with X_{11}

But $X_{12_{11}}$, $X_{21_{11}}$ can start in parallel with $X_{11_{22}}$

Artificial dependency Actual dependency

Cache-oblivious recursive wavefront technique removes artificial dependencies in CORDAC

By executing a task as soon as all its actual dependencies are fulfilled **Cache-oblivious wavefront algorithms (formerly named "COW") eliminate/reduce artificial dependencies.**

Prior work: COW algorithms, first proposed in PPoPP2015 [Tang, You, Kan, Tithi, Ganapathi, Chowdhury]) were **complicated to develop, analyze, implement, and generalize.**

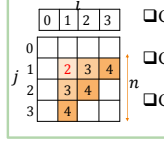
Systematic way to transform CORDAC to COW

Step 1: Construct completion-time function

It is the latest time when a cell gets updated/written

- A closed-form formula that gives the timestep at which each DP grid cell is fully updated in wavefront order – an order in which cells are updated in the fastest wavefront algorithm
- Max of (completion time of all input cells the cell depends on) + # input cells with that max time + 1

Example (Edit Distance):



❑Cell with index (i, j) lies on diagonal $(i + j)$

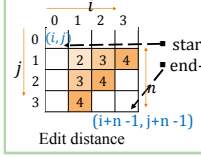
❑Cell (i, j) is fully updated at time step $(i + j)$

❑Completion-time, $C(i, j) = (i + j)$

Step 2: Construct start- and end-time function for each recursive function type in CORDAC

- **Start time:** minimum (start time of all sub-functions called) + (wait time to avoid race, if any)
- **End time:** maximum (end time of all sub-functions called) + (wait time to avoid race, if any)
- depend on the function type and input and output parameters

Example: A region with top-left corner at (i, j) and dimension n



Edit distance

Step 3: Derive the recursive wavefront algorithm

- Augment each function in CORDAC algo. to accept a timestep parameter w
- spawn all functions in parallel provided for whom start-time ≤ w ≤ end-time, remove all serialization in between
- Each functions returns smallest timestep > w, for which it has some update to be applied -- used to find next value of w
- Loop through all timesteps (w) in increasing wavefront order

Original CORDAC algorithm for Edit Distance

```
CORDAC(X, n) {
if(n<=switching_point) Iterative(X, n);
else{
  nn = n / 2;
  CORDAC (X11, nn);
  spawn CORDAC (X12, nn); CORDAC (X21, nn);
  sync; CORDAC (X22, nn); }
}
```

The transformed recursive wavefront code

```
CORDAC(X, n, w) {
if(n<=switching_point){
  if(S_A(X, n) = w) Iterative(X, n); return E_A(X, n);
}
else {
  nn = n /2;
  if(w<S_A(X11, nn))      w1 = S_A(X11, n)
  else if(w<=E_A(X11, nn))w1 = spawn CORDAC(X11,nn, w);

  if(w<S_A(X12, nn))      w2 = S_A(X12, nn)
  else if(w<=E_A(X12, nn))w2 = spawn CORDAC(X12,nn, w);

  if(w<S_A(X21, nn))      w3 = S_A(X21, nn)
  else if(w<=E_A(X21, nn))w3 = spawn CORDAC(X21,nn, w);

  if(w<S_A(X22, nn))      w4 = S_A(X22, nn)
  else if(w<=E_A(X22, nn))w4 = spawn CORDAC(X22,nn, w);

  sync;
  return min(w1, w2, w3, w4);//returns min(w) > the input w
   }
}
```

```
RecursiveWavefront(X, i, j, n) {
w = 0; max_completion_time = C(i+n-1, i+n-1);
while (w < max_completion_time)
      w = CORDAC (X, i, j, n, w);
}
```

Experimental Results

Generated algos for four dynamic programming problems

- Longest common subsequence (LCS) / Edit distance
- Parenthesis problem (Matrix chain multiplication)
- Floyd-Warshall's all pairs shortest paths (FW-APSP)
- Sequence alignment with general gap penalty (Gap problem)

Projected parallelism by CilkView™:

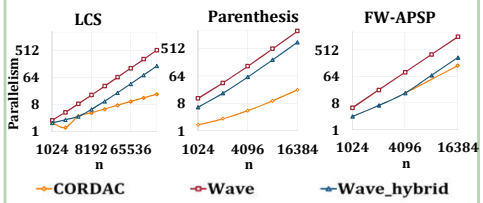


Figure: *In this figure, Wave means recursive wavefront with pure iterative kernel after switching point, wave-hybrid means recursive wavefront with standard CORDAC algorithm after switching point*

Speedup on 24-core Haswell and 71-core KNL compared to standard CORDAC

Algorithm	LCS	Parenthesis	FW APSP
wave	2×, 6×	2.6×, 4×	1.5×, 2×
wave-hybrid	2×	1.9×,	1.1x
COW (PPoPP'15)	1.9×	0.9	1.0

Table3: *Speedup achieved by wave wrt CORDAC on 24 core Haswell and KNL (note: these code are neither vectorized, nor hand optimized)*

