REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
01-09-2013	Final Report	10-Jun-2009 - 9-Jun-2013

4. TITLE AND SUBTITLE
Computer-Aided Design of Drugs on Emerging Hybrid High Performance Computers

5a. CONTRACT NUMBER
W911NF-09-1-0311

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Michela Taufer

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
University of Delaware
210 Hullihen Hall

Newark, DE 19716 -0099

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
54723-MA.17

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
The project's overarching objective is to increase the time scale and length scale of scientific simulations relevant to the Army. To this end, the investigator and her group have been working at the University of Delaware (UD) on the algorithmic, implementation, and optimization aspects of large-scale molecular dynamics (MD) simulations on GPUs. During Years 2009-2012, the investigator designed the advance GPUs algorithms for MD simulations and integrated them into an open-source code called FE NZI; during the past year (Year 2012-2013) the investigator has

15. SUBJECT TERMS
GPU programming, Molecular Dynamics, Petascale and exascale simulations

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Michela Taufer
UU	UU	UU	UU		19b. TELEPHONE NUMBER: 302-831-0071

Report Title

Computer-Aided Design of Drugs on Emerging Hybrid High Performance Computers

ABSTRACT

The project's overarching objective is to increase the time scale and length scale of scientific simulations relevant to the Army. To this end, the investigator and her group have been working at the University of Delaware (UD) on the algorithmic, implementation, and optimization aspects of large-scale molecular dynamics (MD) simulations on GPUs. During Years 2009-2012, the investigator designed the advance GPUs algorithms for MD simulations and integrated them into an open-source code called FE NZI; during the past year (Year 2012-2013) the investigator has focused her effort on the performance optimization and characterization of the MD code across different GPU generations as well as on the integration of the code into a general framework for non-dedicated, high-end clusters that assures high resource utilization and enables coordinated progressions of MD trajectories.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received		Paper
08/15/2011	1.00	Narayan Ganesan, Brad A. Bauer, Timothy R. Lucas, Sandeep Patel, Michela Taufer. Structural, Dynamic, and Electrostatic Properties of FullyHydrated DMPC Bilayers From Molecular DynamicsSimulations Accelerated with Graphical Processing Units(GPUs), Journal of Computational Chemistry, (07 2011): 1. doi:
08/15/2011	8.00	Obaidur Rahaman, Trilce P. Estrada, Douglas J. Doren, Michela Taufer, Charles L. Brooks, Roger S. Armen. Evaluation of Several Two-Step Scoring Functions Based on Linear Interaction Energy, Effective Ligand Size, and Empirical Pair Potentials for Prediction of Protein–Ligand Binding Geometry and Free Energy, Journal of Chemical Information and Modeling, (06 2011): 1. doi: 10.1021/ci1003009
08/17/2012	10.00	Michela Taufer, Narayan Ganesan, Sandeep Patel. GPU enabled Macromolecular Simulation: Challenges and Opportunities, Computing in Science & Engineering, (01 2012): 0. doi: 10.1109/MCSE.2012.42
08/17/2012	11.00	Giorgos Arampatzis, Markos A. Katsoulakis, Petr Plechá?, Michela Taufer, Lifan Xu. Hierarchical fractional-step approximations and parallel kinetic Monte Carlo algorithms, Journal of Computational Physics, (07 2012): 0. doi: 10.1016/j.jcp.2012.07.017
08/17/2012	12.00	BRAD A. BAUER, , JOSEPH E. DAVIS, , MICHELA TAUFER, , SANDEEP PATEL. Molecular Dynamics Simulations of Aqueous Ions at the Liquid–Vapor Interface Accelerated Using Graphics Processors, J Comput Chem , (09 2010): 1. doi:
TOTAL:	**5**	

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper
08/07/2011	2.00 Brad A. Bauer, Joseph E. Davis, Michela Taufer, Sandeep Patel. Molecular dynamics simulations of aqueous ions at the liquid-vapor interface accelerated using graphics processors, Journal of Computational Chemistry, (02 2011): 0. doi: 10.1002/jcc.21578

TOTAL: **1**

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Invited talks and presentations (ARO is acknowledged):

March 2013: Transforming Computing Algorithms and Paradigms in HPC to Enable more Science out of our Day-to-day Simulations, Florida State University, Tallahassee, Florida. (Invited Talk)

March 2013: GPU-enabled Studies of Molecular Systems on Keeneland at ORNL - On pursuing high resource utilization and coordinated simulations' progression. Selected speaker at the NVIDIA GPU Technology Conference, San Jose, California. (Invited Talk with Sandeep Patel)

October 2012: Transforming Computing Algorithms and Paradigms in HPC to Enable more Science out of our Day-to-day Simulations, Oak Ridge national Laboratory, Oak Ridge, Tennessee. (Invited Talk)

October 2012: Transforming Computing Algorithms and Paradigms in HPC to Enable more Science out of our Day-to-day Simulations, Argonne National Laboratory, Chicago, Illinois. (Invited Talk)

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper	
08/07/2011	4.00	Narayan Ganesan , Roger D. Chamberlain, Jeremy Buhler and Michela Taufer. Rolling Partial Prefix-Sums to Speedup Uniform and Affine Recurrence Equations, Proceedings of SPIE, Defense, Security and Sensing Conference. 2011/04/26 00:00:00, . : ,
08/15/2011	5.00	Narayan Ganesan, Roger D. Chamberlain, Jeremy Buhler, Michela Taufer. Accelerating HMMER on GPUs by Implementing Hybrid Data and Task Parallelism. , Proceedings of the International Conference On Bioinformatics and Computational Biology (ACM-BCB), . 2010/08/02 00:00:00, . : ,
08/15/2011	3.00	Narayan Ganesan, Brad Bauer, Sandeep Patel, Michela Taufer. FENZI: GPU-enabled Molecular Dynamics Simulations of Large MembraneRegions based on the CHARMM force field and PME, Proceedings of the Tenth IEEE Workshop on Hi-Performance Computational Biology (HiCOMB). 2011/05/16 00:00:00, . : ,
08/15/2011	6.00	Lifan Xu, Stuart Collins, Dionisios G. Vlachos, and Michela Taufer. Parallelization of Tau-Leap Coarse-Grained Monte Carlo Simulations on GPUs, Proceedings of the IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS). , . : ,
08/15/2011	7.00	Michela Taufer, Omar Padron, Philip Saponaro, and Sandeep Patel. Improving Numerical Reproducibility and Stabilityin Large-Scale Numerical Simulations on GPUs, Proceedings of the IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS), . 2011/04/18 00:00:00, . : ,
08/15/2011	9.00	T. Estrada, R. Armen, M. Taufer. Automatic Selection of Near-Native Protein-LigandConformations using a Hierarchical Clustering andVolunteer Computing, ACM International Conference on Bioinformatics and Computational Biology. 2010/08/01 00:00:00, . : ,
08/20/2013	14.00	Samuel Schlachter, Stephen Herbein, Shuching Ou , Jeremy S. Logan , Sandeep Patel, Michela Taufer. . Efficient SDS Simulations on Multi-GPU Nodes of XSEDE High-end Clusters. , IEEE e-Science 2013. 2013/10/22 00:00:00, . : ,
08/20/2013	15.00	Matthew Wezowicz, Trilce Estrada, Sandeep Patel, Michela Taufer. Performance dissection ofa Molecular Dynamics codeacross CUDA and GPU generations, Proceedings of the 14th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC-13). 2013/05/20 00:00:00, . : ,
08/20/2013	16.00	Matthew Wezowicz, Michela Taufer. On the Cost of a General GPU Framework - The Strange Case of CUDA 4.0 vs. CUDA 5.0. (extended abstract), Proceedings of the ACM/IEEE International Conference for High Performance Computing and Communications conference (SC). 2012/11/11 00:00:00, . : ,
TOTAL:	**9**	

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received	Paper	
08/20/2013	13.00	Samule Schlachter, Sthephen Herbein, Shuching Ou , Jeremy .S. Logan, Sandeep Patel, Michela Taufer. Pursuing Resource Utilization and Coordinated Progression in GPU-enabled Molecular Simulations., (03 2013)

TOTAL: **1**

Number of Manuscripts:

Books

Received	Paper	

TOTAL:

Patents Submitted

Patents Awarded

Awards

Matthew Wezowicz, an UG student who was supported by this grant under Undergraduate Research Program (URP) in Summer 2012, was awarded with the Silver Medal at SC12 - ACM Student Poster competition with the poster "On the cost of a general GPU framework - The strange case of CUDA 4.0 vs. CUDA 5.0".

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Samuel Schlachter	0.10	
FTE Equivalent:	**0.10**	
Total Number:	**1**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
Trilce Estrada	0.60
FTE Equivalent:	**0.60**
Total Number:	**1**

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Michela Taufer	0.12	
FTE Equivalent:	**0.12**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED	Discipline
Mathew Wezowicz	0.10	
FTE Equivalent:	**0.10**	
Total Number:	**1**	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	1.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:	1.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:	1.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):	0.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	0.00

Names of Personnel receiving masters degrees

NAME	
Samuel Schlachter	
Total Number:	**1**

Names of personnel receiving PHDs

NAME
Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

Note: A MS Word version of this report has been attached to this report.

Approach

In Years 2009-2012, the investigator's approach to address the project objective is the implementation of an advanced GPU-based code called FEN ZI (yun dong de FEN ZI in Mandarin or moving MOLECULES in English) for molecular simulations. FEN ZI enables MD simulations at constant energy (NVE), constant temperature (NVT), and constant pressure and temperature (NPT) using a modified version of the CHARMM force field in terms of force field functional forms and measurement units. The entire MD simulation (i.e., intermolecular and long range potentials including PME) is performed on the GPU.
In Year 2012-2013, the investigator's approach has moved from the code implementation to its optimization and performance analysis on diverse generations of GPUs. Motivated by the fact that efficiently scheduled MD simulations on high-end GPU clusters still remains an open problem to be tackled, the investigator has also worked on the prototype of a framework that aims to complement, rather than rewrite, existing workflow and resource managers on such clusters. To this end, the framework relies on a companion module that complements the workflow manager and a several instances of a wrapper module that support the resource managers. While doing so, the modules support diverse programming languages and accelerators while assuring high resource utilization and coordinated progression of trajectories.

Scientific Barriers

Programming tools and technologies have been continuously and steadily evolving in the GPU community. The CUDA programming language and NVIDIA technology have been playing a leading role from the beginning. A first version of CUDA was released in 2007 and reached its maximum performance with CUDA 4.0. In 2013, NVIDIA has released a re-designed version of CUDA (CUDA 5.0) that integrates new programming features such as dynamic parallelism and has the potentials to support lower maintenance costs and higher cross-platform portability. At the same time, the hardware technology has also evolved, i.e., from the Tesla to the Kepler architectures, driven by the search for higher performance, lower power consumption, more efficient instruction pipelines, and more accurate results. When running codes that were initially developed and optimized for old CUDA versions and GPU generations on new GPU platforms with updated CUDA versions scientists may have to cope with an associated performance loss. The investigator has observed this behavior for MD simulations; during Year 2012-2013, the investigator has studied the problem and helped think of ways to reconcile performance with portability and maintainability.

When dealing with MD simulations that rely on an ensemble of independent trajectories, each of which executed on a single GPU on non-dedicated high-end clusters, scientists rely on runtime analysis and verification of properties to give important guidance on the simulation convergence of energies and simulation completion. The runtime analysis requires coordinated trajectories characterized by similar simulation stages at the time of analysis and makes the coordinated evolution of trajectories an important feature of our studies. Existing workflow and resource managers do not have adequate support for resource isolation for these GPUs. Consequently, when scientists submit MD jobs, they cannot guarantee isolated access to the resources they request. In the instance where nodes have multiple GPUs, the resource manager can assign multiple GPU jobs to a single node, but the scientist has to explicitly define in a submission script which job is allocated to which accelerator. Issues can arise when two scientists request the same GPU accelerator on the same node without knowing it. Sharing GPUs causes the substantial slow down of both jobs' execution times. In MD simulations, this can ultimately result in uncoordinated trajectories: while some trajectories use GPUs in isolation and reach orders of hundreds of nanoseconds, other trajectories share GPUs with other users' jobs and are several orders of magnitude behind.

Currently available solutions for efficient, coordinated simulations on the nodes of high-end, non-dedicated clusters may or may not rely on virtualization. When clusters do include virtualization, such as when using Shadowfax, scientists can schedule isolated CPU/GPU pairs and associate failures with GPUs. However, virtualization imposes significant overhead in terms of power, performance, and noise or jitter. Moreover, when available, solutions based on virtualization are GPU-language specific, e.g., for CUDA only, and require the cluster to have a hypervisor and Virtual Machines (VMs). Therefore, on most high-end clusters, virtualization is not available. Alternative solutions to virtualization include lightweight, user-level implementations on Linux operating systems, but these solutions are often dependent on a specific version of the GPU programming language or GPU generation. When virtualization or lightweight OS layers are not available, a workflow manager (e.g., Pegasus) has to couple with a cluster resource manager (e.g., Torque) to run the simulations. Jobs have to be packed into customized bundles (one bundle for each node including as many jobs as GPUs) before being submitted to the resource manager, resulting in tedious, error-prone manual work for the scientists. During Year 2012-2013, the investigator has studies solutions to make the simultaneous pursuit of efficient accelerator utilization and coordination of trajectories possible.
.
Significance

The investigator's work during Year 2012-2013 deals with MD simulations composed of multiple, independent GPU jobs that face the challenge of efficiently using multiple accelerators on single nodes while coordinating the trajectories' evolutions – i.e.,

avoiding major time gaps across trajectories for the sake of meaningful analysis. The performance of single jobs across different generations of GPU and CUDA versions were analysis. The efficiency of a framework prototype was quantified for two scenarios: (1) when the workflow manager on a non-dedicated, high-end cluster cannot handle GPU jobs with dynamically variable length in terms of performed number of steps per day, resulting in idle GPU times for jobs that are shorter than the allowed access time or unexpected terminations for jobs that are longer, and (2) when the resource manager cannot handle job failures, both hardware and application failures, resulting in idle GPUs.

The investigator modeled the maximum utilization with and without her framework for two ensembles of MD simulations on the Keeneland cluster at Oak Ridge National Laboratory and using the MD-code FEN ZI: SDS systems with dynamically variable job runtimes and carbon nanotube systems with computer system and application failures. The framework enables a higher utilization in both ensembles than traditional approaches (up to 10%) for simulations including a large number of trajectories such as the carbon nanotube systems. In general when dealing with hundreds of independent trajectories, runtime analysis becomes unfeasible, even when a small number fall behind. Intuitively scientists can expect that the increased utilization and the more dynamic assignment of tasks to GPUs that is supported by the investigator's framework implicitly assure a more coordinated progression of their long trajectories, allowing the scientists to perform analysis and verification of properties as the simulation evolves. The generality of the two modules building the framework potentially allows scientists to easily adapt the modules to wrap and handle other accelerators' codes as well as other types of simulations that require runtime analysis of properties across large ensembles of step-based jobs.

Accomplishments

In 2012-2013, Taufer and her group have been focusing on (1) performance dissection of Molecular Dynamics (MD) simulations across CUDA and GPU generations using the code FEN ZI; and (2) design of a framework for coordinated trajectory progression and efficient resource utilization of GPU-enabled molecular dynamics simulations on non-dedicated, high-end GPU clusters.
Performance dissection of Molecular Dynamics simulations across CUDA and GPU generations: This effort was motivated by the fact that programming tools and technologies have been continuously and steadily evolving in the GPU community since this project started. The CUDA programming language and NVIDIA technology have been playing a leading role from the beginning. A first version of CUDA was released in 2007 and reached its maximum performance with CUDA 4.0. In 2012, NVIDIA has started a re-design of the CUDA framework driven by software engineering perspectives characterized by the search for a general, multi-layer compilation infrastructure which compiler back-end is unified with OpenCL. This can ultimately have a significant impact on both maintenance costs and cross- platform portability. The software engineering community applauded this direction. At the same time, the hardware technology has also evolved, i.e., from the Tesla to the Kepler architectures, driven by the search for higher performance, lower power consumption, more efficient instruction pipelines, and more accurate results.

Work supported by this award indicates that the new direction of CUDA comes at some performance loss for large-scale simulations such as MD simulations. Code developers have in the past heavily optimized their codes for older generations of CUDA such as CUDA 4.0 and GPUs such as the C2050. When running these codes on platforms with updated CUDA versions and updated GPU architectures, scientists may have to cope with the associated performance loss. In this paper, rather than denying the problem and demonizing the emerging directions, we want to study the problem and help think of ways to reconcile performance with portability and maintainability. To this end, Taufer looked at the performance of our code FEN ZI from two different perspectives (i.e., the scientist and the computer scientist perspectives). Taufer dissected a diverse set of kernels from our code at three different levels for different code implementations, input data sizes, CUDA variants, and GPU architectures. First, for the different scenarios resulting for the possible combinations of codes, input data, CUDAs and GPUs, we looked at the amount of science each MD simulation can perform in terms of nanosecond per day (ns/day). Second, we zoomed into their executions, identify critical kernels, and present their performance from the point of view of their wall-clock times. Third, we dig into the hardware and look at the same critical kernels and any unusual behavior from their hardware resource point of view, e.g., registers, memory, I/O. The set of kernels in FEN ZI include diverse algorithmic components that can serve as basic building blocks in other real applications. Our analysis was performed on both kernels that expand and contract their number of threads in the thread pool to accommodate larger or smaller inputs (i.e., number of molecular atoms) and kernels that expand and contract the thread load on a fixed-size thread pool to accommodate the larger or smaller inputs. We identified performance sweet spots and trade-offs that reconcile antagonistic software generality and hardware improvements. Our main contributions to this effort are: (1) to capture driving factors at both software and hardware levels that impact performance and (2) to translate this new knowledge into important lessons for the community of GPU users and code developers.

More specifically, in Paper [7] and Poster [13] we documented the trade-offs between software generality and hardware improvements for a diverse set of kernels in an open-source molecular dynamics code. Our performance dissection allowed us to identify sweet spots where the loss in performance due to portability/maintenance is compensated by the hardware evolution. This is for large molecular systems on Kepler GPUs and CUDA 5.0. In this case, the faster hardware architecture is able to compensate the penalty associated to the software generality and ultimately can catch up in performance to become the fastest

simulation. We also documented how the trade-off between portability/maintenance and performance is a tough choice.

Coordinated trajectory progression and efficient resource utilization of GPU-enabled molecular dynamics simulations on non-dedicated, high-end GPU clusters: With the increase in non-dedicated high-end clusters including GPUs, a new challenge has emerged of pursuing coordinated trajectory progression and efficient resource utilization of GPU-enabled molecular dynamics (MD) simulations on these clusters. Because applying fully atomistically-resolved molecular models and force fields, MD simulations are more accurate than coarse-grained simulations and, therefore, are preferred for the study of thermodynamic properties in molecular systems. Accuracy in MD simulations comes at a high computing cost. Fortunately, the generation of an MD trajectory benefits from accelerators (i.e., GPUs, FPGAs, Xeon Phi) due to the parallelism embedded in the MD algorithm that is closely aligned with the accelerator's architecture. Thus, accelerators across the nodes of a high-end cluster are used to generate the trajectory ensemble at higher performance than traditional clusters. However, existing resource managers only support GPU accelerators but do not have adequate support for resource isolation for these GPUs. Consequently, when scientists submit MD jobs, they cannot guarantee isolated access to the resources they request. In the instance where nodes have multiple GPUs, the resource manager can assign multiple GPU jobs to a single node, but the scientist has to explicitly define in a submission script which job is allocated to which accelerator. Issues can arise when two scientists request the same GPU accelerator on the same node without knowing it. Sharing GPUs causes the substantial slow down of both jobs' execution times. In MD simulations, this can ultimately result in uncoordinated trajectories: while some trajectories use GPUs in isolation and reach orders of hundreds of nanoseconds, other trajectories share GPUs with other users' jobs and are several orders of magnitude behind.