Mariam Umar

An Application and Hardware Driven Co-design for Current and Future Architectures Using Domain Specific Language

As we approach the exascale era, manual hardware-software co-design is becoming a challenge particularly because of increases in complexity and scale. Solutions such as memory throttling, DVFS alone are not as impactful and it is becoming difficult to extract similar gains as we saw in the past. We expect these challenges to exacerbate particularly when we consider the variety of applications and application-specific optimizations. We, therefore propose automation of hardware-software co-design as a suitable path towards obtaining faster results. Our proposal includes a runtime framework for the application and underlying machine, that uses Aspen, a domain specific language for analytical performance modeling. The framework automatically extracts information at runtime from the application and hardware, formats them in accordance with Aspen's grammar, tests and verifies the output, and amalgamates them to generate a hardware-software co-design recommendation. This framework has numerous applications, including testing of an application and machine combination before deployment on a supercomputer and obtaining a recommendation for near-optimal configuration. The proposals include sensitivity of the application with respect to hardware configurations and vice versa. In this poster, we summarize such scenarios. We also show an emulated version of an exascale system in order to verify the applicability of the framework for future architectures. In future, we plan to use Aspen and extend this work by implementing and experimenting with other memory schemes.

Mariam is a Ph.D. student and a member of the SCAPE laboratory (scape.cs.vt.edu) in the Department of Computer Science at Virginia Tech, where her assistantship is funded through NSF grants. Her research interests include analytical and empirical models for performance and power-consumption for (current and future) high-performance-computing (HPC) systems. She also has experience in developing digital-signal-processing methods for embedded systems and models for routing and channel optimization for wireless networks. In future, she plans to investigate power-aware architectures for exascale systems and be involved with efforts that meet the goals set by DOE for power consumption by for HPC.

An Application and Hardware Driven Co-design for Current and Future Architectures Using Domain Specific Language

Mariam Umar+, Jeremy S. Meredith*, Jeffrey S. Vetter*, Kirk W. Cameron+
Department of Computer Science, Virginia Tech+, Oak Ridge National Laboratory, TN *
{mariam.umar,kirk.w.cameron} @vt.edu,{jsmeredith,vetter}@ornl.gov

Motivation

- Manual/Fixed hardware-software co-design is no more powerful because of:
  - Increase in complexity of architecture
  - Increase in scale of machines
- Previous solutions e.g., memory throttling and DVFS are less useful
- Problem becomes more challenging when:
  - Implementing and using application specific hardware, integrating hardware with software, changing application/hardware specifications at runtime

Our Approach:

- Automating hardware-software co-design, using Aspen, which generates:
  - Automated application model
  - Automated machine model
- Automation helps us:
  - Test application and micro benchmarks without requiring real hardware, using abstract model of machines
  - Applying solutions to current hardware as well as extrapolating them to future architectures
  - Saves programmers effort and time

Aspen DSL

- **Aspen DSL: (Abstract Scalable Performance Engineering Notation),** an environment for rapid exploration of new algorithm and architectures. It uses DSL to describe 2 main notations:
  - **Application Model**: used to measure flops, loads, stores etc.
  - **Machine Model**: finds theoretical peak performances etc.

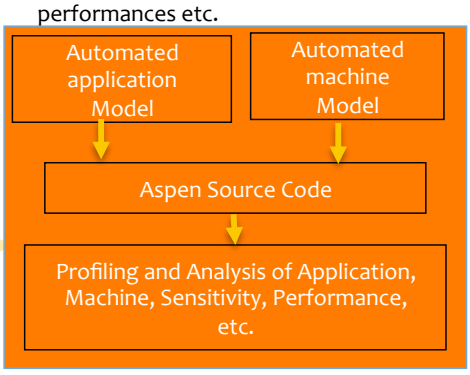


Fig1: Automated Application and Machine Model

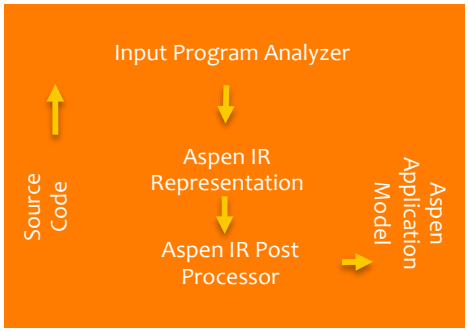


Fig2: Automated Application Model

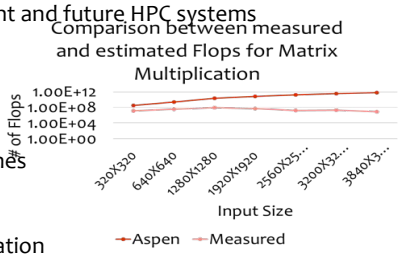
Results and Analysis

CPU	Core Clock	Memory	Memory BW	GPU	GPU Memory	GPU Clock
Intel I7-6660	2.4 GHz	32 GB	34.1 GB/sec	Tesla K20c	5GB	2.6 GHz

- Experiments:
  - Application analysis, Machine analysis, Sensitivity analysis of application with respect to hardware
  - Energy profiling of application on current and future HPC systems

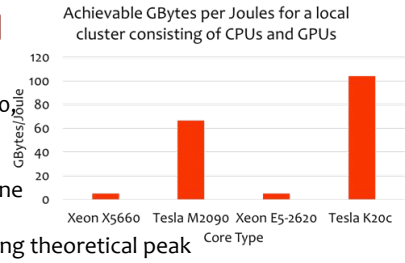
Application Analysis

- Application: Matrix Multiplication
- Analysis:
  - Estimated Flops by Aspen closely matches with measured value
- Uses:
  - Finding theoretical peak flops of application before running the experiment hence, avoiding wastage of resources.



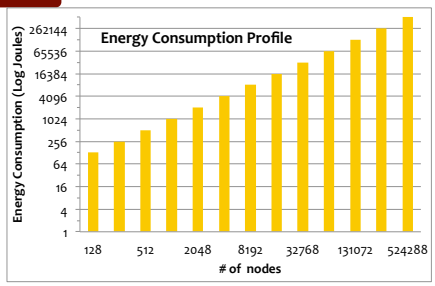
Machine Analysis

- Machine: 2 CPUs (Intel Xeon E5-2620, Intel Xeon X5660) and 2 GPUs (Tesla M2090 Tesla K20c)
- Our framework helps find out theoretical GBytes/Joule, using activity factor of machine
- Uses:
  - Selecting the best device at runtime using theoretical peak performance hence finishing the work in less time



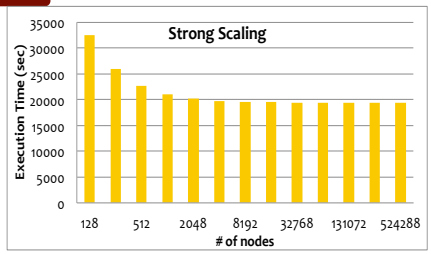
Energy Profiling on HPC Systems

- System: Titan
- Application: Lulesh
- Energy Profile:
  - Linearly increases with increasing number of nodes
  - Energy budget dictates the number of nodes to be used



Scalability Study on HPC Systems

- System: Titan
- Application: Lulesh
- Analysis:
  - Application is amenable to strong scaling
  - Execution time increases until 4096 nodes and then remains stable.



Acknowledgements

This material is based upon work supported in part by the NSF Grants No. 1422788, 0910784 and 0905187. The submitted manuscript has been authored by a contractor of the U.S. Government under Contract No. DE-AC05-00OR22725.

References

[1] S. Lee, J. S. Meredith, and J. S. Vetter, "COMPASS: A framework for automated performance modeling and prediction," ICS-2015. [2] Ang, J. A., et.al., Abstract Machine Models and Proxy Architectures for Exascale Computing. Co-HPC '14, New Orleans, Louisiana. [3] K. L. Spafford and J. S. Vetter, "Aspen: a domain specific language for performance modeling," SC12.

Bharti Wadhwa

An Object-based Data Storage Interface for Future HPC Storage Hierarchy

Exascale high performance computing (HPC) systems are expected to have deep memory and storage hierarchies and also require more efficient storage and I/O mechanisms than ever. Traditional disk block based POSIX storage systems may face severe challenges in satisfying these requirements due to their strong consistency requirement and the lack of hierarchy support and semantic interface. Various object-based storage systems have been evolved in past years to facilitate data storage and management but most of them are developed considering immutable data in Cloud computing environment and no object-based storage system is developed yet which can support high performance computing and upcoming deep memory storage hierarchy in exascale systems. To explore solutions to these challenges, we propose new object-based and semantic-rich data abstractions for scientific data management on exascale systems. The new data abstraction can also simplify users' involvement in data management. In this poster, we present initial design of such an object and its associated interface. We have also explored how this object based interface can facilitate next generation HPC systems by presenting the mapping of scientific data of one of our use-cases, i.e. I/O of a plasma physics simulation code, called VPIC, to objects.

Bharti is a 2nd year PhD student and member of Distributed Systems and Storage Laboratory in the Department of Computer Science at Virginia Tech. Her research interests include High Performance Computing, Cloud Computing, Large Scale Distributed Systems and Green computing.

She is currently working on a project in collaboration with Lawrence Berkeley National Lab (LBNL), Berkeley, USA, to develop object-based data abstractions which aim to facilitate storage and I/O in next generation HPC systems. She completed her summer internship at LBNL where she worked in the Scientific Data management group. She is also a Graduate Teaching Assistant for the course Computer Organization II in the Department of Computer Science at Virginia Tech.