A solution to this problem is important when often scientists rely on runtime analysis and verification of properties to give important guidance on the convergence of energies and simulation completion. The runtime analysis requires coordinated MD trajectories characterized by similar simulation stages at the time of analysis and cannot be guaranteed on current non-dedicated high-end clusters because an adequate resource isolation is not available in existing workflow and resource managers as we point out in Papers [1, 6]. Our work developed during Year 2012-2013 aimed to build a first prototype of a system that is able to supplement, rather than rewrite, existing workflow and resource managers. To this end, we proposed a companion module that complements workflow managers and a wrapper module that supports resource managers. We used our modules for scenarios in which (1) the workflow manager cannot handle GPU jobs with dynamically variable length in terms of performed number of steps per day, resulting in idle GPU times for jobs that are shorter than the allowed access time or unexpected terminations for jobs that are longer; and (2) the resource manager cannot handle job failures, both hardware and application failures, resulting in idle GPUs.

Computationally, we targeted the efficient study of the formation of sodium dodecyl sulfate (SDS) molecules in the presence of different types of salt concentrations and the energetics of carbon nanotubes in aqueous and electrolyte solutions when using molecular dynamics simulations on non-dedicated high-end GPU clusters. SDS molecules are relevant for the scientific community because studies indicate that SDS can play a key role in protein functions. Carbon nanotubes are relevant to understand cell penetrations. We modeled the maximum utilization of our approach in comparison to the traditional common approach for these two molecular simulations i.e., the SDS system with dynamically variable job runtimes and the carbon nanotube system with hardware and application failures. In light of our solution, we estimated increased utilization in both simulations. More specifically, with our framework prototype, we can expect that our utilization increases for both molecular systems and is much higher than traditional approaches (up to 10%) for simulations including a large number of trajectories such as the carbon nanotube systems. In general when dealing with hundreds of independent trajectories, runtime analysis becomes unfeasible, even when a small number fall behind.

Intuitively we can expect that the increased utilization and the more dynamic assignment of tasks to GPUs that is supported by our approach implicitly assure a more coordinated progression of the long trajectories, allowing scientists to perform analysis and verification of properties as the simulation evolves. As suggested by our work presented in Papers [1, 6], not only is our approach efficient for GPUs, but the generality of our two modules allows us to easily adapt them to wrap and handle other accelerators' codes as well as other types of simulations that require runtime analysis of properties across large ensembles of step-based jobs.

Peer-reviewed papers in journals (ARO is acknowledged):

[1] S. Schlachter, S. Herbein, S. Ou, J.S. Logan, S. Patel, and M. Taufer. Pursuing Resource Utilization and Coordinated Progression in GPU-enabled Molecular Simulations. IEEE Design&Test of Computers, 2013. (In press)

[2] M. Taufer, N. Ganesan, and S. Patel: GPU enabled Macromolecular Simulation: Challenges and Opportunities. IEEE Computing in Science and Engineering (CiSE), 15(1): 56 - 64, 2013.

[3] G. Arampatzis, M.A. Katsoulakis, P. Plechac, M. Taufer, and L. Xu: Hierarchical Fractional-step Approximations and Parallel Kinetic Monte Carlo Algorithms. J. Comput. Physics, 231(23): 7795-7814, 2012.

[4] N. Ganesan, B.A. Bauer, T. Lucas, S. Patel, and M. Taufer: Structural, Dynamic, and Electrostatic Properties of Fully Hydrated DMPC Bilayers from Molecular Dynamics Simulations Accelerated with Graphical Processing Units (GPUs). J. Comp. Chem., 32(14): 2958 – 2973, 2011.

[5] B.A. Bauer, J.E. Davis, M. Taufer, and S. Patel: Molecular Dynamics Simulations of Aqueous Ions at the Liquid-Vapor Interface Accelerated Using Graphics Processors. J. Comp. Chem., 32(3): 375–385, 2011.

Peer-reviewed papers in journals (ARO is acknowledged):

[6] S. Schlachter, S. Herbein, S. Ou, J.S. Logan, S. Patel, and M. Taufer. Efficient SDS Simulations on Multi-GPU Nodes of XSEDE High-end Clusters. eScience Conference 2013, October 2013, Beijing, China.

[7] M. Wezowicz, T. Estrada, S. Patel, and M. Taufer. Performance Dissection of a MD Code across CUDA and GPU Generations. In Proceedings of the 14th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC-13), April 2013, Boston, Massachusetts, USA. (Acceptance Rate: 16/42, 38%)

[8] N. Ganesan, B.A. Bauer, S. Patel, and M. Taufer: FEN ZI: GPU-enabled Molecular Dynamics Simulations of Large Membrane Regions based on the CHARMM force field and PME. In Proceedings of the Tenth IEEE International Workshop on High Performance Computational Biology, May 2011, Anchorage, Alaska, USA. (Acceptance Rate: 11/21, 52.3%)

[9] N. Ganesan, R. D. Chamberlain, J. Buhler, and M. Taufer: Rolling Partial Prefix-Sums To Speedup Evaluation of Uniform and Affine Recurrence Equations. In Proceedings of the DSS11 SPIE Defense, Security, and Sensing Symposium - Modeling and Simulation for Defense Systems and Applications VI, April 2011, Orlando, Florida, USA.

[10] N. Ganesan, R. D. Chamberlain, J. Buhler, and M. Taufer: Breaking the Sequential Dependency Bottleneck: Extracting Data Parallelism in the Presence of Serializing Data Dependencies. In Proceedings of the ACM International Conference on Bioinformatics and Computational Biology, August 2010, New York, USA. (Short Paper – Acceptance Rate: 30/99, 33%)

[11] M. Taufer, O. Padron, P. Saponaro, and S. Patel: Improving Numerical Reproducibility and Stability in Large-Scale Numerical Simulations on GPUs. In Proceedings of the IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS), April 2010, Atlanta, Georgia, USA. (Acceptance Rate: 127/527, 24%)

[12] L. Xu, M. Taufer, S. Collins, and D. G. Vlachos: Parallelization of Tau-Leap Coarse-Grained Monte Carlo Simulations on GPUs. In Proceedings of the IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS), April 2010, Atlanta, Georgia, USA. (Acceptance Rate: 127/528, 24%)
Peer-reviewed posters presented at conferences (ARO is acknowledged):

[13] M. Wezowicz and M. Taufer. On the Cost of a General GPU Framework - The Strange Case of CUDA 4.0 vs. CUDA 5.0. Poster in Proceedings of the ACM/IEEE International Conference for High Performance Computing and Communications conference (SC), November 2012, Salt Lake City, Utah, USA. (Acceptance Rate: 47%)

[14] N. Ganesan, S. Patel, and M. Taufer: Structural, Dynamics, and Electrostatic Properties of Fully Hydrated DMPC Bile's from Molecular Dynamics Simulations Accelerated with GPUs. Poster at the Research Symposium on Bioinformatics and Systems Biology, May 27, 2011. Delaware Bioinformatics Institute, University of Delaware, Newark, Delaware, USA.

[15] N. Ganesan, S. Patel, and M. Taufer: Simulations of Large Membrane Regions using GPU-enabled Computations - Preliminary Results. Poster in Proceedings of the 2010 Symposium on Application Accelerators in High Performance Computing (SAAHPC '10), July 13-15, 2010. University of Tennessee Conference Center, Knoxville, Tennessee, USA.

[16] L. Xu, S. Collin, M. Taufer, and D.G. Vlachos: Parallelization of Tau-Leaping Coarse-Grained Monte Carlo Method for Efficient and Accurate Simulations on GPUs. Poster in Proceedings of the ACM/IEEE International Conference for High Performance Computing and Communications conference (SC'09), November 2009, Portland, Washington, USA.

[17] M. Taufer, P. Saponaro, and O. Padron: Improving Reproducibility and Stability of Numerically Intensive Applications on Graphics Processing Units. Poster at the NVIDIA Research Summit, September 30- October 2, 2009, San Jose, CA, USA.

Invited talks and presentations (ARO is acknowledged):

[18] March 2013: Transforming Computing Algorithms and Paradigms in HPC to Enable more Science out of our Day-to-day Simulations, Florida State University, Tallahassee, Florida. (Invited Talk)

[19] March 2013: GPU-enabled Studies of Molecular Systems on Keeneland at ORNL - On pursuing high resource utilization and coordinated simulations' progression. Selected speaker at the NVIDIA GPU Technology Conference, San Jose, California. (Invited Talk with Sandeep Patel)

[20] October 2012: Transforming Computing Algorithms and Paradigms in HPC to Enable more Science out of our Day-to-day Simulations, Oak Ridge national Laboratory, Oak Ridge, Tennessee. (Invited Talk)

[21] October 2012: Transforming Computing Algorithms and Paradigms in HPC to Enable more Science out of our Day-to-day Simulations, Argonne National Laboratory, Chicago, Illinois. (Invited Talk)

[22] May 2012: GPU-enabled Macromolecular Simulation: Challenges and Opportunities, Selected speaker at the NVIDIA GPU Technology Conference, San Jose, California. (Invited Talk with Sandeep Patel)

[23] March 2012: GPU-enabled Macromolecular Simulation: Challenges and Opportunities, 2012 HPC Symposium at Lehigh University, Bethlehem, Pennsylvania. (Invited Talk)

[24] March 2012: Reengineering High-throughput Molecular Datasets for Scalable Clustering using MapReduce, Workshop on Trends in High-Performance Distributed Computing, Vrije Universiteit, Amsterdam, NL. (Invited Talk)

[25] February 2012: GPU-enabled Macromolecular Simulation: Challenges and Opportunities, NVIDIA Headquarter, San Jose, California. (Invited Talk)

[26] February 2012: GPU-enabled Macromolecular Simulation: Challenges and Opportunities. SIG-SYS Seminar, University of Delaware. (Invited Talk)

[27] December 2011: GPU-enabled Macromolecular Simulation: Challenges and Opportunities. NVIDIA webinar (Invited Talk)

[28] May 2011: FEN ZI: GPU-enabled Molecular Dynamics Simulations of Large Membrane Regions based on the CHARMM force field and PME. HiCOMB Workshop (joined with IPDPS), Anchorage, Alaska. (Conference Talk)

[29] March 2011: Enabling Faster Large-Scale Simulations with GPU Programming. Aberdeen Army Research Laboratory. (Invited Talk)

[30] October 2010: Enabling Faster Molecular Dynamics Simulations and Protein Motif-Finding with GPU Programming, Harvard Medical School. (Invited Talk)

[31] September 2010: MD simulations of large membrane. NVIDIA GPU Technology Conference, San Jose, California. (Invited Talk with Sandeep Patel and Narayan Ganesan)

[32] September 2010: Reformulating Algorithms for the GPU. NVIDIA GPU Technology Conference, San Jose, California. (Invited Talk with Narayan Ganesan)

[33] April 2010: Improving Numerical Reproducibility and Stability in Large-Scale Numerical Simulations on GPUs. IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS), Atlanta, Georgia. (Conference Talk)

Collaborations and Leveraged Funding

Taufer has been establishing collaborative research with faculty in chemical engineering and chemistry at the UD targeting large scale, multi-scale modeling simulations. She has also established new collaborative research with a local company called EM Photonics, working on hybrid computing and GPUs. These awards leverage this ARO project by providing Taufer with a richer set of computing infrastructures and applications that can be parallelized on and can benefit from hybrid resources.

In 2012, a HSAP/URAP mentorship proposal was awarded to support an undergraduate student. Research leveraged thie intial proposed research.

ARO - High School/Undergraduate Apprenticeship Program (HSAP/UGAP), $3,000, single PI
Title: Re-engineering and Optimizing the MD code FEN ZI for GPUs
Duration: Summer 2012 (8 weeks)
Description: Support one undergraduate student to learn how to use GPU programming optimizing techniques on applications relevant to the Army.

In 2011, two AFOSR projects were awarded to Taufer and collaborators at EM Photonics to leverage this project.

AFOSR STTR program – Highly Scalable Computational-Based Engineering Algorithms for Emerging Parallel Machine Architectures (Topic BT13), $99,999 ($29,997 at UD), Collaborating PI, with J. Humphrey (PI).
Title: Scalable Aero-Load and Aero-Elasticity Solvers for Massively Parallel Heterogeneous Computing Architectures
Duration: Spring 2012 – Spring 2013
Description: Support development of innovative algorithms for scientific computing, modeling and simulation on a multi-GPU environment. Emphasis is on parallelization of scientific applications across multiple GPUs.

AFOSR STTR program – Highly Scalable Computational-Based Engineering Algorithms for Emerging Parallel Machine Architectures (Topic BT13), $700,000 ($161,101 at UD), Taufer is PI of sub-contract at UD, with E. Kelmelis (PI, EM Photonics).
Title: Collaborative Research: Accelerated Linear Algebra Solvers for Multi-Core GPU-Based Computing Architecture (Phase II)
Duration: September 1, 2012 – August 31, 2014
Description: Support development of innovative algorithms for scientific computing, modeling and simulation on a multi-GPU environment. Emphasis is on algorithms related to sparse and dense linear algebra problems.

In 2010, one AFOSR project and one ARO HSAP project were awarded to Taufer and collaborators to leverage this project.

AFOSR STTR program – Highly Scalable Computational-Based Engineering Algorithms for Emerging Parallel Machine Architectures (Topic BT13), $99,000 ($34,125 at UD), Taufer is PI of sub-contract at UD, with E. Kelmelis (PI, EM Photonics)
Title: Collaborative Research: Accelerated Linear Algebra Solvers for Multi-Core GPU-Based Computing Architecture
Duration: June 8, 2010 – June 7, 2011
Description: Support development of innovative algorithms for scientific computing, modeling and simulation on a multi-GPU environment. Emphasis is on algorithms related to sparse and dense linear algebra problems.

ARO - High School Apprenticeship Program (HSAP), $3,000, single PI
Title: Exploring the Potentials of GPU Programming in Scientific Applications Relevant to the Army
Duration: Summer 2010 (8 weeks)
Description: Support one high-school student to learn and use GPU programming on applications relevant to the Army.

In 2009, two NSF projects were awarded to Taufer and collaborators to leverage this project.

NSF CDI #0941318, $463,657, Taufer is co-PI with Sandeep Patel (PI)
Title: CDI-Type I: Bridging the Gap Between Next-Generation Hybrid High Performance Computers and Physics Based Computational Models for Quantitative Description of Molecular Recognition
Duration: October 1, 2009 – September 30, 2012
Description: Design and implement advanced algorithms and middleware packages for polarizable force fields on multi-core and GPU systems, supported by the MapReduce paradigm.

NSF MRI #0922657, $451,051, Taufer is co-PI, with Douglas Doren (PI), Sandeep Patel, Dionisios Vlachos.
Title: Acquisition of a Facility for Computational Approaches to Molecular-Scale Problems
Duration: September 15, 2009 - September 14, 2012
Description: Support the acquisition of a hybrid-computing cluster, with GPU-accelerated computing nodes, for theoretical and experimental researchers at UD to study a number of problems in chemical sciences.

Conclusions

Over the past four years (2009-2013) this project has generated these main results:

(1) Algorithms for a realist and accurate representation of macro molecular systems and their dynamics on GPU-based high-end clusters, including an algorithm for Particle Mesh Ewald entirely performed on GPUs.

(2) An open-source GPU code called FEN ZI that can be downloaded from Google code and enables large-scale MD simulations on single GPUs. FEN ZI currently includes: NVT and NVE ensembles; the CHARMM force field; the Lennard-Jones interactions switching and shifting; long distance electrostatic interactions in terms of either reaction field or Ewald summation method including Particle Mesh Ewald (PME); explicit solvent with TIP3 or flexible SPC/Fw models.

(3) Fully atomistic molecular dynamics simulations of several molecular systems, such as the study of structural properties (i.e., atomic number density, electron density, and electrostatic potentials) of large DMPC lipid bilayers (on the order of a

quarter million atoms); the interaction of a WALP16 peptide with a model DMPC lipid bilayer; the formation of sodium dodecyl sulfate (SDS) molecules in the presence of different types of salt concentrations; and the energetics of carbon nanotubes in aqueous and electrolyte solutions using the FEN ZI code on GPU clusters.

(4) A first prototype of a framework that enables higher utilization of GPUs while pursing coordinated progression of MD trajectories for non-dedicated, high-end GPU clusters.

The research has resulted in six papers in peer-reviewed journals and six papers in peer-reviewed conference and workshop venues. These results were also presented in sixteen talks (invited and conference talks).

Two post-doctoral researchers have worked on this project. Both have received faculty positions in research universities at the end of their research experience in Taufer's group. One high school student, one undergraduate student, two master students, and one PhD candidate have been working partially or fully sported by this project. The high school student was accepted to the Undergraduate Computer Science Program at the Worcester Polytechnic Institute. The UG student is still involved in research in Taufer's group and is spending Summer 2013 as a summer intern at Oak ridge National Lab supported by the DoE Science Undergraduate Laboratory Internship (SULI) program. The two master students graduated; one accepted a position at Philips and the other works now as a software developer in Taufer's group. The graduate student left the group after a short research experience.

Technology Transfer

The investigator has interacted with these ARL members as possible users of the code: Dale Shires, Margaret Hurley, and Michael S. Lee at the U.S. Army Research Laboratory at the Aberdeen Proving Ground, Maryland. In the past two years the investigator has meet with Dale Shires and Margaret Hurley. Taufer is working with Margaret Hurley to pass the MD code FEN ZI to her group. FEN ZI as been released in Google code as open-source code. She is also committed to share the framework prototype of trajectory coordination with Dale Shires. A meeting was scheduled in May 2013 but had to be postponed to fall 2013 because of sudden commitments from both parties.

Future Plans

Future plans targeting technology transfer includes:
- Collaborate with Dale Share to tailor the framework for simulations and clusters at ARL in Aberdeen
- Collaborate with Margaret Hurley to complete the transfer and use of the FEN ZI code to her group

Taufer will explore the possibility to write a new project proposal with Share and Hurley targeting these two goals.

Technology Transfer

Computer-Aided Design of Drugs on Emerging Hybrid High Performance Computers
Proposal Number 54723-CS
Professor Michela Taufer, University of Delaware

Objectives

The project's overarching objective is to increase the time scale and length scale of scientific simulations relevant to the Army. To this end, the investigator and her group have been working at the University of Delaware (UD) on the algorithmic, implementation, and optimization aspects of large-scale molecular dynamics (MD) simulations on GPUs. During Years 2009-2012, the investigator designed the advance GPUs algorithms for MD simulations and integrated them into an open-source code called FE NZI; during the past year (Year 2012-2013) the investigator has focused her effort on the performance optimization and characterization of the MD code across different GPU generations as well as on the integration of the code into a general framework for non-dedicated, high-end clusters that assures high resource utilization and enables coordinated progressions of MD trajectories.