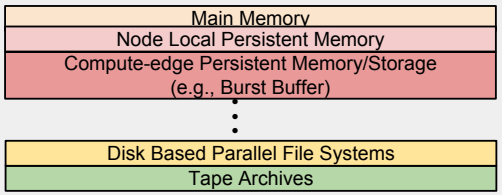
An Object-based Data Storage Interface for Future HPC Storage Hierarchy

Bharti Wadhwa Suren Byna Bin Dong Ali R. Butt

Goal: To represent our initial design of mapping scientific data to objects to facilitate storage and I/O in next generation HPC systems

Motivation

- Next generation HPC systems will face high complexity due to deep memory and storage hierarchy



Expected Exascale Systems Storage Stack

- Traditional file and block storage require strong consistency and lack:
  - Hierarchy storage support
  - Semantic Interface
- Object-based storage provide better performance and scalability over file and block storage

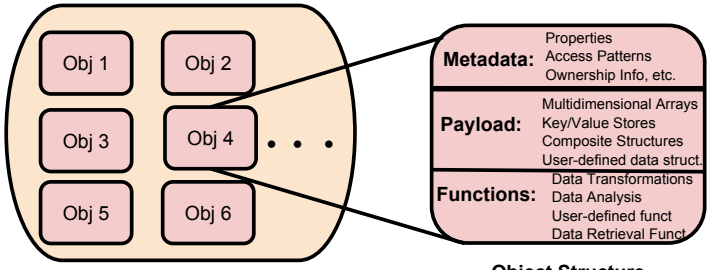
Object Storage Systems Comparison

	Ceph	Lustre	Swift	DAOS-M
Main Components	Metadata Cluster RADOS MDS OSDs CRUSH	MDSs MDTs OSSs OSTs Lustre N/w	Proxy server The Rings Object Server Container Account Server	DAOS Container Container Shards Versioned OSDs
Fundamental Primitive	key/value data	Objects stripes	key/value data	KV Stores
Open Source	Yes	Yes	Yes	Work in prog.
Fault Tolerance	Yes	detects during client ops	Yes	Yes
Written In	C++	C	Python	C
Support for Block and File Storage	Yes	Yes	No	Yes
Scalability	Linear(no. of OSDs)	depends on capacity of MDS	Highly Scalable	Highly Scalable

- Mostly developed for immutable data in Cloud Computing
- No real object storage system for HPC systems

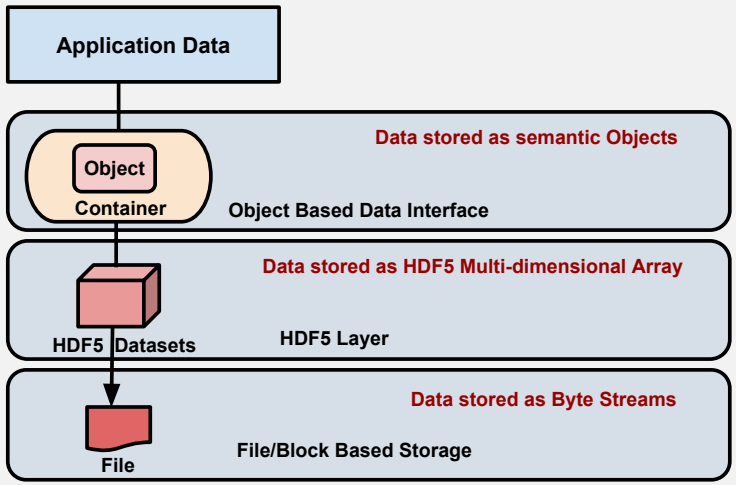
Design Overview

- Goal is to provide **object-based data storage interface** for scientific data management:
  - To facilitate storage and I/O in complex memory/storage hierarchy
  - Simplify users' involvement in data management



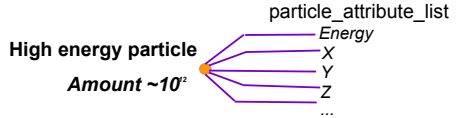
Object-based Data Container **Object Structure**

- Our proposed object-based data interface provides scientific applications with a much simpler and flexible gateway to handle storage and I/O
- This interface lies above HDF5 layer and maps the scientific data into objects
- It supports both single-node and distributed architecture



- Using HDF5 layer below our object interface is a temporary solution to test APIs
- We are working on to remove HDF5 dependency and to make more robust object interface

Data Storage for a Use Case: VPIC-IO



Without Object Interface

VPIC-IO Simulation

```
fid = H5FCreate();
for attr in
particle_attribute_list
{
 did =H5DCreate(fid, attr,
..);
 H5Sselect_hyperslab(did,
                local_size,
...);
 H5DWrite(did, buf, ..);
 H5DClose(did);
}
H5Fclose();
```

With Object Interface

VPIC-IO Simulation

```
cid = create_Container(attr, …,
);
plist = {"Persist" };
i = 0;
//Create attribute object
for attr in particle_attribute_list
{
   oid[i] = create_Object(cid,
attr,
            local_size,  plist, ...);
   write_Object(oid[], buf, ...);
i++;
}
finalize_Container(cid)
```