Approach

In Years 2009-2012, the investigator's approach to address the project objective is the implementation of an advanced GPU-based code called FEN ZI (*yun dong de FEN ZI* in Mandarin or *moving MOLECULES* in English) for molecular simulations. FEN ZI enables MD simulations at constant energy (NVE), constant temperature (NVT), and constant pressure and temperature (NPT) using a modified version of the CHARMM force field in terms of force field functional forms and measurement units. The entire MD simulation (i.e., intermolecular and long range potentials including PME) is performed on the GPU.

In Year 2012-2013, the investigator's approach has moved from the code implementation to its optimization and performance analysis on diverse generations of GPUs. Motivated by the fact that efficiently scheduled MD simulations on high-end GPU clusters still remains an open problem to be tackled, the investigator has also worked on the prototype of a framework that aims to complement, rather than rewrite, existing workflow and resource managers on such clusters. To this end, the framework relies on a companion module that complements the workflow manager and a several instances of a wrapper module that support the resource managers. While doing so, the modules support diverse programming languages and accelerators while assuring high resource utilization and coordinated progression of trajectories.

Scientific Barriers

Programming tools and technologies have been continuously and steadily evolving in the GPU community. The CUDA programming language and NVIDIA technology have been playing a leading role from the beginning. A first version of CUDA was released in 2007 and reached its maximum performance with CUDA 4.0. In 2013, NVIDIA has released a re-designed version of CUDA (CUDA 5.0) that integrates new programming features such as dynamic parallelism and has the potentials to support lower maintenance costs

and higher cross-platform portability. At the same time, the hardware technology has also evolved, i.e., from the Tesla to the Kepler architectures, driven by the search for higher performance, lower power consumption, more efficient instruction pipelines, and more accurate results. When running codes that were initially developed and optimized for old CUDA versions and GPU generations on new GPU platforms with updated CUDA versions scientists may have to cope with an associated performance loss. The investigator has observed this behavior for MD simulations; during Year 2012-2013, the investigator has studied the problem and helped think of ways to reconcile performance with portability and maintainability.

When dealing with MD simulations that rely on an ensemble of independent trajectories, each of which executed on a single GPU on non-dedicated high-end clusters, scientists rely on runtime analysis and verification of properties to give important guidance on the simulation convergence of energies and simulation completion. The runtime analysis requires coordinated trajectories characterized by similar simulation stages at the time of analysis and makes the coordinated evolution of trajectories an important feature of our studies. Existing workflow and resource managers do not have adequate support for resource isolation for these GPUs. Consequently, when scientists submit MD jobs, they cannot guarantee isolated access to the resources they request. In the instance where nodes have multiple GPUs, the resource manager can assign multiple GPU jobs to a single node, but the scientist has to explicitly define in a submission script which job is allocated to which accelerator. Issues can arise when two scientists request the same GPU accelerator on the same node without knowing it. Sharing GPUs causes the substantial slow down of both jobs' execution times. In MD simulations, this can ultimately result in uncoordinated trajectories: while some trajectories use GPUs in isolation and reach orders of hundreds of nanoseconds, other trajectories share GPUs with other users' jobs and are several orders of magnitude behind.

Currently available solutions for efficient, coordinated simulations on the nodes of high-end, non-dedicated clusters may or may not rely on virtualization. When clusters do include virtualization, such as when using Shadowfax, scientists can schedule isolated CPU/GPU pairs and associate failures with GPUs. However, virtualization imposes significant overhead in terms of power, performance, and noise or jitter. Moreover, when available, solutions based on virtualization are GPU-language specific, e.g., for CUDA only, and require the cluster to have a hypervisor and Virtual Machines (VMs). Therefore, on most high-end clusters, virtualization is not available. Alternative solutions to virtualization include lightweight, user-level implementations on Linux operating systems, but these solutions are often dependent on a specific version of the GPU programming language or GPU generation. When virtualization or lightweight OS layers are not available, a workflow manager (e.g., Pegasus) has to couple with a cluster resource manager (e.g., Torque) to run the simulations. Jobs have to be packed into customized bundles (one bundle for each node including as many jobs as GPUs) before being submitted to the resource manager, resulting in tedious, error-prone manual work for the scientists. During Year 2012-2013, the investigator has studies solutions to make the simultaneous pursuit of efficient accelerator utilization and coordination of trajectories possible.

.

Significance

The investigator's work during Year 2012-2013 deals with MD simulations composed of multiple, independent GPU jobs that face the challenge of efficiently using multiple accelerators on single nodes while coordinating the trajectories' evolutions – i.e., avoiding major time gaps across trajectories for the sake of meaningful analysis. The performance of single jobs across different generations of GPU and CUDA versions were analysis. The efficiency of a framework prototype was quantified for two scenarios: (1) when the workflow manager on a non-dedicated, high-end cluster cannot handle GPU jobs with dynamically variable length in terms of performed number of steps per day, resulting in idle GPU times for jobs that are shorter than the allowed access time or unexpected terminations for jobs that are longer, and (2) when the resource manager cannot handle job failures, both hardware and application failures, resulting in idle GPUs.

The investigator modeled the maximum utilization with and without her framework for two ensembles of MD simulations on the Keeneland cluster at Oak Ridge National Laboratory and using the MD-code FEN ZI: SDS systems with dynamically variable job runtimes and carbon nanotube systems with computer system and application failures. The framework enables a higher utilization in both ensembles than traditional approaches (up to 10%) for simulations including a large number of trajectories such as the carbon nanotube systems. In general when dealing with hundreds of independent trajectories, runtime analysis becomes unfeasible, even when a small number fall behind. Intuitively scientists can expect that the increased utilization and the more dynamic assignment of tasks to GPUs that is supported by the investigator's framework implicitly assure a more coordinated progression of their long trajectories, allowing the scientists to perform analysis and verification of properties as the simulation evolves. The generality of the two modules building the framework potentially allows scientists to easily adapt the modules to wrap and handle other accelerators' codes as well as other types of simulations that require runtime analysis of properties across large ensembles of step-based jobs.

Accomplishments

In 2012-2013, Taufer and her group have been focusing on (1) performance dissection of Molecular Dynamics (MD) simulations across CUDA and GPU generations using the code FEN ZI; and (2) design of a framework for coordinated trajectory progression and efficient resource utilization of GPU-enabled molecular dynamics simulations on non-dedicated, high-end GPU clusters.

Performance dissection of Molecular Dynamics simulations across CUDA and GPU generations: This effort was motivated by the fact that programming tools and technologies have been continuously and steadily evolving in the GPU community since this project started. The CUDA programming language and NVIDIA technology have been playing a leading role from the beginning. A first version of CUDA was released in 2007 and reached its maximum performance with CUDA 4.0. In 2012, NVIDIA has started a re-design of the CUDA framework driven by software engineering perspectives characterized by the search for a general, multi-layer

compilation infrastructure which compiler back-end is unified with OpenCL. This can ultimately have a significant impact on both maintenance costs and cross- platform portability. The software engineering community applauded this direction. At the same time, the hardware technology has also evolved, i.e., from the Tesla to the Kepler architectures, driven by the search for higher performance, lower power consumption, more efficient instruction pipelines, and more accurate results.

Work supported by this award indicates that the new direction of CUDA comes at some performance loss for large-scale simulations such as MD simulations. Code developers have in the past heavily optimized their codes for older generations of CUDA such as CUDA 4.0 and GPUs such as the C2050. When running these codes on platforms with updated CUDA versions and updated GPU architectures, scientists may have to cope with the associated performance loss. In this paper, rather than denying the problem and demonizing the emerging directions, we want to study the problem and help think of ways to reconcile performance with portability and maintainability. To this end, Taufer looked at the performance of our code FEN ZI from two different perspectives (i.e., the scientist and the computer scientist perspectives). Taufer dissected a diverse set of kernels from our code at three different levels for different code implementations, input data sizes, CUDA variants, and GPU architectures. First, for the different scenarios resulting for the possible combinations of codes, input data, CUDAs and GPUs, we looked at the amount of science each MD simulation can perform in terms of nanosecond per day (ns/day). Second, we zoomed into their executions, identify critical kernels, and present their performance from the point of view of their wall-clock times. Third, we dig into the hardware and look at the same critical kernels and any unusual behavior from their hardware resource point of view, e.g., registers, memory, I/O. The set of kernels in FEN ZI include diverse algorithmic components that can serve as basic building blocks in other real applications. Our analysis was performed on both kernels that expand and contract their number of threads in the thread pool to accommodate larger or smaller inputs (i.e., number of molecular atoms) and kernels that expand and contract the thread load on a fixed-size thread pool to accommodate the larger or smaller inputs. We identified performance sweet spots and trade-offs that reconcile antagonistic software generality and hardware improvements. Our main contributions to this effort are: (1) to capture driving factors at both software and hardware levels that impact performance and (2) to translate this new knowledge into important lessons for the community of GPU users and code developers.

More specifically, in Paper [7] and Poster [13] we documented the trade-offs between software generality and hardware improvements for a diverse set of kernels in an open-source molecular dynamics code. Our performance dissection allowed us to identify sweet spots where the loss in performance due to portability/maintenance is compensated by the hardware evolution. This is for large molecular systems on Kepler GPUs and CUDA 5.0. In this case, the faster hardware architecture is able to compensate the penalty associated to the software generality and ultimately can catch up in performance to become the fastest simulation. We also documented how the trade-off between portability/maintenance and performance is a tough choice.

Coordinated trajectory progression and efficient resource utilization of GPU-enabled molecular dynamics simulations on non-

dedicated, high-end GPU clusters: With the increase in non-dedicated high-end clusters including GPUs, a new challenge has emerged of pursuing coordinated trajectory progression and efficient resource utilization of GPU-enabled molecular dynamics (MD) simulations on these clusters. Because applying fully atomistically-resolved molecular models and force fields, MD simulations are more accurate than coarse-grained simulations and, therefore, are preferred for the study of thermodynamic properties in molecular systems. Accuracy in MD simulations comes at a high computing cost. Fortunately, the generation of an MD trajectory benefits from accelerators (i.e., GPUs, FPGAs, Xeon Phi) due to the parallelism embedded in the MD algorithm that is closely aligned with the accelerator's architecture. Thus, accelerators across the nodes of a high-end cluster are used to generate the trajectory ensemble at higher performance than traditional clusters. However, existing resource managers only support GPU accelerators but do not have adequate support for resource isolation for these GPUs. Consequently, when scientists submit MD jobs, they cannot guarantee isolated access to the resources they request. In the instance where nodes have multiple GPUs, the resource manager can assign multiple GPU jobs to a single node, but the scientist has to explicitly define in a submission script which job is allocated to which accelerator. Issues can arise when two scientists request the same GPU accelerator on the same node without knowing it. Sharing GPUs causes the substantial slow down of both jobs' execution times. In MD simulations, this can ultimately result in uncoordinated trajectories: while some trajectories use GPUs in isolation and reach orders of hundreds of nanoseconds, other trajectories share GPUs with other users' jobs and are several orders of magnitude behind.

A solution to this problem is important when often scientists rely on runtime analysis and verification of properties to give important guidance on the convergence of energies and simulation completion. The runtime analysis requires coordinated MD trajectories characterized by similar simulation stages at the time of analysis and cannot be guaranteed on current non-dedicated high-end clusters because an adequate resource isolation is not available in existing workflow and resource managers as we point out in Papers [1, 6]. Our work developed during Year 2012-2013 aimed to build a first prototype of a system that is able to supplement, rather than rewrite, existing workflow and resource managers. To this end, we proposed a companion module that complements workflow managers and a wrapper module that supports resource managers. We used our modules for scenarios in which (1) the workflow manager cannot handle GPU jobs with dynamically variable length in terms of performed number of steps per day, resulting in idle GPU times for jobs that are shorter than the allowed access time or unexpected terminations for jobs that are longer; and (2) the resource manager cannot handle job failures, both hardware and application failures, resulting in idle GPUs.

Computationally, we targeted the efficient study of the formation of sodium dodecyl sulfate (SDS) molecules in the presence of different types of salt concentrations and the energetics of carbon nanotubes in aqueous and electrolyte solutions when using molecular dynamics simulations on non-dedicated high-end GPU clusters. SDS molecules are relevant for the scientific community because studies indicate that SDS can play a key role in protein functions. Carbon nanotubes are relevant to understand cell penetrations. We modeled the maximum utilization of our approach in comparison to the traditional common approach for these two

molecular simulations i.e., the SDS system with dynamically variable job runtimes and the carbon nanotube system with hardware and application failures. In light of our solution, we estimated increased utilization in both simulations. More specifically, with our framework prototype, we can expect that our utilization increases for both molecular systems and is much higher than traditional approaches (up to 10%) for simulations including a large number of trajectories such as the carbon nanotube systems. In general when dealing with hundreds of independent trajectories, runtime analysis becomes unfeasible, even when a small number fall behind.

Intuitively we can expect that the increased utilization and the more dynamic assignment of tasks to GPUs that is supported by our approach implicitly assure a more coordinated progression of the long trajectories, allowing scientists to perform analysis and verification of properties as the simulation evolves. As suggested by our work presented in Papers [1, 6], not only is our approach efficient for GPUs, but the generality of our two modules allows us to easily adapt them to wrap and handle other accelerators' codes as well as other types of simulations that require runtime analysis of properties across large ensembles of step-based jobs.

Peer-reviewed papers in journals (ARO is acknowledged):

[1] S. Schlachter, S. Herbein, S. Ou, J.S. Logan, S. Patel, and M. Taufer. Pursuing Resource Utilization and Coordinated Progression in GPU-enabled Molecular Simulations. *IEEE Design&Test of Computers*, 2013. (In press)

[2] M. Taufer, N. Ganesan, and S. Patel: GPU enabled Macromolecular Simulation: Challenges and Opportunities. *IEEE Computing in Science and Engineering (CiSE),* 15(1): 56 - 64, 2013.

[3] G. Arampatzis, M.A. Katsoulakis, P. Plechac, M. Taufer, and L. Xu: Hierarchical Fractional-step Approximations and Parallel Kinetic Monte Carlo Algorithms. *J. Comput. Physics,* 231(23): 7795-7814, 2012.

[4] N. Ganesan, B.A. Bauer, T. Lucas, S. Patel, and M. Taufer: Structural, Dynamic, and Electrostatic Properties of Fully Hydrated DMPC Bilayers from Molecular Dynamics Simulations Accelerated with Graphical Processing Units (GPUs). *J. Comp. Chem.,* 32(14): 2958 – 2973, 2011.

[5] B.A. Bauer, J.E. Davis, M. Taufer, and S. Patel: Molecular Dynamics Simulations of Aqueous Ions at the Liquid-Vapor Interface Accelerated Using Graphics Processors. *J. Comp. Chem.,* 32(3): 375–385, 2011.

Peer-reviewed papers in journals (ARO is acknowledged):

[6] S. Schlachter, S. Herbein, S. Ou, J.S. Logan, S. Patel, and M. Taufer. Efficient SDS Simulations on Multi-GPU Nodes of XSEDE High-end Clusters. *eScience Conference 2013*, October 2013, Beijing, China.

[7] M. Wezowicz, T. Estrada, S. Patel, and M. Taufer. Performance Dissection of a MD Code across CUDA and GPU Generations. In *Proceedings of the 14th IEEE International Workshop on Parallel and Distributed Scientific and Engineering*

Computing (PDSEC-13), April 2013, Boston, Massachusetts, USA. (*Acceptance Rate:* 16/42, 38%)

[8] N. Ganesan, B.A. Bauer, S. Patel, and M. Taufer: FEN ZI: GPU-enabled Molecular Dynamics Simulations of Large Membrane Regions based on the CHARMM force field and PME. In *Proceedings of the Tenth IEEE International Workshop on High Performance Computational Biology,* May 2011, Anchorage, Alaska, USA. (*Acceptance Rate:* 11/21, 52.3%)

[9] N. Ganesan, R. D. Chamberlain, J. Buhler, and M. Taufer: Rolling Partial Prefix-Sums To Speedup Evaluation of Uniform and Affine Recurrence Equations. In *Proceedings of the DSS11 SPIE Defense, Security, and Sensing Symposium - Modeling and Simulation for Defense Systems and Applications VI,* April 2011, Orlando, Florida, USA.

[10] N. Ganesan, R. D. Chamberlain, J. Buhler, and M. Taufer: Breaking the Sequential Dependency Bottleneck: Extracting Data Parallelism in the Presence of Serializing Data Dependencies. In *Proceedings of the ACM International Conference on Bioinformatics and Computational Biology*, August 2010, New York, USA. (Short Paper – *Acceptance Rate:* 30/99, 33%)

[11] M. Taufer, O. Padron, P. Saponaro, and S. Patel: Improving Numerical Reproducibility and Stability in Large-Scale Numerical Simulations on GPUs. In *Proceedings of the IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS),* April 2010, Atlanta, Georgia, USA. (*Acceptance Rate:* 127/527, 24%)

[12] L. Xu, M. Taufer, S. Collins, and D. G. Vlachos: Parallelization of Tau-Leap Coarse-Grained Monte Carlo Simulations on GPUs. In *Proceedings of the IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS),* April 2010, Atlanta, Georgia, USA. (*Acceptance Rate:* 127/528, 24%)

Peer-reviewed posters presented at conferences (ARO is acknowledged):

[13] M. Wezowicz and M. Taufer. On the Cost of a General GPU Framework - The Strange Case of CUDA 4.0 vs. CUDA 5.0. Poster in *Proceedings of the ACM/IEEE International Conference for High Performance Computing and Communications conference (SC),* November 2012, Salt Lake City, Utah, USA. (*Acceptance Rate:* 47%)

[14] N. Ganesan, S. Patel, and M. Taufer: Structural, Dynamics, and Electrostatic Properties of Fully Hydrated DMPC Bile's from Molecular Dynamics Simulations Accelerated with GPUs. Poster at the *Research Symposium on Bioinformatics and Systems Biology,* May 27, 2011. Delaware Bioinformatics Institute, University of Delaware, Newark, Delaware, USA.

[15] N. Ganesan, S. Patel, and M. Taufer: Simulations of Large Membrane Regions using GPU-enabled Computations - Preliminary Results. Poster in *Proceedings of the 2010 Symposium on Application Accelerators in High Performance Computing (SAAHPC '10),* July 13-15, 2010. University of Tennessee Conference Center, Knoxville, Tennessee, USA.

[16] L. Xu, S. Collin, M. Taufer, and D.G. Vlachos: Parallelization of Tau-Leaping Coarse-Grained Monte Carlo Method for Efficient and Accurate Simulations on GPUs. Poster in *Proceedings of the ACM/IEEE International Conference for High Performance Computing and Communications conference (SC'09),* November 2009, Portland, Washington, USA.

[17] M. Taufer, P. Saponaro, and O. Padron: Improving Reproducibility and Stability of Numerically Intensive Applications on Graphics Processing Units. Poster at the *NVIDIA Research Summit*, September 30- October 2, 2009, San Jose, CA, USA.

Invited talks and presentations (ARO is acknowledged):

[18] March 2013: Transforming Computing Algorithms and Paradigms in HPC to Enable more Science out of our Day-to-day Simulations, Florida State University, Tallahassee, Florida. (Invited Talk)

[19] March 2013: GPU-enabled Studies of Molecular Systems on Keeneland at ORNL - On pursuing high resource utilization and coordinated simulations' progression. Selected speaker at the NVIDIA GPU Technology Conference, San Jose, California. (Invited Talk with Sandeep Patel)

[20] October 2012: Transforming Computing Algorithms and Paradigms in HPC to Enable more Science out of our Day-to-day Simulations, Oak Ridge national Laboratory, Oak Ridge, Tennessee. (Invited Talk)

[21] October 2012: Transforming Computing Algorithms and Paradigms in HPC to Enable more Science out of our Day-to-day Simulations, Argonne National Laboratory, Chicago, Illinois. (Invited Talk)

[22] May 2012: GPU-enabled Macromolecular Simulation: Challenges and Opportunities, Selected speaker at the NVIDIA GPU Technology Conference, San Jose, California. (Invited Talk with Sandeep Patel)

[23] March 2012: GPU-enabled Macromolecular Simulation: Challenges and Opportunities, 2012 HPC Symposium at Lehigh University, Bethlehem, Pennsylvania. (Invited Talk)

[24] March 2012: Reengineering High-throughput Molecular Datasets for Scalable Clustering using MapReduce, Workshop on Trends in High-Performance Distributed Computing, Vrije Universiteit, Amsterdam, NL. (Invited Talk)

[25] February 2012: GPU-enabled Macromolecular Simulation: Challenges and Opportunities, NVIDIA Headquarter, San Jose, California. (Invited Talk)

[26] February 2012: GPU-enabled Macromolecular Simulation: Challenges and Opportunities. *SIG-SYS Seminar*, University of Delaware. (Invited Talk)

[27] December 2011: GPU-enabled Macromolecular Simulation: Challenges and Opportunities. *NVIDIA webinar* (Invited Talk)

[28] May 2011: FEN ZI: GPU-enabled Molecular Dynamics Simulations of Large Membrane Regions based on the CHARMM force field and PME. *HiCOMB Workshop (joined with IPDPS),* Anchorage, Alaska. (Conference Talk)

[29] March 2011: Enabling Faster Large-Scale Simulations with GPU Programming. *Aberdeen Army Research Laboratory.* (Invited Talk)

[30] October 2010: Enabling Faster Molecular Dynamics Simulations and Protein Motif-Finding with GPU Programming, *Harvard Medical School.* (Invited Talk)

[31] September 2010: MD simulations of large membrane. *NVIDIA GPU Technology Conference,* San Jose, California. (Invited Talk with Sandeep Patel and Narayan Ganesan)

[32] September 2010: Reformulating Algorithms for the GPU. *NVIDIA GPU Technology Conference,* San Jose, California. (Invited Talk with Narayan Ganesan)

[33] April 2010: Improving Numerical Reproducibility and Stability in Large-Scale Numerical Simulations on GPUs. *IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS),* Atlanta, Georgia. (Conference Talk)

Collaborations and Leveraged Funding

Taufer has been establishing collaborative research with faculty in chemical engineering and chemistry at the UD targeting large scale, multi-scale modeling simulations. She has also established new collaborative research with a local company called EM Photonics, working on hybrid computing and GPUs. These awards leverage this ARO project by providing Taufer with a richer set of computing infrastructures and applications that can be parallelized on and can benefit from hybrid resources.

In 2012, a HSAP/URAP mentorship proposal was awarded to support an undergraduate student. Research leveraged thie intial proposed research.

ARO - High School/Undergraduate Apprenticeship Program (HSAP/UGAP), $3,000, single PI
Title: ***Re-engineering and Optimizing the MD code FEN ZI for GPUs***
Duration: Summer 2012 (8 weeks)
Description: Support one undergraduate student to learn how to use GPU programming optimizing techniques on applications relevant to the Army.

In 2011, two AFOSR projects were awarded to Taufer and collaborators at EM Photonics to leverage this project.

AFOSR STTR program – Highly Scalable Computational-Based Engineering Algorithms for Emerging Parallel Machine Architectures (Topic BT13), $99,999 ($29,997 at UD), Collaborating PI, with J. Humphrey (PI).
Title: ***Scalable Aero-Load and Aero-Elasticity Solvers for Massively Parallel Heterogeneous Computing Architectures***
Duration: Spring 2012 – Spring 2013
Description: Support development of innovative algorithms for scientific computing, modeling and simulation on a multi-GPU environment. Emphasis is on parallelization of scientific applications across multiple GPUs.

AFOSR STTR program – Highly Scalable Computational-Based Engineering Algorithms for Emerging Parallel Machine Architectures (Topic BT13), $700,000 ($161,101 at UD), Taufer is PI of sub-contract at UD, with E. Kelmelis (PI, EM Photonics).
Title: ***Collaborative Research: Accelerated Linear Algebra Solvers for Multi-Core GPU-Based Computing Architecture (Phase II)***
Duration: September 1, 2012 – August 31, 2014
Description: Support development of innovative algorithms for scientific computing, modeling and simulation on a multi-GPU environment. Emphasis is on algorithms related to sparse and dense linear algebra problems.

In 2010, one AFOSR project and one ARO HSAP project were awarded to Taufer and collaborators to leverage this project.

AFOSR STTR program – Highly Scalable Computational-Based Engineering Algorithms for Emerging Parallel Machine Architectures (Topic BT13), $99,000 ($34,125 at UD), Taufer is PI of sub-contract at UD, with E. Kelmelis (PI, EM Photonics)
Title: ***Collaborative Research: Accelerated Linear Algebra Solvers for Multi-Core GPU-Based Computing Architecture***
Duration: June 8, 2010 – June 7, 2011
Description: Support development of innovative algorithms for scientific computing, modeling and simulation on a multi-GPU environment. Emphasis is on algorithms related to sparse and dense linear algebra problems.

ARO - High School Apprenticeship Program (HSAP), $3,000, single PI
Title: ***Exploring the Potentials of GPU Programming in Scientific Applications Relevant to the Army***
Duration: Summer 2010 (8 weeks)
Description: Support one high-school student to learn and use GPU programming on applications relevant to the Army.

In 2009, two NSF projects were awarded to Taufer and collaborators to leverage this project.

NSF CDI #0941318, $463,657, Taufer is co-PI with Sandeep Patel (PI)
Title: ***CDI-Type I: Bridging the Gap Between Next-Generation Hybrid High Performance Computers and Physics Based Computational Models for Quantitative Description of Molecular Recognition***
Duration: October 1, 2009 – September 30, 2012
Description: Design and implement advanced algorithms and middleware packages for polarizable force fields on multi-core and GPU systems, supported by the MapReduce paradigm.

NSF MRI #0922657, $451,051, Taufer is co-PI, with Douglas Doren (PI), Sandeep Patel, Dionisios Vlachos.
Title: ***Acquisition of a Facility for Computational Approaches to Molecular-Scale Problems***
Duration: September 15, 2009 - September 14, 2012
Description: Support the acquisition of a hybrid-computing cluster, with GPU-accelerated computing nodes, for theoretical and experimental researchers at UD to study a number of problems in chemical sciences.

Conclusions

Over the past four years (2009-2013) this project has generated these main results:

(1) Algorithms for a realist and accurate representation of macro molecular systems and their dynamics on GPU-based high-end clusters, including an algorithm for Particle Mesh Ewald entirely performed on GPUs.
(2) An open-source GPU code called FEN ZI that can be downloaded from Google code and enables large-scale MD simulations on single GPUs. FEN ZI currently includes: NVT and NVE ensembles; the CHARMM force field; the Lennard-Jones interactions switching and shifting; long distance electrostatic interactions in terms of either reaction field or Ewald summation method including Particle Mesh Ewald (PME); explicit solvent with TIP3 or flexible SPC/Fw models.
(3) Fully atomistic molecular dynamics simulations of several molecular systems, such as the study of structural properties (i.e., atomic number density, electron density, and electrostatic potentials) of large DMPC lipid bilayers (on the order of a quarter million atoms); the interaction of a WALP16 peptide with a model DMPC lipid bilayer; the formation of sodium dodecyl sulfate (SDS) molecules in the presence of different types of salt concentrations; and the energetics of carbon nanotubes in aqueous and electrolyte solutions using the FEN ZI code on GPU clusters.
(4) A first prototype of a framework that enables higher utilization of GPUs while pursing coordinated progression of MD trajectories for non-dedicated, high-end GPU clusters.

The research has resulted in six papers in peer-reviewed journals and six papers in peer-reviewed conference and workshop venues. These results were also presented in sixteen talks (invited and conference talks).

Two post-doctoral researchers have worked on this project. Both have received faculty positions in research universities at the end of their research experience in Taufer's group. One high school student, one undergraduate student, two master students, and one PhD candidate have been working partially or fully sported by this project. The high school student was accepted to the Undergraduate Computer Science Program at the Worcester Polytechnic Institute. The UG student is still involved in research in Taufer's group and is spending Summer 2013 as a summer intern at Oak ridge National Lab supported by the DoE Science Undergraduate Laboratory Internship (SULI) program. The two master students graduated; one accepted a position at Philips and the other works now as a software developer in Taufer's group. The graduate student left the group after a short research experience.

Technology Transfer

The investigator has interacted with these ARL members as possible users of the code: Dale Shires, Margaret Hurley, and Michael S. Lee at the U.S. Army Research Laboratory at the Aberdeen Proving Ground, Maryland. In the past two years the investigator has meet with Dale Shires and Margaret Hurley. Taufer is working with Margaret Hurley to pass the MD code FEN ZI to her group. FEN ZI as been released in Google code as open-source code. She is also committed to share the framework prototype fo trajectory coordination with Dale Shires. A meeting was scheduled in May 2013 but had to be postponed to fall 2013 because of sudden commitments from both parties.

Future Plans

Future plans targeting technology transfer includes:

- Collaborate with Dale Share to tailor the framework for simulations and clusters at ARL in Aberdeen
- Collaborate with Margaret Hurley to complete the transfer and use of the FEN ZI code to her group

Taufer will explore the possibility to write a new project proposal with Share and Hurley targeting these two goals.

UNIVERSITY of DELAWARE

Computer-Aided Design of Drugs on Emerging Hybrid High Performance Computers – Final Report

Project Number 54723-CS

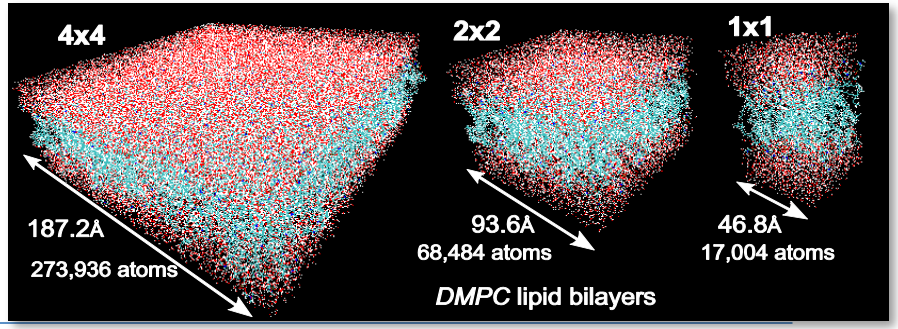
Michela Taufer
University of Delaware

Results from Year 2010-2011: FEN ZI: GPU-Enabled MD Simulations based on CHARMM Force Field and PME

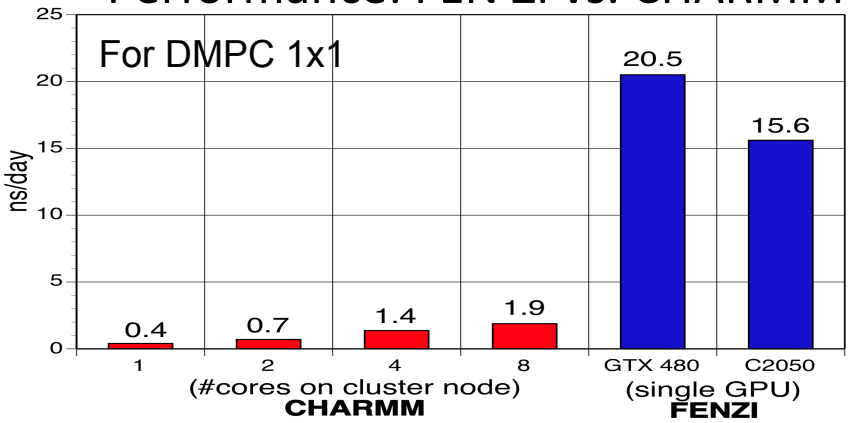
- Use FEN ZI, a MD GPU-based code for NVT, NVE, and NTP ensembles including PME, to study:
  - Structural and electrostatic properties of DMPC lipid bilayers membranes
  - Pathological conditions and behaviors of protein-membrane interactions (**work in progress on Keeneland**)

URL: http://gcl.cis.udel.edu/projects/fenzi/

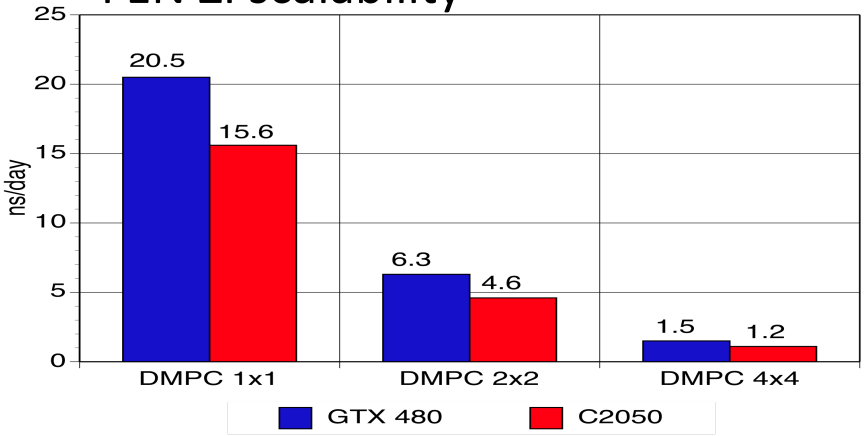
- GPUs enable fast simulations of larger membranes over longer simulated times (>100ns)



- Performance: FEN ZI vs. CHARMM



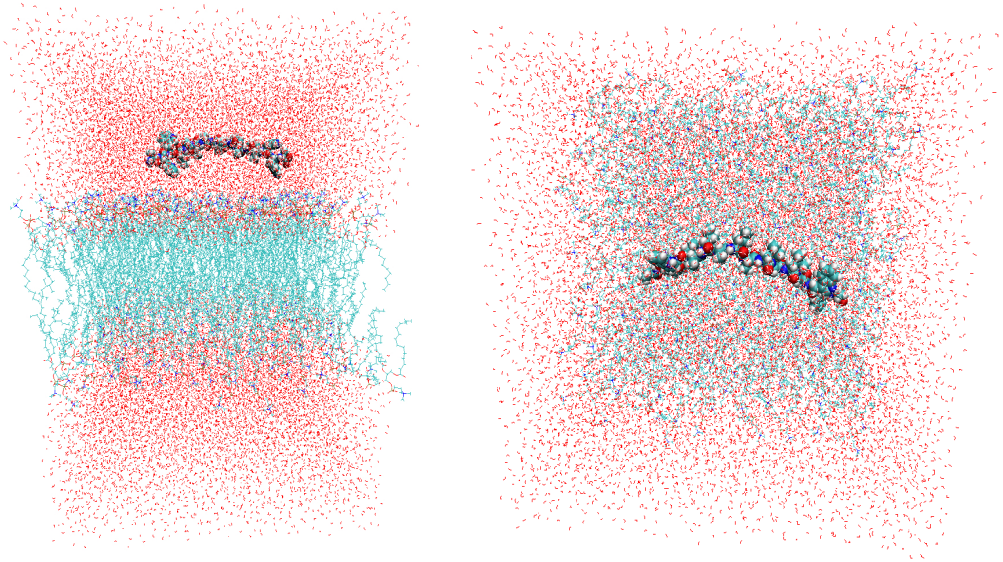
- FEN ZI scalability



N. Ganesan, B. Bauer, S. Patel, and M. Taufer, *J. Comp. Chemistry* 32(14):2958-2973

Results from Year 2011-2012: WALP16 Interacting with DMPC Membranes

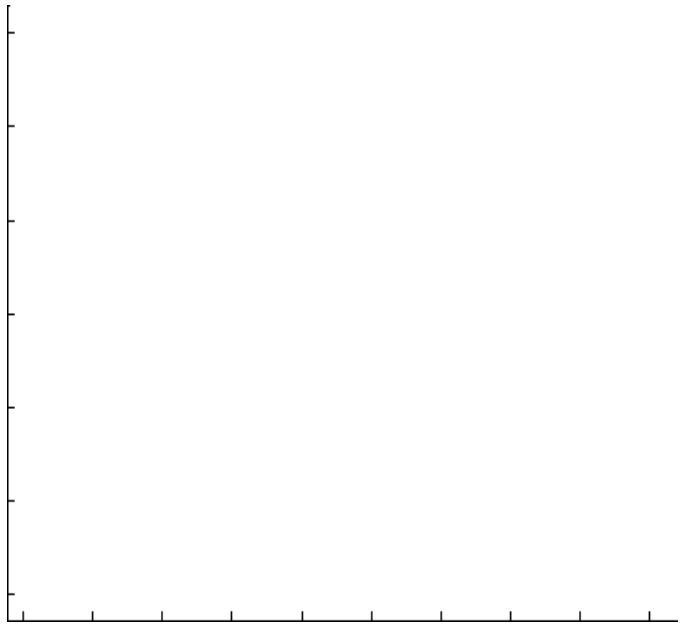
DMPC lipid bilayers and interacting peptide (different perspectives)



Side view

Top level view

Map of peptide-membrane surface interaction



Larger size scale membranes over longer simulated time scales on GPU cluster makes the scooting phenomena observable

M. Taufer, N. Ganesan, and S. Patel, CiSE, 2013.

Red regions: diffusion while peptide external to membrane

Blue regions: peptide directly interacting with lipid bilayer atoms

Year 2012-2013: Taxonomy of simulations

- Simulations applying fully atomistically resolved molecular models and force fields
  - GPUs enable longer time and space scales
- Variable job lengths (ns/day):
  - As a trajectory evolves
  - Across trajectories with different e.g., concentrations
- Fully or partially coordinated simulation progression:
  - Fully coordinated needed for e.g., replica-exchange molecular dynamics (REMD)
  - Partially coordinated for e.g., SDS and nanotubes systems

Constraints on high–end computer systems

- Resource constraints on high-end clusters:
  - Limited wall-time limit per job (e.g., 24 hours)
  - Mandatory use of resource managers
  - No direct submission and monitoring of GPU jobs
- Logical GPU job does not map to physical GPU job
  - Workflow managers still in infancy
- System and application failures on GPUs are undetected
  - Resource managers remain with no notion of job terminations on GPUs

Moving beyond virtualization

- When clusters **do include virtualization**
  - E.g., Shadowfax
- We can schedule isolated CPU/GPU pairs
  - This allows us to associate failures with a specific GPU
- Virtualization imposes overheads
  - Power
  - Performance
  - Noise or jitter
  - Portability and maintainability