Conclusion and Future Work

- We have proposed a novel object-based data interface to facilitate storage and I/O of scientific data for future generation HPC systems
- Our object interface provides a flexible and simple gateway to manage application data
- Currently, we are working to provide this object-based interface to the top most layer of storage stack
- In future, we plan to extend this interface to all levels of exascale storage stack.
- Our goal is to eventually build a robust and autonomous runtime system, which will manage scientific data in object-oriented manner all over the exascale storage stack

Acknowledgement

We acknowledge the contribution of Quincey Koziol, John Readey, Houjun Tang and other members of 'Proactive Data Containers' project for their contribution

Zhengkai Wu

Predictive Ring Path Planning via 3D GPU Graphical Simulation in Subtractive 3D Printing

Graphical simulation provides a helpful visualization way to understand data and is used widely in software simulation and interface design. In this work, we first describe multi-axis graphical simulation based on CNC machine platform as subtractive 3D printing machining simulator. Then we introduce Predictive Ring Path Scalability for GPU volume pattern simulation in path planning. We describe the simulation process of GPU volume pattern as material removal process based on the target geometry. The process is equal to a blackbox optimization of ring parameter tuning with user interactive parameter selection. The ring simulation is based on spiral machining path and represents GPU volume ring scale for accessible orientation computation of the path generation. The accessibility map takes care of the 3D orientation calculation based on 2D feasible sequence of path slices in partition plane depending on tool movement position. Overall, we design the process of subtractive 3D printing from graphical modeling into path simulation layer by layer around the stock geometry as model based subtractive 3D path planning. For efficiency of path planning process, we optimize the output of accessibility map production of g-code as adaptive filtering so that the sequence of g-code is compressed and adaptively partitioned by storage limit. In the end, the roughing path planning based on general shape of the model produces rough shape of the fabrication model and finishing path provides refined surface fabrication product such as horse model of STL input. The entire model ring path generation can be predicted as iterative optimization process.

Zhengkai Wu is a PhD student at Georgia Tech working in high performance computing group under Dr Rich Vuduc and Dr Kurfess. Previously she got her MS degree from University of Central Florida in computer science. During her Ph.D years in Georgia Tech in Electrical and Computing Engineering department, she has worked on software optimization and green computing in data analysis and modeling application. She has published several papers in cloud computing and energy informatics. She is interested in 3D visualization and GPU based parallel computing application. She is now working on subtractive 3D printing and parallel computing algorithms in high performance computing for data analysis of GPU simulation and model based path planning.

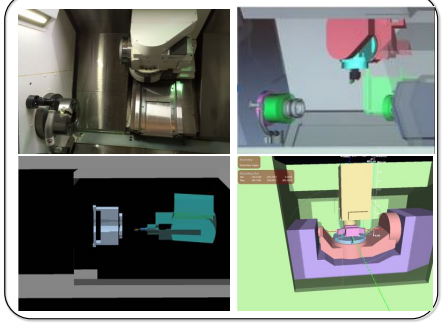
Predictive Ring Path Planning via 3D GPU Graphical Simulation in Subtractive 3D Printing

Zhengkai Wu, Rich Vuduc, Thomas Tucker, Thomas Kurfess

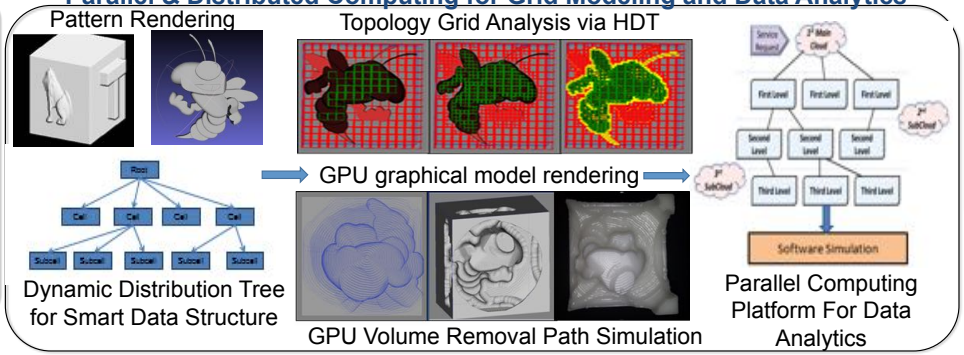
Supported by the National Science Foundation, CMMI – 1329742;Web Ref: http://cps-vo.org/node/16589

Paper Link: http://manufacturingscience.asmedigitalcollection.asme.org/article.aspx?articleid=2553186&resultClick=1

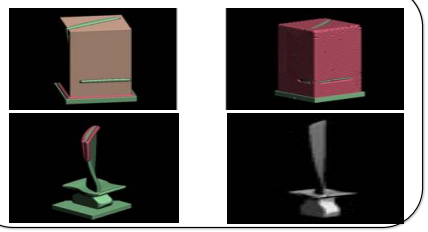
Subtractive 3D Printing Platform



Parallel & Distributed Computing for Grid Modeling and Data Analytics

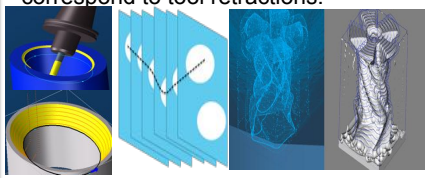


Layer based Path Planning



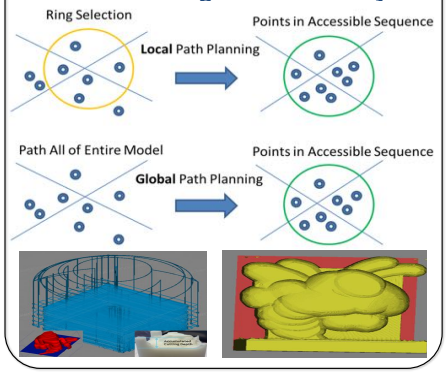
Ring Path Scalability

A tool position should follow an orientation that avoids collisions. "Accessibility map"[1] provides that allowable orientations for the tool. The jump lines in accessibility space correspond to tool retractions.



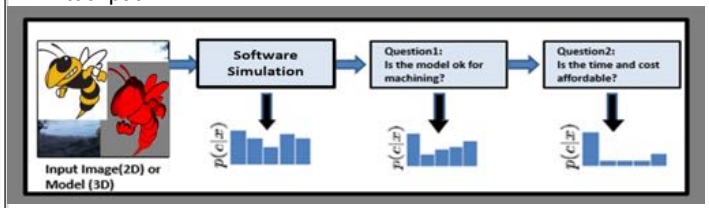
The 2D accessibility sequences form the 3D accessibility map space.

Predictive Ring Path Scalability



Model Ring Path 3D Simulation via GPU Rendering and Software Simulation

Data processing via advanced computing platform for interactive software path simulation. Successive offset volumes provide a sequence of XYZ points for a target tool path.



Black box Optimization

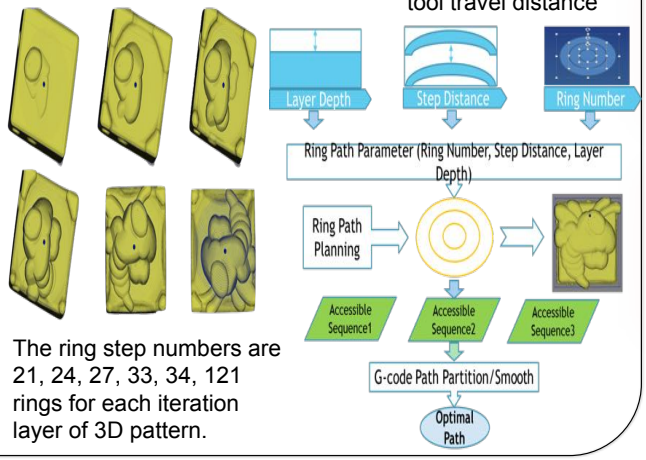
3-axis local region for linear maximal solution;
Reduce material removal cost, GPU rendering time and tool travel distance

Ring Predictive Algorithm

Initialize ring step numbers by rough estimate with fixed layer depth & step distance;

Increase ring circle addition as layer increases

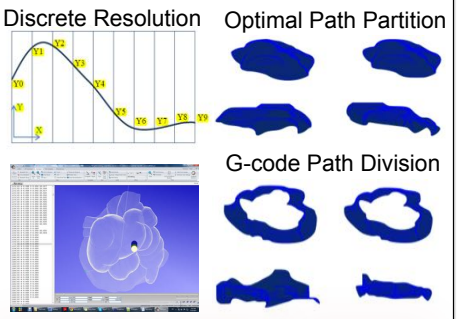
- If no path retraction & ring pattern satisfies→ pass
- If ring cover pattern not satisfies→ increase ring addition circles
- If path retraction → reduce ring addition circles to redo planning



The ring step numbers are 21, 24, 27, 33, 34, 121 rings for each iteration layer of 3D pattern.

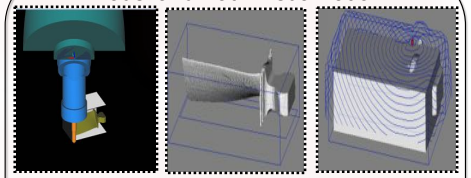
Path Planning Efficiency via G code

G-code simulation via distributed computing



Machining Visualization

Model and Path Visualization



Simulation, Roughing and Finishing

Reference: [1] D. Konobrytskyi, "Automatic CNC Toolpath Planning and Machining Simulation on Highly Parallel Computing Architectures", Ph.D. Thesis, Clemson University, 2013.

Hongjie Zheng

Large-scale Tsunami Run-up and Inundation Simulation Using an Explicit Moving Particle Simulation Solver Framework

There were many great earthquakes caused tsunami around the world in the past. Especially in the most recently Japan has experienced the Great East Japan Earthquake in 2011 and the Tohoku area was severely damaged. It is an urgent problem to predict the detail of tsunami and minimize the damage from the disaster. The large-scale numerical simulation can efficiently help us to achieve these purposes.