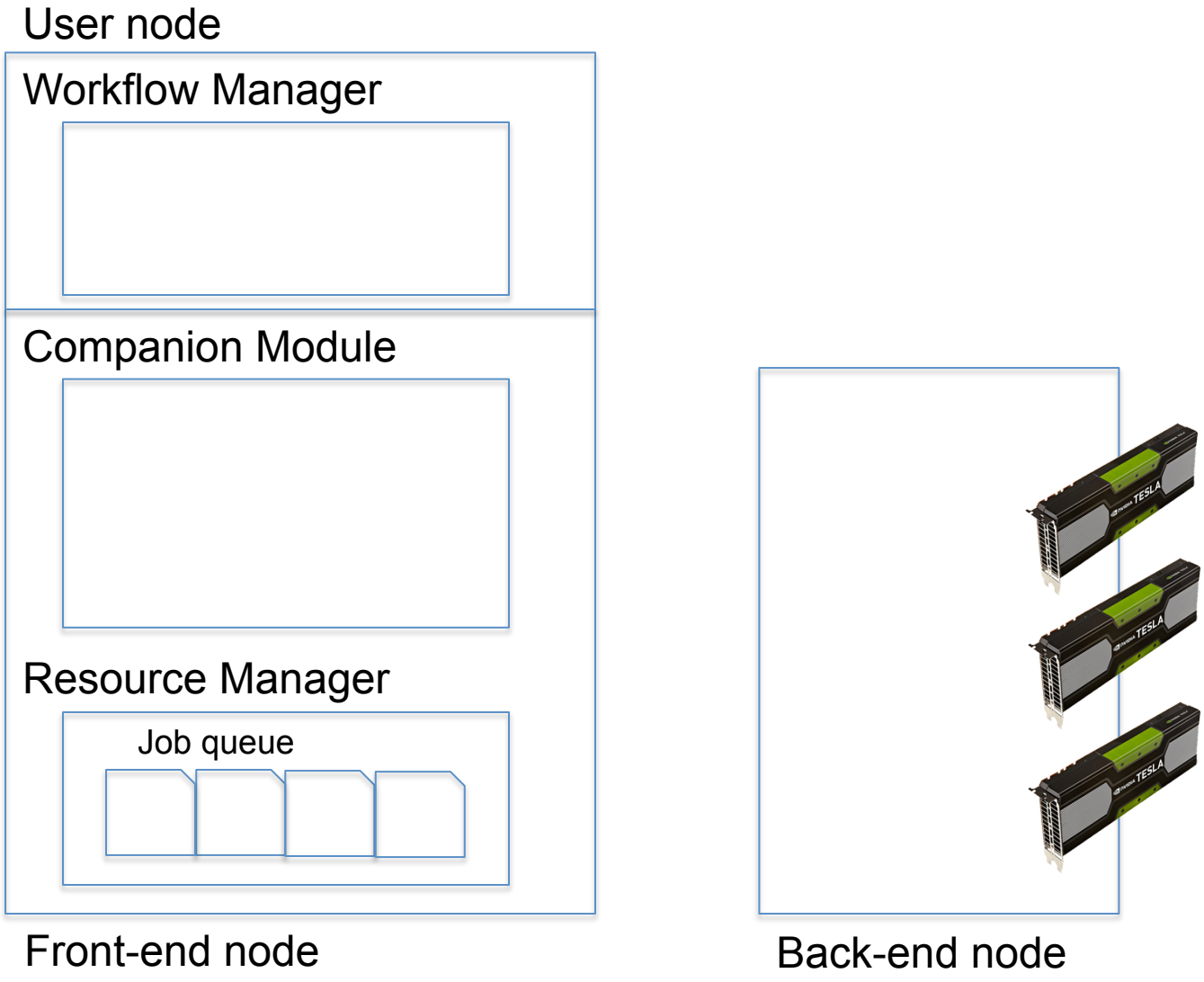
… and may not be available

Our goal: Pursuing BOTH high accelerators' utilization and (fully or partially) coordinated simulations' progression on GPUs in effective and cross-platform ways

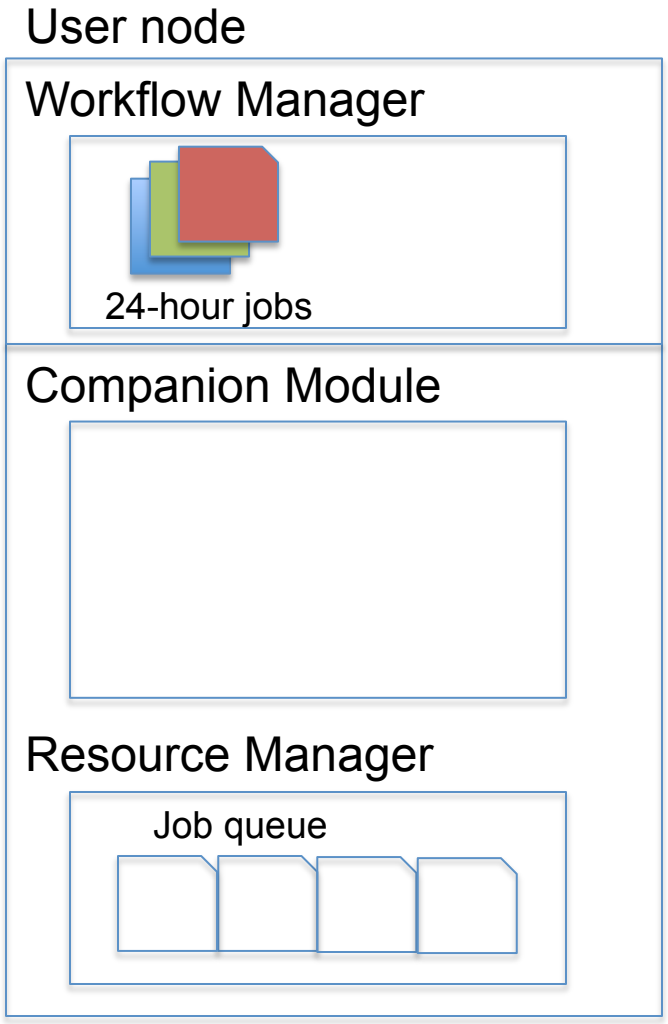
Our approach

- Two software modules that plug into existing resource managers and workflow managers
  - No virtualization to embrace diverse clusters and programming languages
- A companion module:
  - Runs on the head node of the cluster
  - Accepts jobs from workflow manager
  - Instantiates "children" wrapper modules
  - Dynamically splits jobs and distributes job segments to wrapper modules
- A wrapper module:
  - Launches on compute node as a resource manager job
  - Receives and runs job segments from companion module
  - Reports status of job segments to companion module

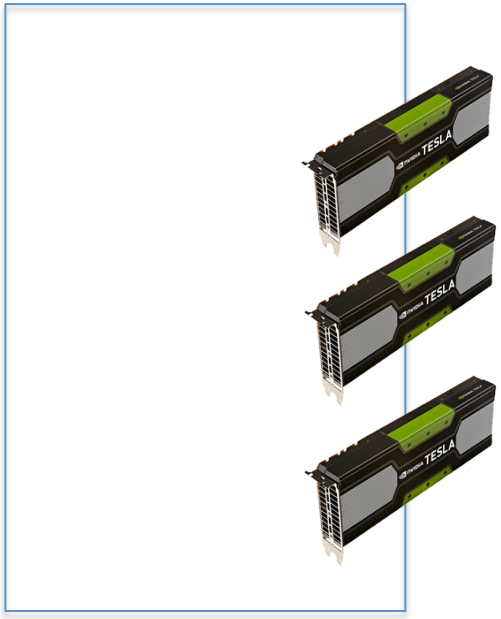
Modules in action



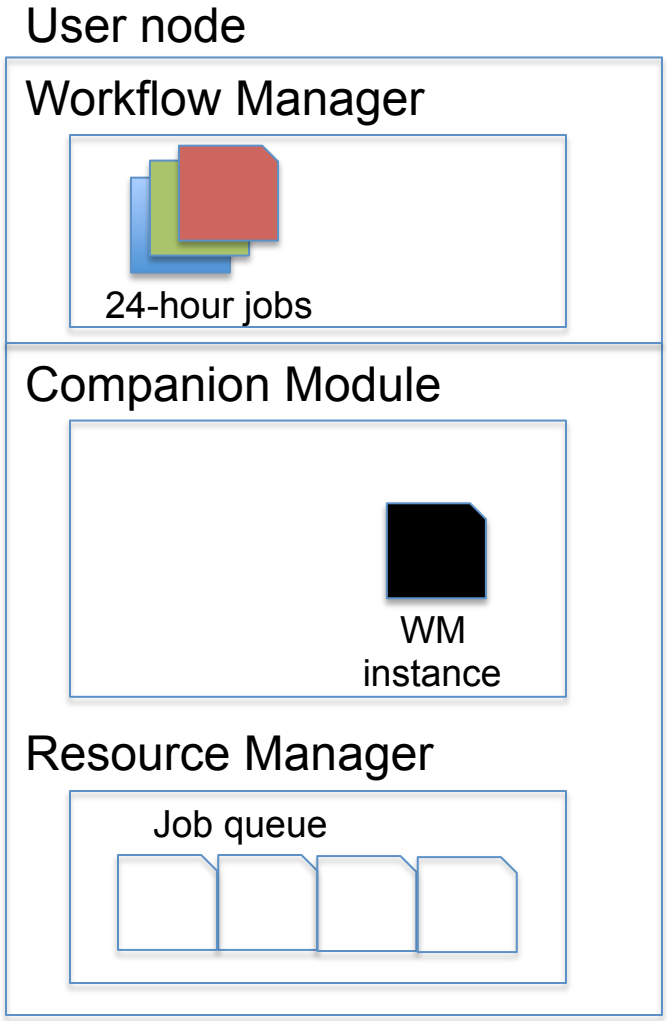
Modules in action



Workflow Manager:
- generate set of 24-hour jobs

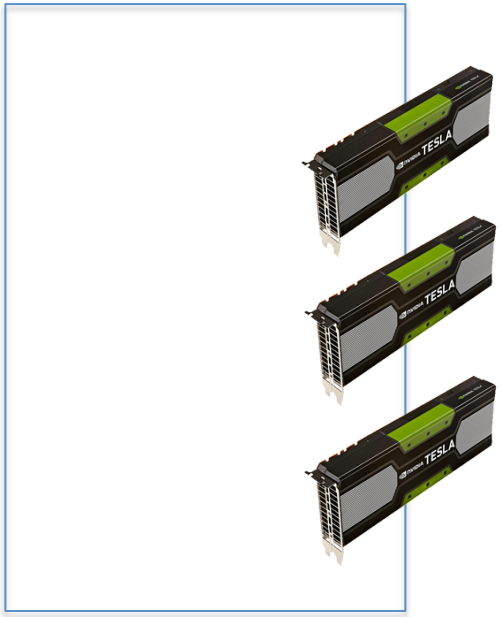


Modules in action

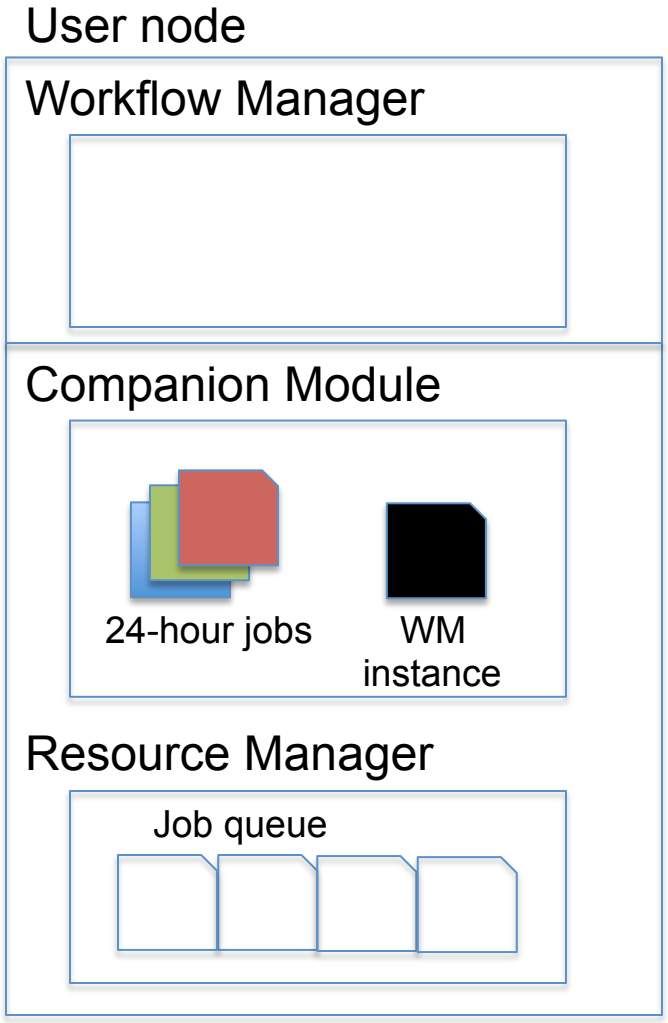


Workflow Manager:
- send set of 24-hour jobs to companion module

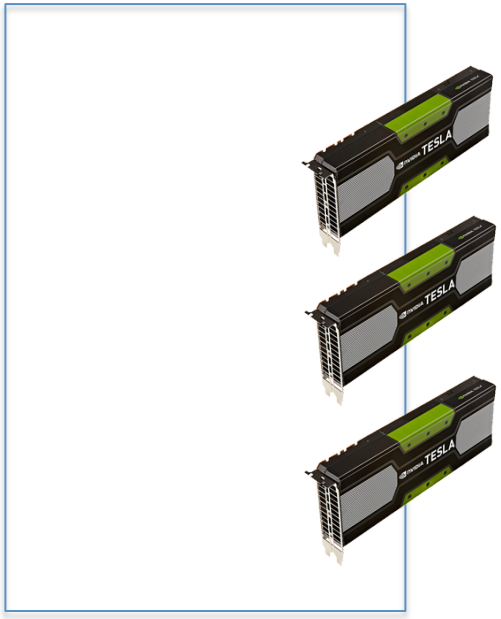
Companion Module:
- receive 24-hour jobs
- generate a Wrapper Module (WM) instance per back-node



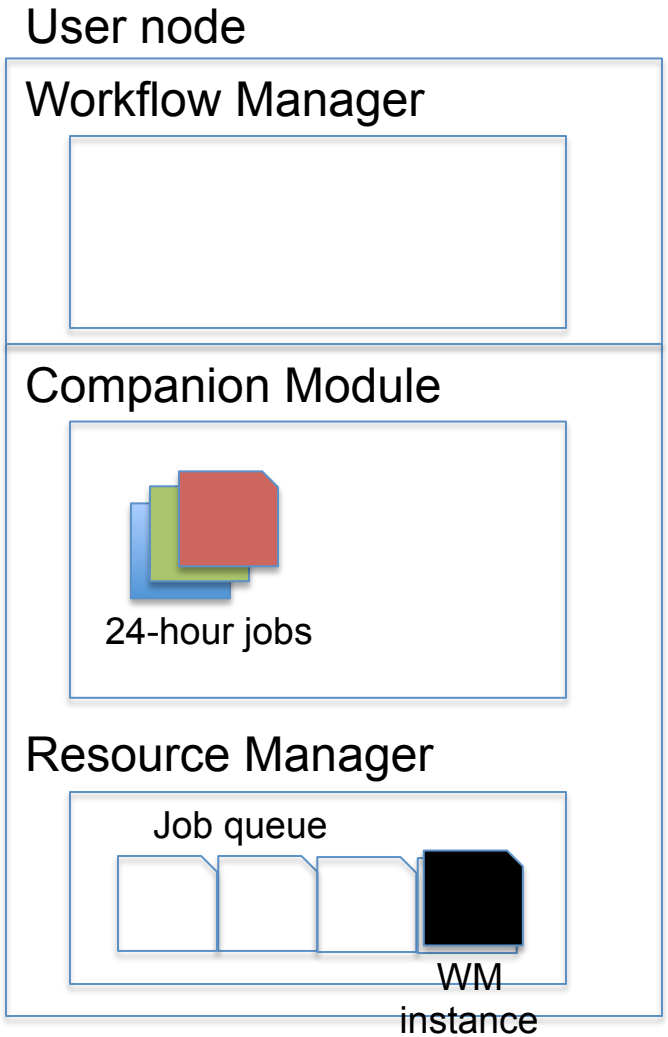
Modules in action



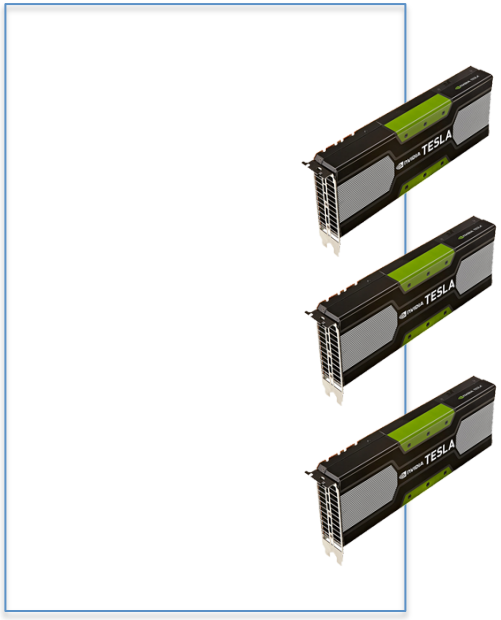
Companion Module:
- submit WM instance as a job to resource manager



Modules in action



Companion Module:
- submit WM instance as a job to resource manager



Modules in action

User node

Workflow Manager

Companion Module

24-hour jobs

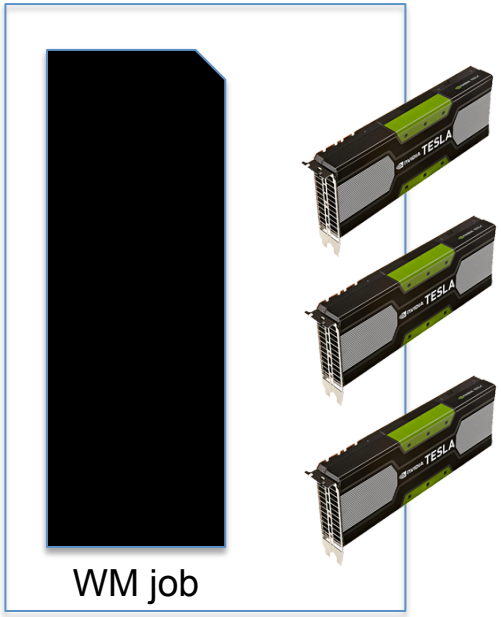
Resource Manager

Job queue

WM instance

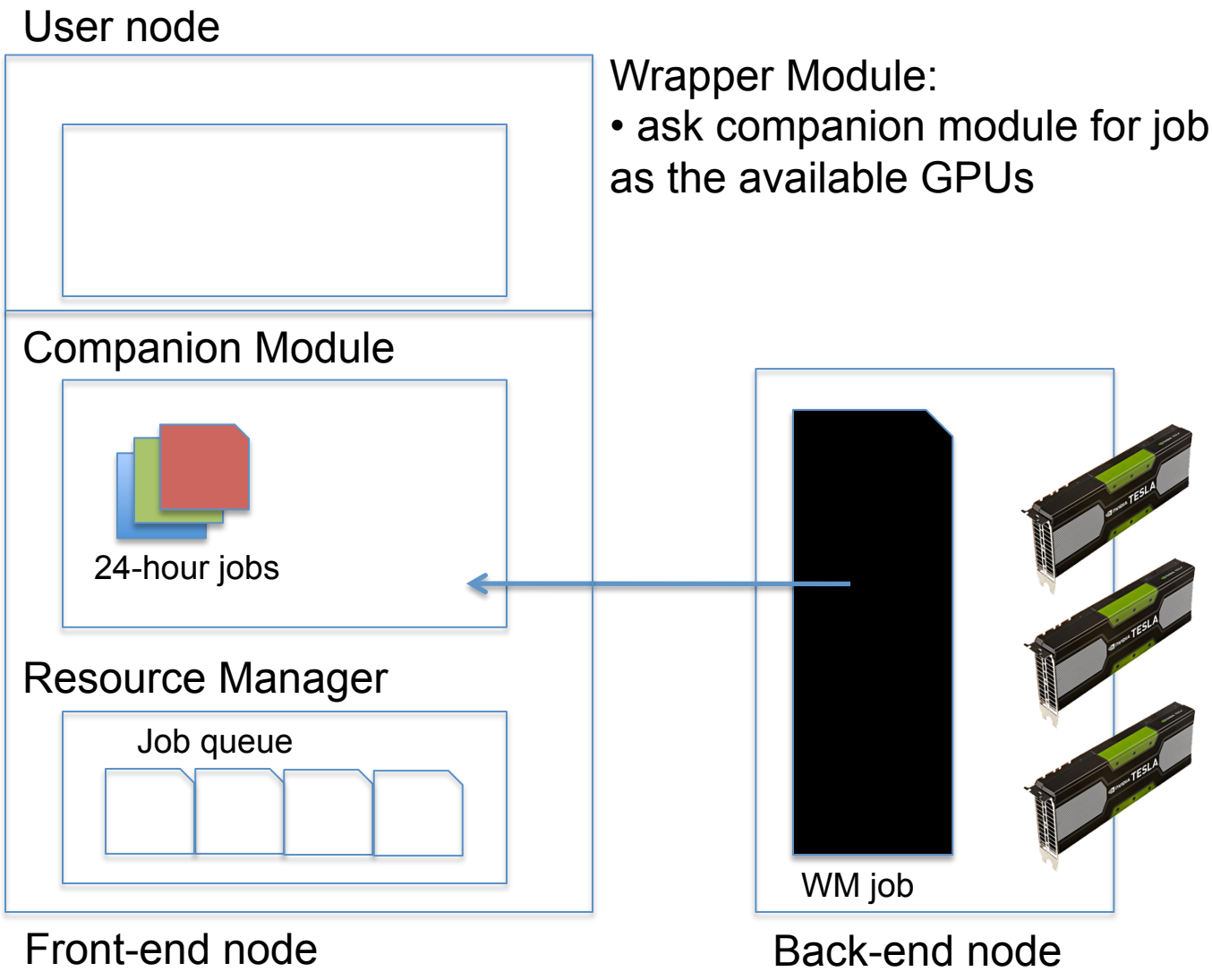
Front-end node

Resource Manager:
• launch WM instance as a job on back-end node



Back-end node

Modules in action



Wrapper Module:

- ask companion module for job segments, as many as the available GPUs

Modules in action

User node

Workflow Manager

Companion Module

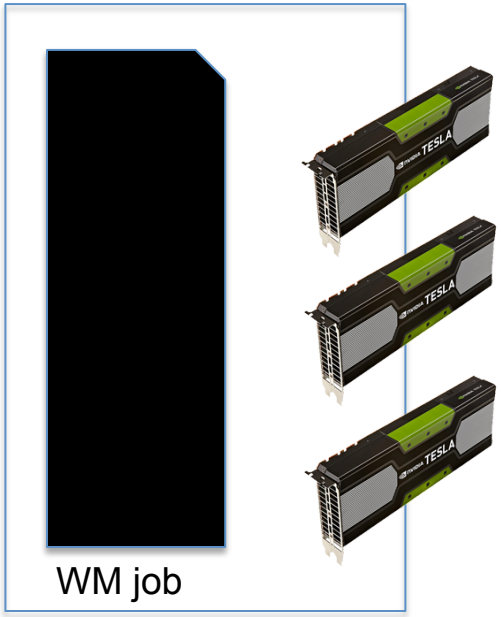
24-hour jobs

Resource Manager

Job queue

Front-end node

Companion Module:
• fragment jobs into 6-hour subjobs



Back-end node

Modules in action

User node

Workflow Manager

Companion Module

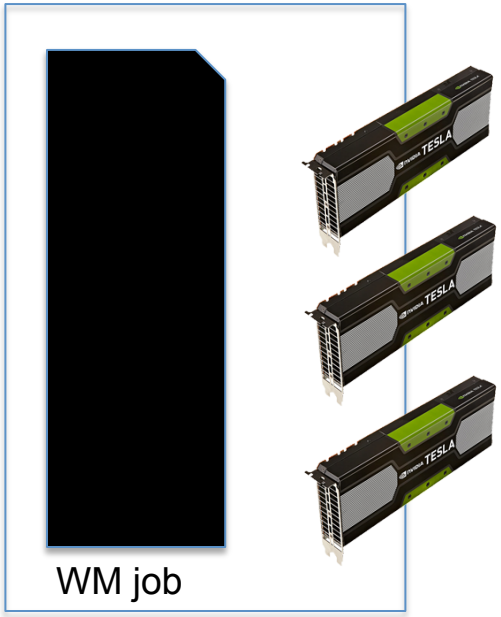
24-hour jobs

Resource Manager

Job queue

Front-end node

Companion Module:
- fragment jobs into 6-hour subjobs
- send bundle of 3 subjobs to WM job



Back-end node

Modules in action

User node

Workflow Manager

Companion Module

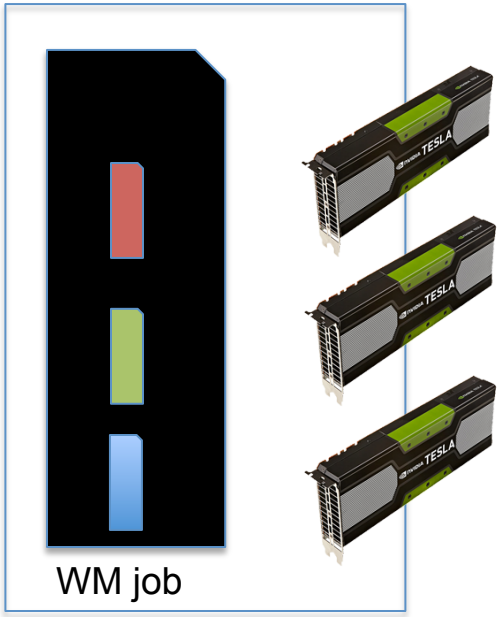
24-hour jobs

Resource Manager

Job queue

Front-end node

Companion Module:
- fragment jobs into 6-hour subjobs
- send bundle of 3 subjobs to WM job



WM job

Back-end node

Modules in action

User node

Workflow Manager

Companion Module

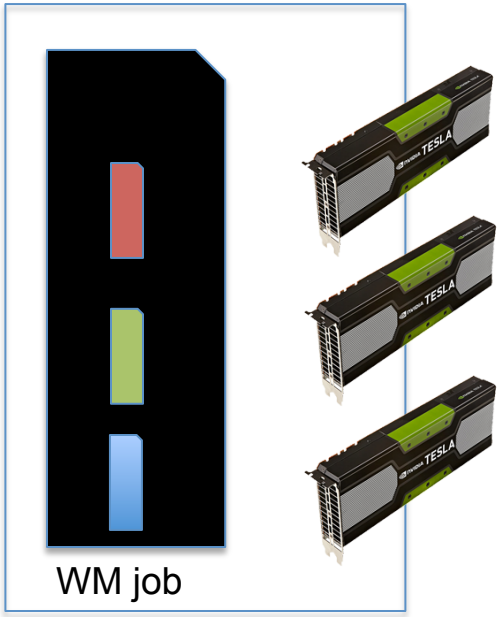
24-hour jobs

Resource Manager

Job queue

Front-end node

Wrapper Module:
- instantiate subjobs on GPUs
- monitor system and application failures as well as time constraints



Back-end node

Modules in action

User node

Workflow Manager

Companion Module

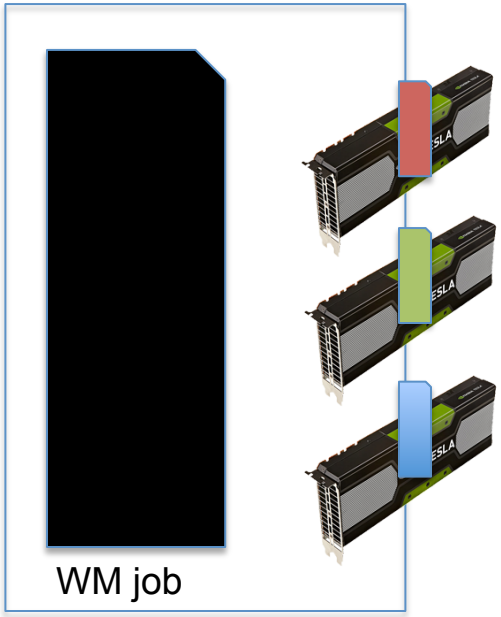
24-hour jobs

Resource Manager

Job queue

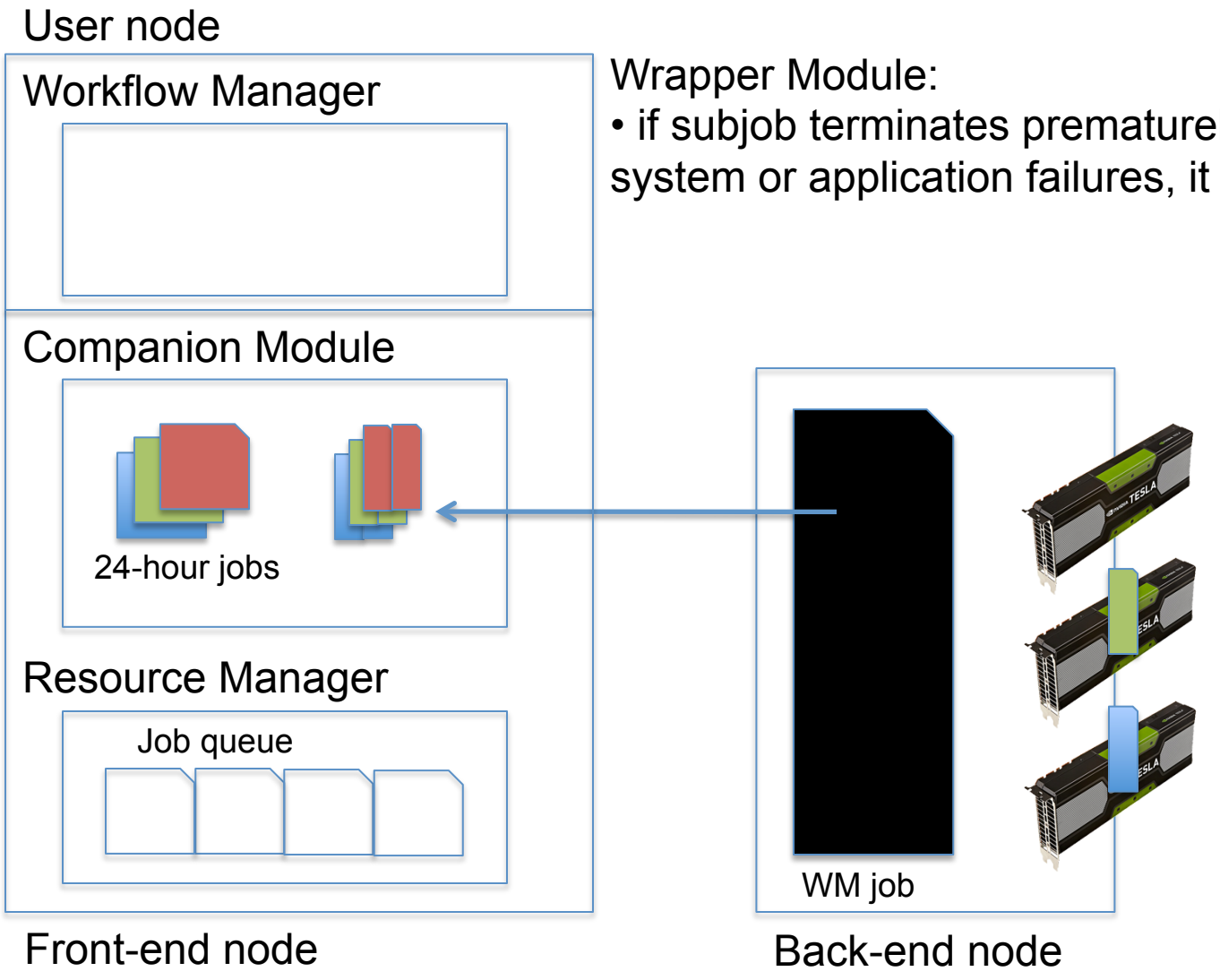
Front-end node

Wrapper Module:
- instantiate subjobs on GPUs
- monitor system and application failures as well as time constraints



Back-end node

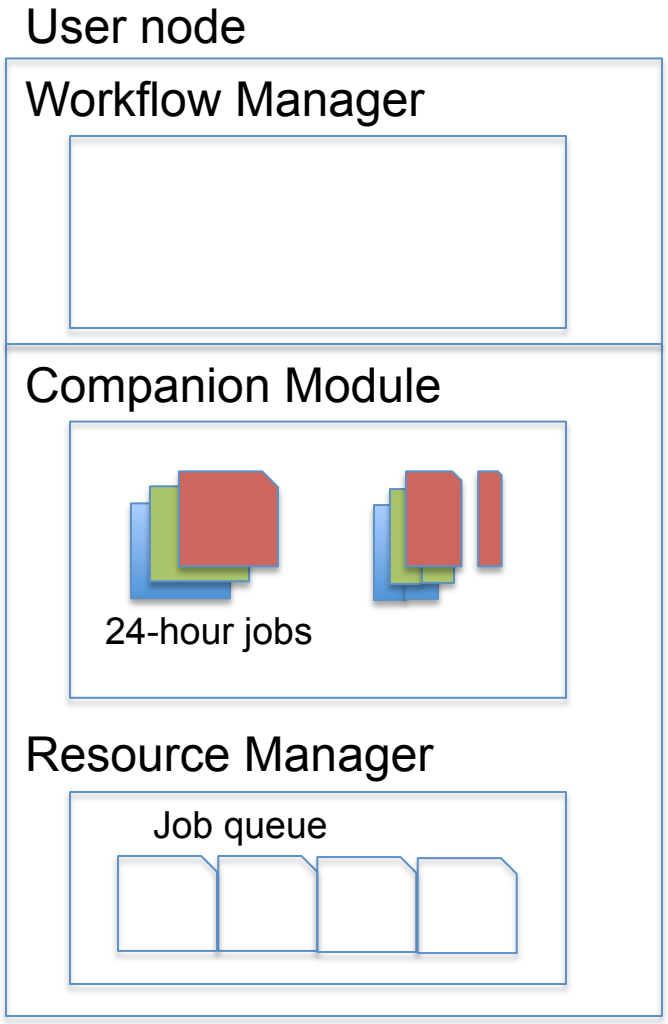
Modules in action



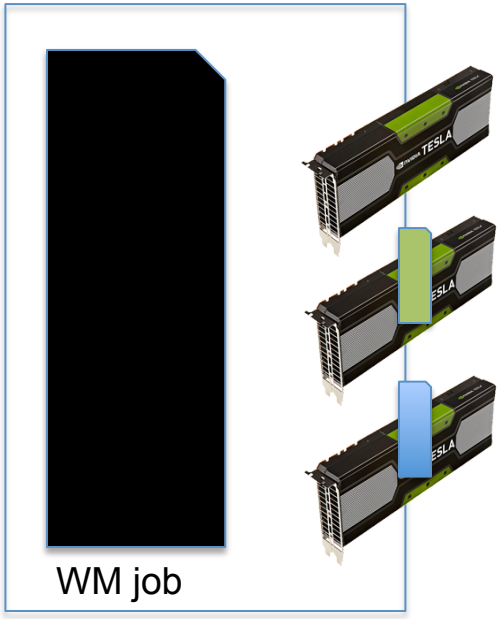
Wrapper Module:

- if subjob terminates prematurely because of e.g., system or application failures, it request new subjob

Modules in action



Companion Module:
- adjust length of new subjob based on heuristics, e.g., to complete initially 6-hour period
- send subjob to wrapper module for execution



Modules in action

User node

Workflow Manager

Companion Module

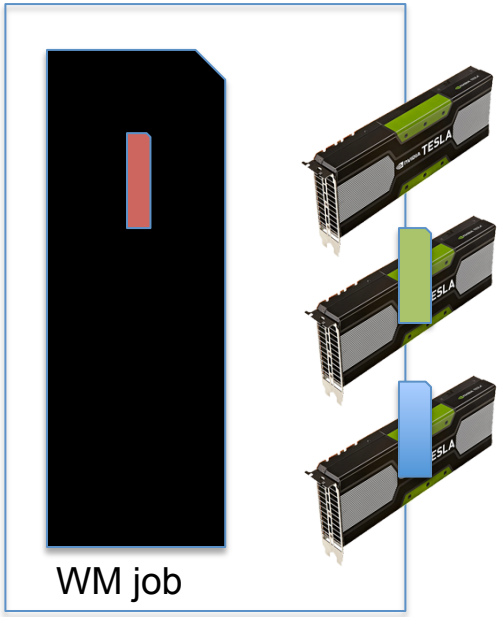
24-hour jobs

Resource Manager

Job queue

Front-end node

Companion Module:
• adjust length of new subjob based on heuristics, e.g., to complete initially 6-hour period
• send subjob to wrapper module for execution



Back-end node

MD Simulations

- MD simulations:
  - Case study 1: Study of sodium dodecyl sulfate (SDS) molecules aqueous solutions and electrolyte solutions
  - Case study 2: Study of nanotubes in aqueous solutions and electrolyte solutions
- GPU code FEN ZI (*Yun Dong de FEN ZI = Moving MOLECULES)*
  - MD simulations in NVT and NVE ensembles and energy minimization in explicit solvent
  - Constraints on interatomic distances e.g., shake and rattle, atomic restraints, and freezing fast degrees of motions
  - Electrostatic interactions, i.e., Ewald summation, performed on GPU
- Metrics of interest:
  - Utilization of GPUs – i.e., time ratio accountable for simulation's progression

The Keeneland system

- GPU description:
  - 3 M2090 GPUs per node
- Software:
  - TORQUE Resource Manager
  - Globus allows for the use of Pegasus Workflow Manager
  - Shared Lustre file system
- Constraints:
  - 24-hour time limit
  - 1 job per node (cannot have multiple jobs on one node)
  - Can set GPUs into Shared/Exclusive mode but not complete isolation (e.g., user that get access first can steal all the GPUs)
  - Vendor specific with specific version of NVIDIA driver (>260))