In this research, we have been developing the solver of LexADV_EMPS which is a large-scale parallel Explicit Moving Particle Simulation (E-MPS) solver framework in our ADVENTURE [1] system and LexADV [2] library. The LexADV_EMPS is especially useful for the 3D fluid for analysis involving free surfaces such as tsunami run-up simulations [3][4][5] on HPC. Our target problem in size is 10 million to 1 billion particles or more for practical problem. To achieve high parallel efficiency of the particle methods, we adopted three-level bucket structure as an efficient data management: the 1st level is used for domain decomposition, the 2nd level is used for halo exchange which communicates among the neighboring nodes, and the 3rd level is used for neighboring particle search.

As a practical problem, we simulated the tsunami run-up and inundation which occurred by the Great East Japan Earthquake in 2011. We used the Japan's petaflops supercomputer — K computer at RIKEN and Fujitsu Supercomputer PRIMEHPC FX100 installed in Nagoya University (Japan) as computational platform. The Number of particles is 1 billion for the K computer and 500 million for the FX100. We achieved about 90% of parallel efficiency both on the K computer and on the FX100.

1. ADVENTURE website: http://adventure.sys.t.u-tokyo.ac.jp/
2. LexADV website: http://adventure.sys.t.u-tokyo.ac.jp/lexadv/
3. Murotani, K., Koshizuka, S., Tamai, T., Shibata, K., Mitsume, N., Yoshimura, S., Tanaka, S., Hasegawa, K., Nagai E., and Fujisawa, T. 2014. Development of hierarchical domain decomposition explicit MPS method and application to large-scale tsunami analysis with floating objects. Journal of Advanced Simulation in Science and Engineering, 1, 1 (Oct. 2014), 16-35. DOI= http://doi.org/10.15748/jasse.1.16
4. Murotani, K., Koshizuka, S., Ogino, M., Shioya, R., Nakabayashi, Y., 2014. Development of distributed parallel explicit moving particle simulation (MPS) method and zoom up tsunami analysis on urban areas. SC14 Poster (Nov. 2014).
5. Murotani, K., Koshizuka, S., Ogino, M., and Shioya, R. 2015. Development of explicit moving particle simulation framework and zoom-up tsunami analysis system. SC15 Poster (Nov. 2015).

Hongjie Zheng is a postdoctoral researcher in the Center of Computational Mechanics Research (CCMR) of Toyo University. She received her Ph.D. and M.S from Kyushu University, where participated in the development of ADNVETURE project (an open-source for large-scale analysis and design) using Hierarchical Domain Decomposition Method with parallel data processing techniques. Her research interests include large-scale parallel computing and performance evaluation on the HPC (such as the K computer), Moving Particle Simulation, fluid-structure interaction simulation, and magnetic analysis.

Large-scale Tsunami Run-up and Inundation Simulation Using an Explicit Moving Particle Simulation Solver Framework

Hongjie Zheng[1] , Masao Ogino[2] , Kohei Murotani[3] , Seiichi Koshizuka[3], Ryuji Shioya[1]
[1]Toyo University, [2]Nagoya University, [3]Tokyo University

Background

There were many great earthquakes which caused tsunami around the world. Most recently Japan experienced the Great East Japan Earthquake in 2011 and the Tohoku area was severely damaged. It is an urgent problem to predict the detail of tsunami and minimize the damage from the disaster.

To achieve this purpose, we have been developing the solver of LexADV_EMPS which is a large-scale parallel Explicit Moving Particle Simulation (E-MPS) solver framework. This is especially useful for the 3D fluid analysis involving free surfaces such as tsunami run-up simulations on HPC.

Introduction of LexADV_EMPS

- A Parallel Explicit MPS (Moving Particle simulation) solver framework
- Especially useful for the 3-dimensional fluid for analysis involving free surfaces such as tsunami run-up simulations on HPC
- Target problem size
  10 million - 1 billion or more particles
- Functions for parallel computing
  Two-level domain decomposition
  Dynamic load balancing
  Halo exchange pattern of communication
- Hierarchical bucket structure of three levels
  Domain decomposition on 1st level bucket
  Halo exchange on 2nd level bucket
  Neighboring particle search on 3rd level bucket
- Download free from
  http://adventure.sys.t.u-tokyo.ac.jp/lexadv/
- Development and operating environment
  OS : UNIX, Linux
  Compiler : C
  Communication library : MPI