Modeling max utilization

- With our approach using n segments in 24-hour period:

$$utilization = \sum_{days} \sum_{GPUs} \frac{t_{\max} - \sum_{i=1}^{n-1} \left[\left(t_{arrival}(i) - t_{lastchk}(i) \right) + t_{restart} \right] - \left(t_{\max} - t_{arrival}(n) \right)}{t_{\max}}$$

- Without our approach:

$$utilization = \sum_{days} \sum_{GPUs} \frac{t_{\max} - \left(t_{arrival}(1) - t_{lastchk}(1) \right) - \left(t_{\max} - t_{arrival}(1) \right)}{t_{\max}}$$

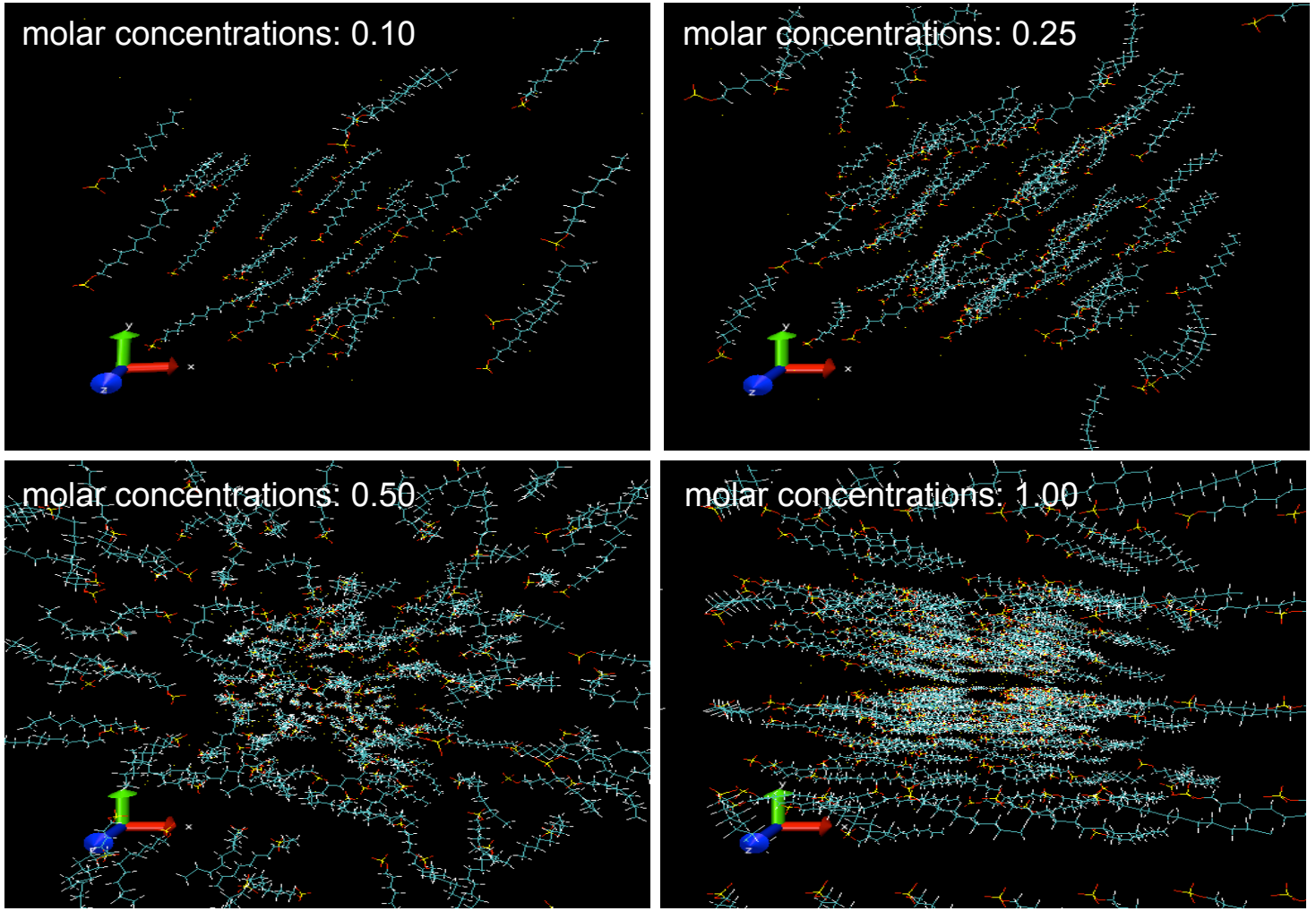
where:

$$t_{arrival}(i) = \begin{cases} t_{lastcheck}(i) & when \quad t_{arrival}(i) > t_{\max} \\ t_{arrival}(i) & otherwise \end{cases}$$

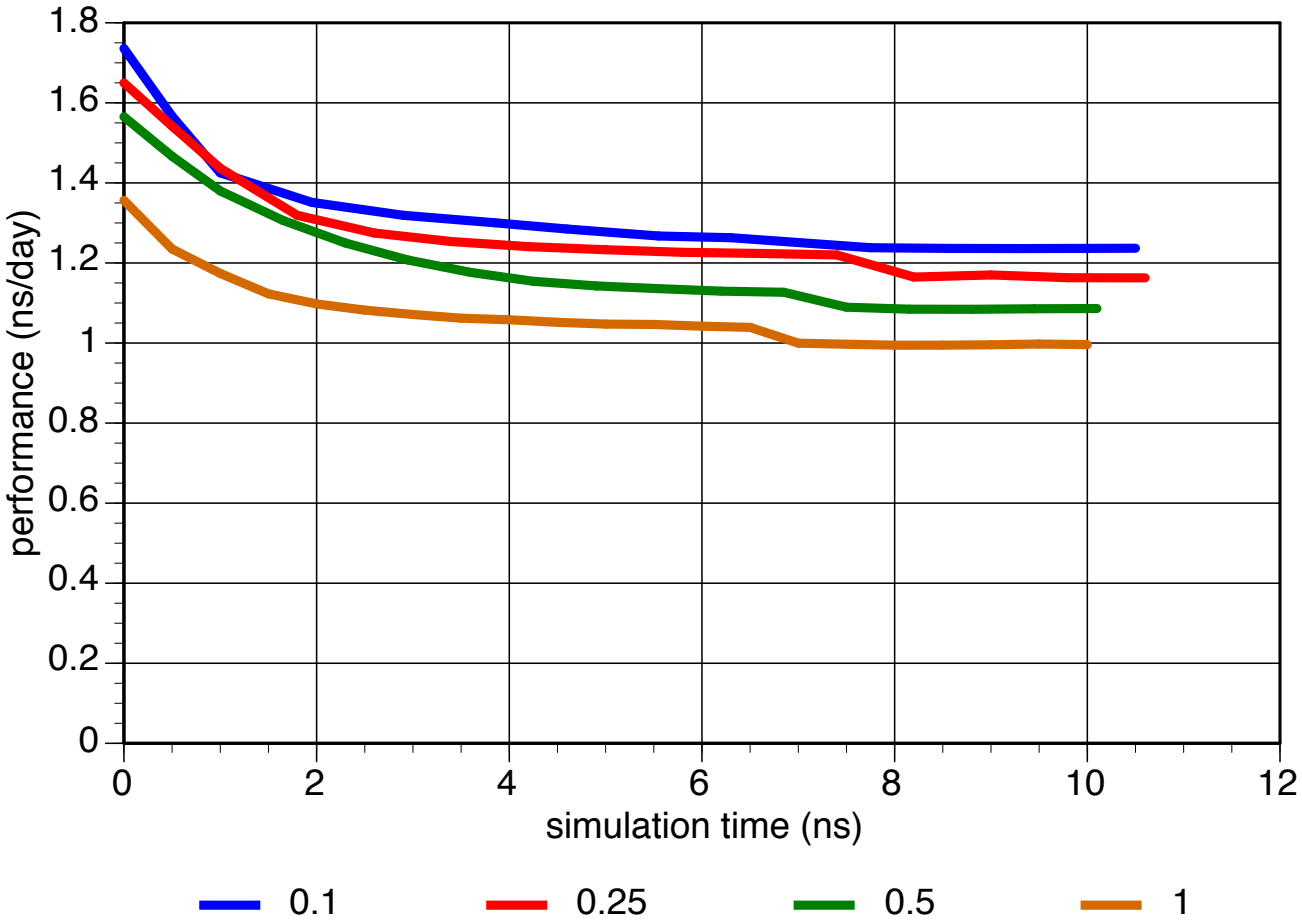
$$t_{lastchk}(n) = f(molecular\_system)$$

Case study 1: Sodium Dodecyl Sulfate (SDS)

Initial structures: surfactant molecules randomly distributed

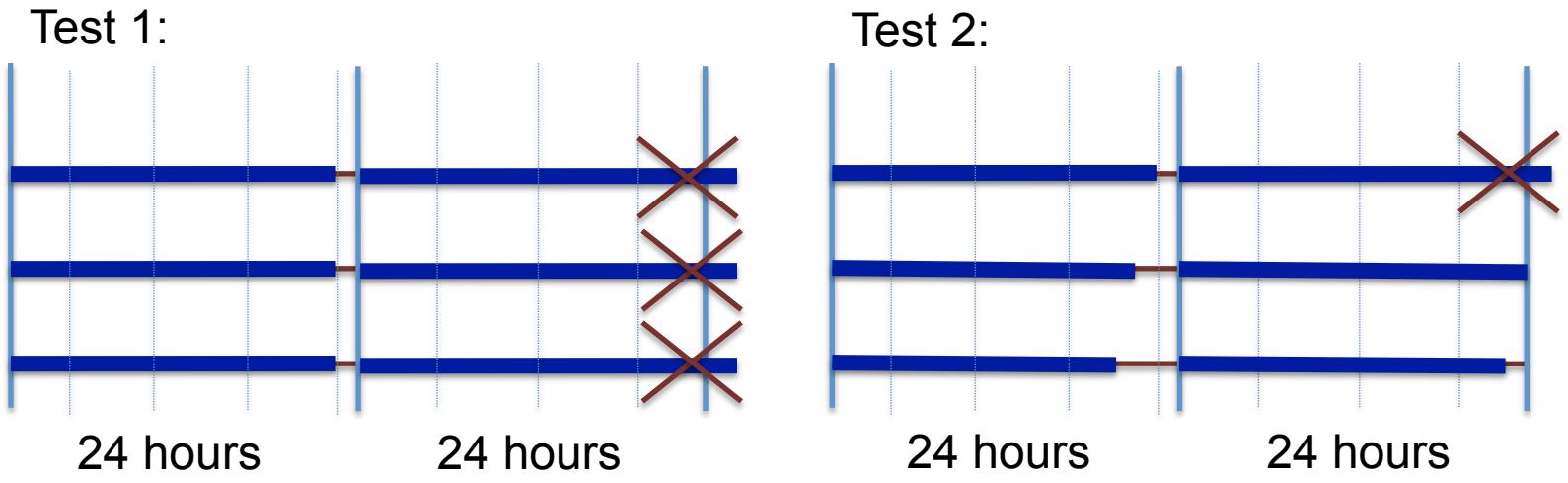


Case study 1: variable simulation times



Case study 1: testbeds

- Taxonomy of our simulations:
  - 4 concentrations and 3 200-ns trajectories per concentration at 298K
- Test 1:
  - Jobs with same concentrations assigned to same node
- Test 2:
  - Jobs with different concentrations assigned to same node



Case study 1: modeling max utilization

- With our approach using n segments in 24-hour period:

$$utilization = \sum_{days} \sum_{GPUs} \frac{t_{\max} - \sum_{i=1}^{n-1}\left[t_{restart}\right] - \left(t_{\max} - t_{arrival}(n)\right)}{t_{\max}}$$

- Without our approach:

$$utilization = \sum_{days} \sum_{GPUs} \frac{t_{\max} - \left(t_{\max} - t_{arrival}(1)\right)}{t_{\max}}$$

where:

$$t_{arrival}(i) = \begin{cases} t_{lastcheck}(i) & when \quad t_{arrival}(i) > t_{\max} \\ t_{arrival}(i) & otherwise \end{cases}$$

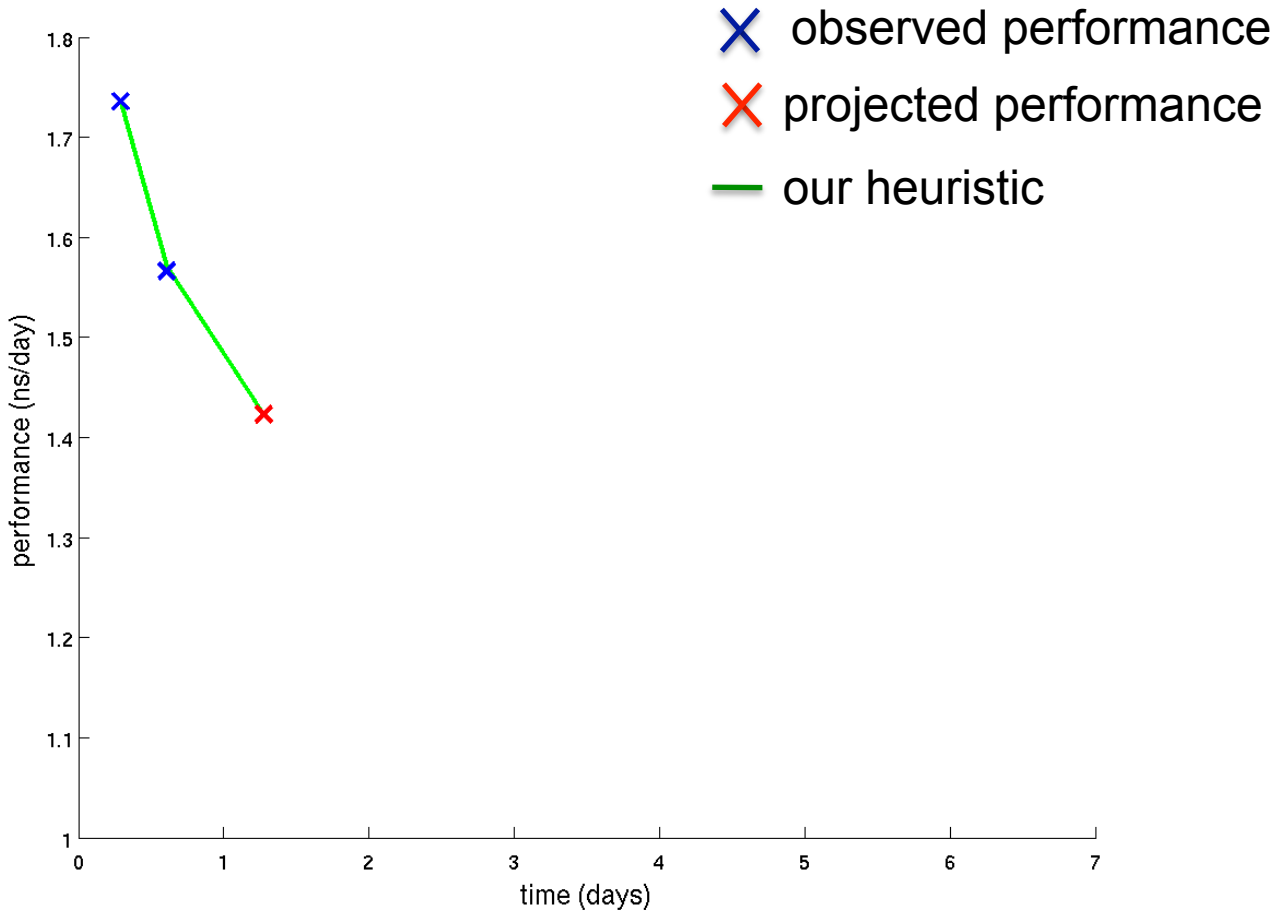
$$t_{\max} = 24 hours$$

Case study 1: modeling arrival time

We model $t_{arrival}(i)$ in two ways:

- Scientists: run short simulation, compute ns/day, define job's speed to constant rate to fit into 24-hour period
- Our approach: segment 24-hour job in segments, adjust segment length based on heuristic that takes into account change in ns/day

Case study 1: our heuristic



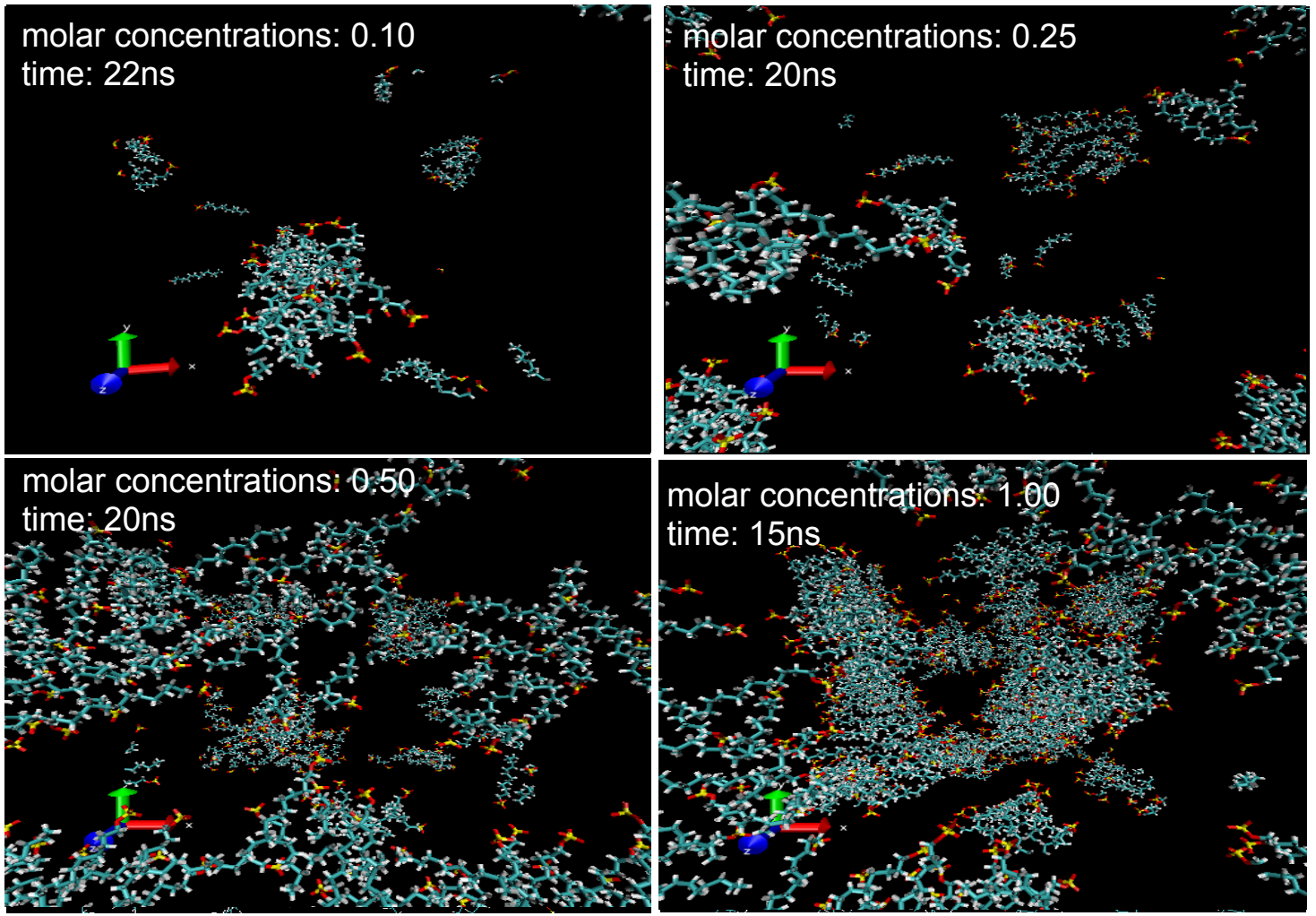
Case study 1: results

- Run 12 10-day trajectories with 4 concentrations and 3 different seeds on Keeneland, three trajectories per node

t_{chkpnt} (hours)	With our approach		W/o our approach	
	test 1	test 2	test 1	test 2
0.5	99.54%	98.82%	98.78%	97.08%
1	99.18%	98.44%	98.26%	96.53%
3	97.83%	96.98%	96.21%	94.49%
6	95.83%	94.85%	93.50%	91.72%