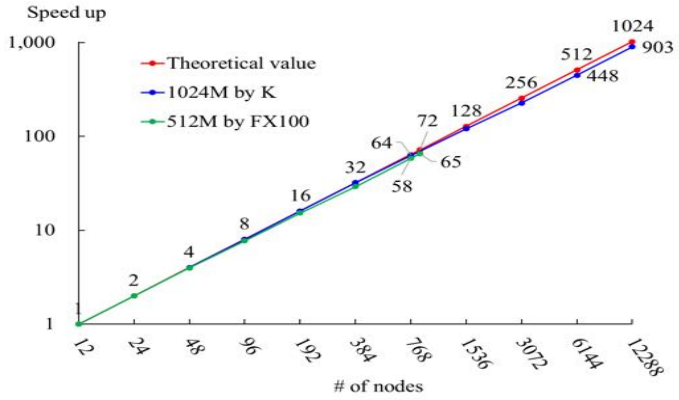


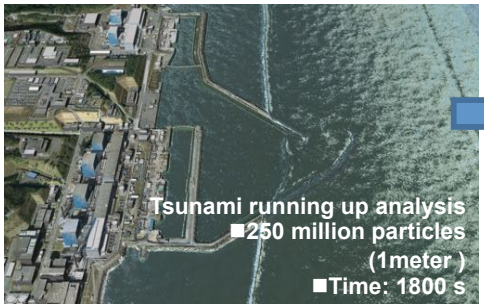
Figure 1: Strong scaling by K computer and FX100

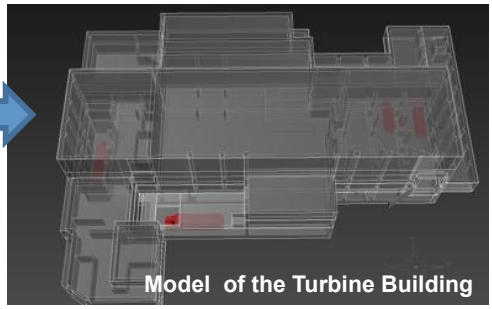
System	K computer	FX100
Site	RIKEN	Nagoya Univ.
Processor (peak FLOPS)	2.0 GHz SPARC64 VIIIfx, 8-Core (128 GFLOPS)	2.2 GHz Fujitsu SPARC64 XIfx, 32-Core (1126.4 GFLOPS)
Nodes	88,128	2,880
Number of Particles	1 billion	500 million

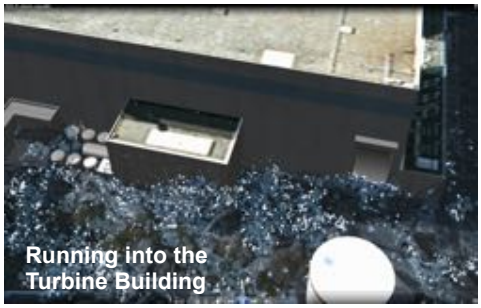
Table 1: Supercomputers used in this research

Figure 2: Tsunami Run-up and Inundation Simulation

To simulate tsunami effects on coastal areas, we analyzed the tsunami inundation area of Fukushima Daiichi Nuclear Power Station and the Turbine Building of Fukushima Daiichi NPS Unit 1.







ACKNOWLEDGMENTS This research was financially supported by JST CREST project "Development of a Numerical Library based on Hierarchical Domain Decomposition for Post Petascale Simulation". This research was partially supported by the HPCI Research Projects (Project ID hp150189) to make numerical experiments on K computer. This work was partially supported by "Nagoya University High Performance Computing Research Project for Joint Computational Science" in Japan to make numerical experiments on FX100. The authors also wish to thank Prometech Software, Inc., KOZO KEIKAKU ENGINEERING Inc., and all the members of the ADVENTURE project for their cooperation. Thanks to Prometech Software, Inc. providing photo-realistic visualization.

Working together to improve diversity in the HPC community

Top ten things you can do to support Women in HPC's mission

1. Get in touch: Let us know what you are doing to promote women in HPC. Do you have a local group, an event you would like to advertise, or an inspiring story to tell us? Share with us!
2. Measure: To understand our community and improve we first need to know our baseline. If we don't count we can't act! Please share with Women in HPC the demographics of your workplace, or any conferences, workshops or training events you run.
3. Understand: Find out how your team can improve diversity. Do you have an approximately equal gender split in your group? Do you have a "leaky pipeline"? Do women apply for your jobs? Are women applying but not appointed? Are women more likely to leave your workforce than men? Use this information to address the problem and develop your own best practice.
4. Share: disseminate the best practices you develop, what works and what doesn't. Discuss with colleagues the importance of diversity.
5. Participate: Attend our international workshops and discuss with like-minded people the methods to help recruit and retain women in the workforce.
6. Present: If you are an early career woman come along and present your HPC related work at a WHPC event, or encourage your early career colleagues to.
7. Contribute: Do you want to share the work you are doing with our readers, have an event, or an opinion about diversity? We have a blog and love contributions from our community. Email us with your ideas.
8. Join: Sign up and join as www.womeninhpc.org/membership
9. Build: Set up your own local Women in HPC group. Do you have a small team of women in your company, business, lab or university? Consider setting up your own local Women in HPC group, encourage like-minded women to meet up and network and encourage women to use HPC for their work!
10. Follow: Follow us on social media.

womeninhpc women_in_hpc groups/8105215

www.womeninhpc.org