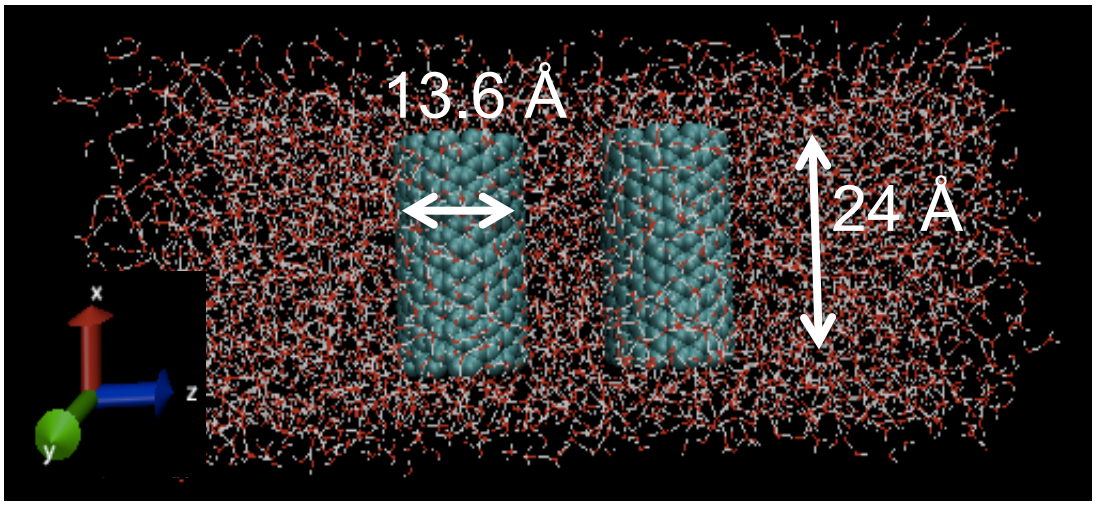
Case study 1: snapshots of ongoing simulations

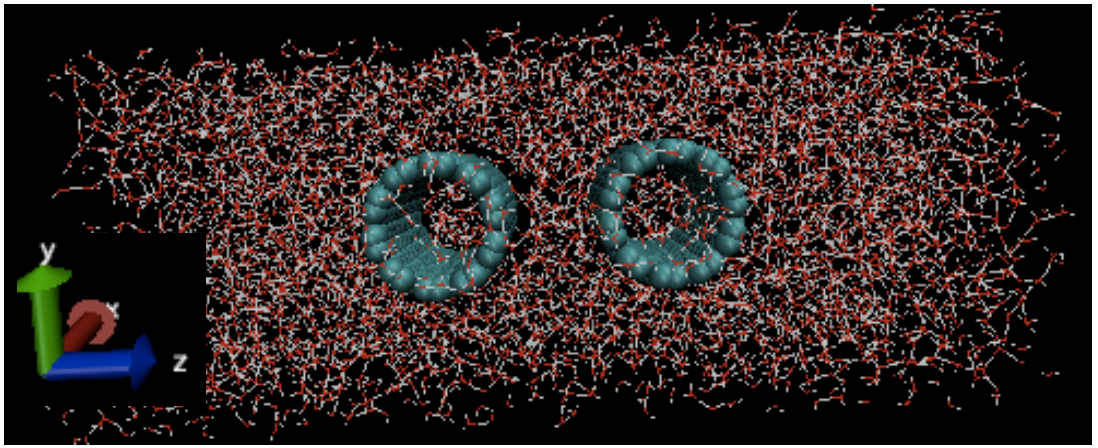
Initial structures: surfactant molecules randomly distributed



Case study 2: Carbon Nanotubes

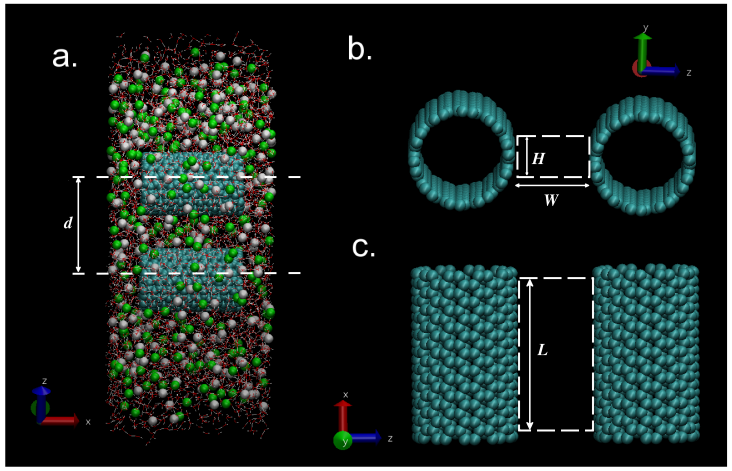
- Study nanotubes in aqueous solutions and electrolyte solutions
  - Different temperatures
  - Different separations
- Scientific metrics:
  - Potential of mean force
  - Effect of electrolytes, ie, sodium chloride and iodide
  - Ion spatial distributions

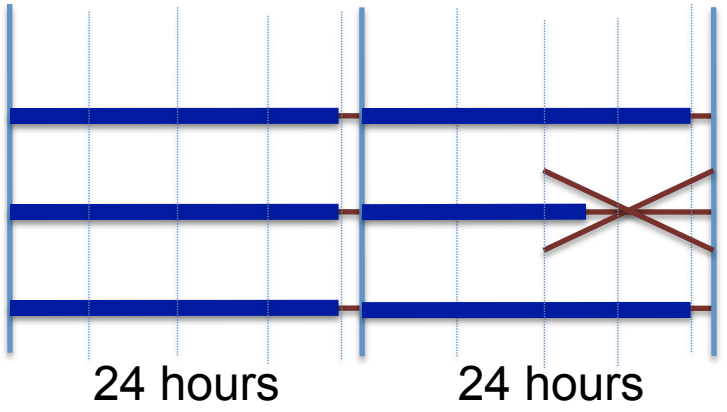




Case study 2: testbeds

- Taxonomy of the simulations:
  - 10 temperatures ranging from 280K to 360K along with 20 tube separations
  - 200ns per trajectory with 5.8ns+/-3% per day on 64 nodes
- Test 1:
  - Hardware errors, i.e., ECC error and system failures
- Test 2:
  - Hardware and application errors





Modeling max utilization

- With our approach:

$$utilization = \sum_{days} \sum_{GPUs} \frac{t_{\max} - \sum_{i=1}^{n-1}\left[\left(t_{arrival}(i) - t_{lastchk}(i)\right) + t_{restart}\right] - \left(t_{\max} - t_{arrival}(n)\right)}{t_{\max}}$$

- Without our approach:

$$utilization = \sum_{days} \sum_{GPUs} \frac{t_{\max} - \left(t_{arrival}(1) - t_{lastchk}(1)\right) - \left(t_{\max} - t_{arrival}(1)\right)}{t_{\max}}$$

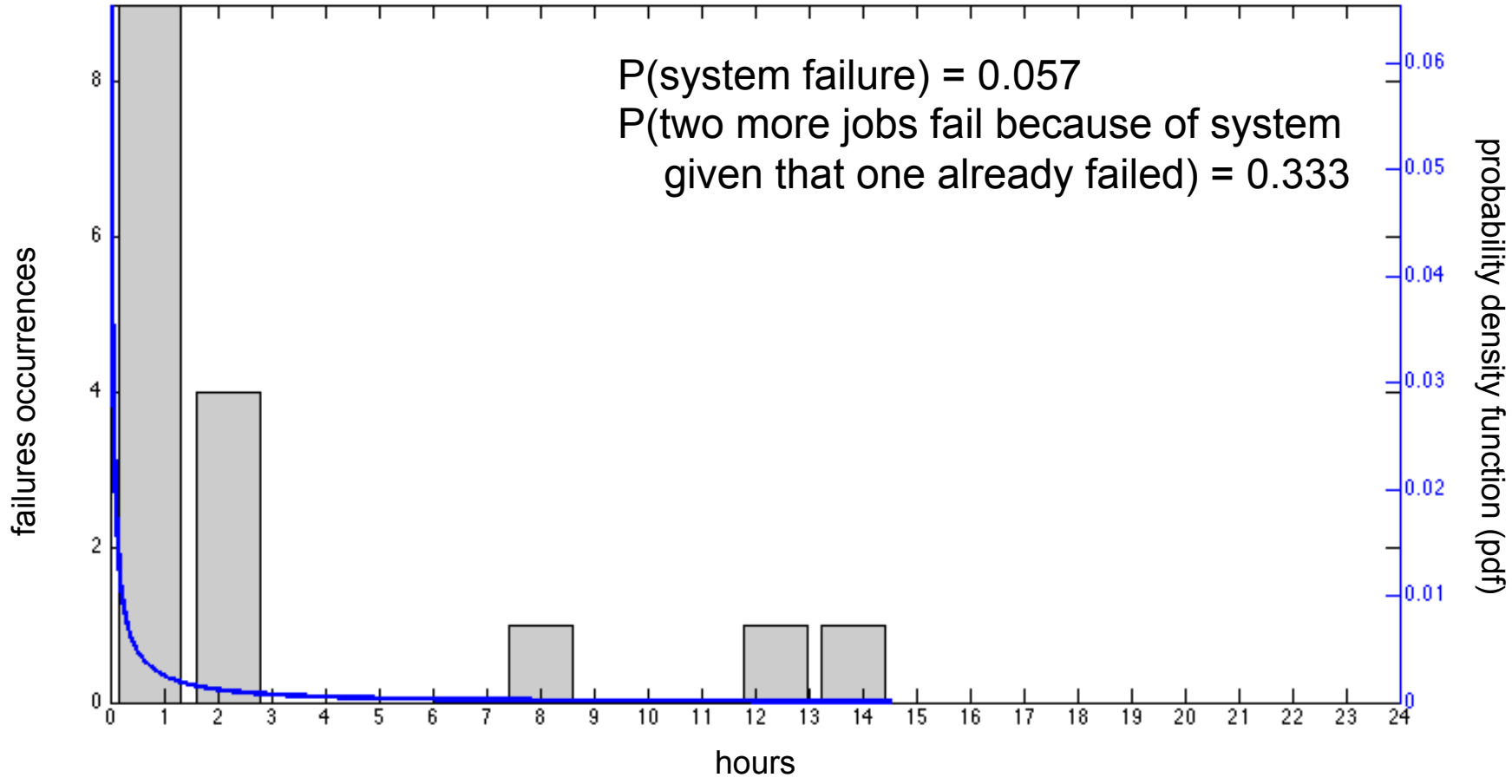
where:

$$t_{arrival}(i)_{i<n} = weilbul(scale, shape)$$

$$t_{arrival}(n) = 0.03 \times t_{\max}$$

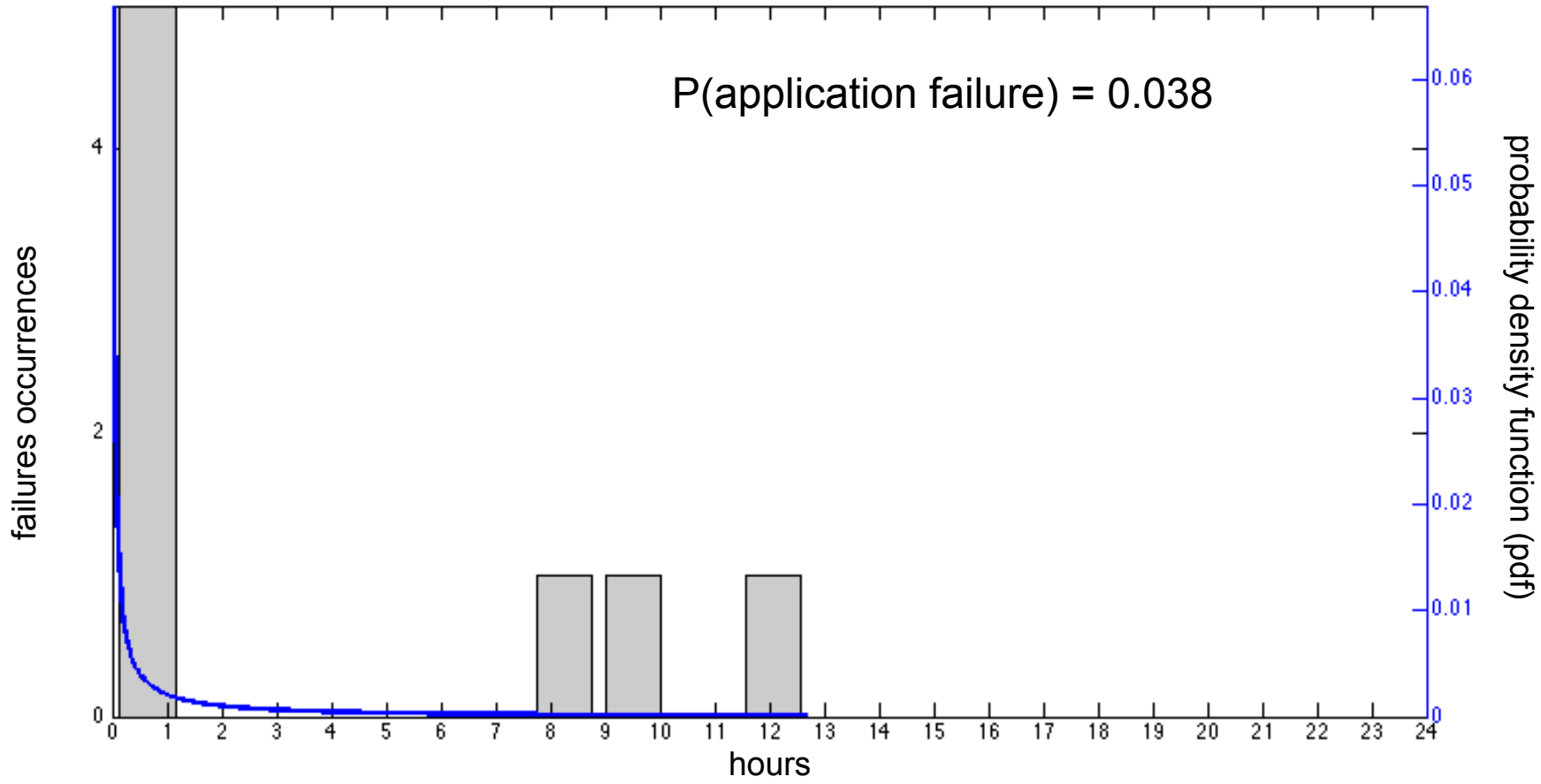
$$t_{\max} = 24 hours$$

Case study 2: modeling system failures



- Weibul distribution: scale = 203.8 and shape = 0.525

Case study 2: modeling application failures



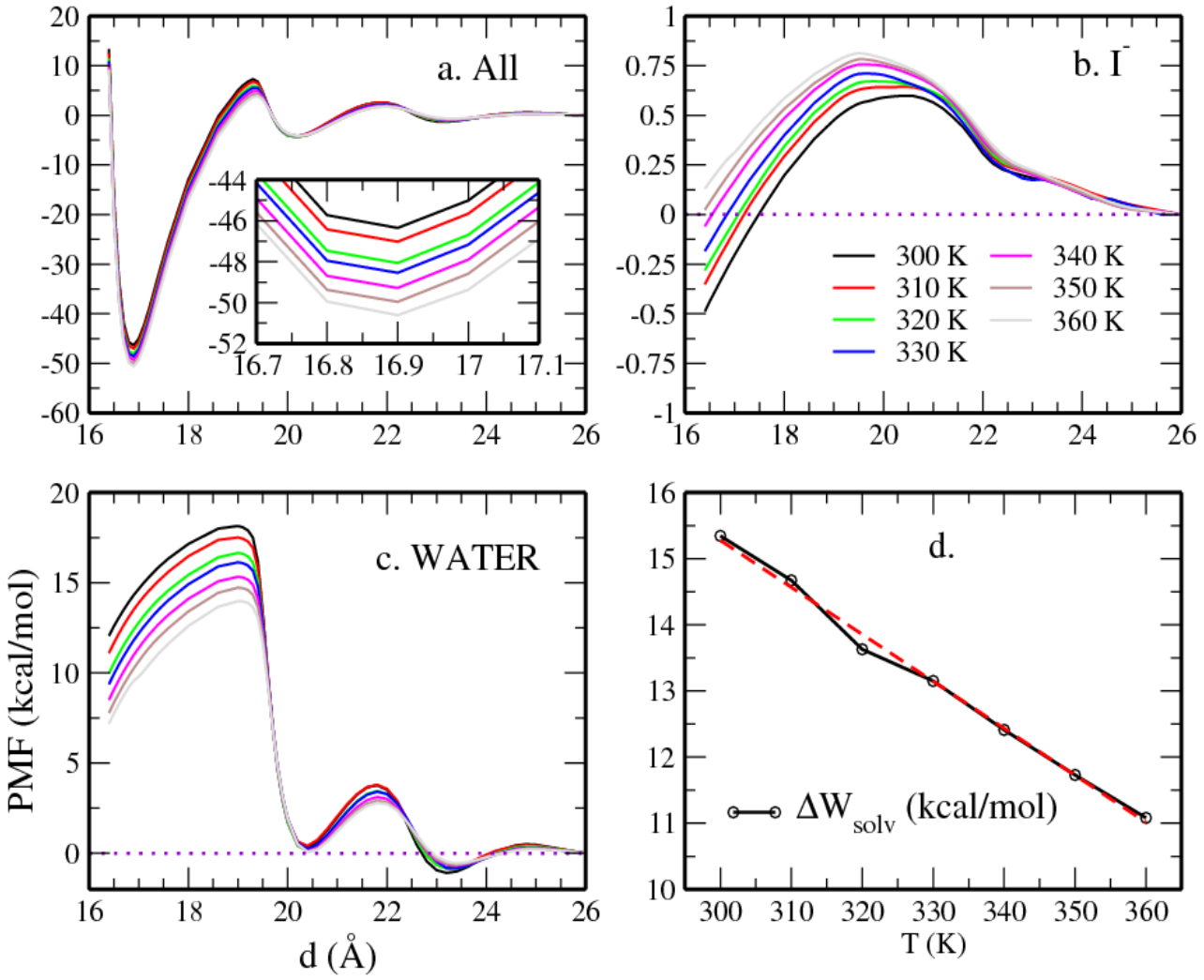
- Weibul distribution: scale = 56.56, shape = 0.3361

Case study 2: results

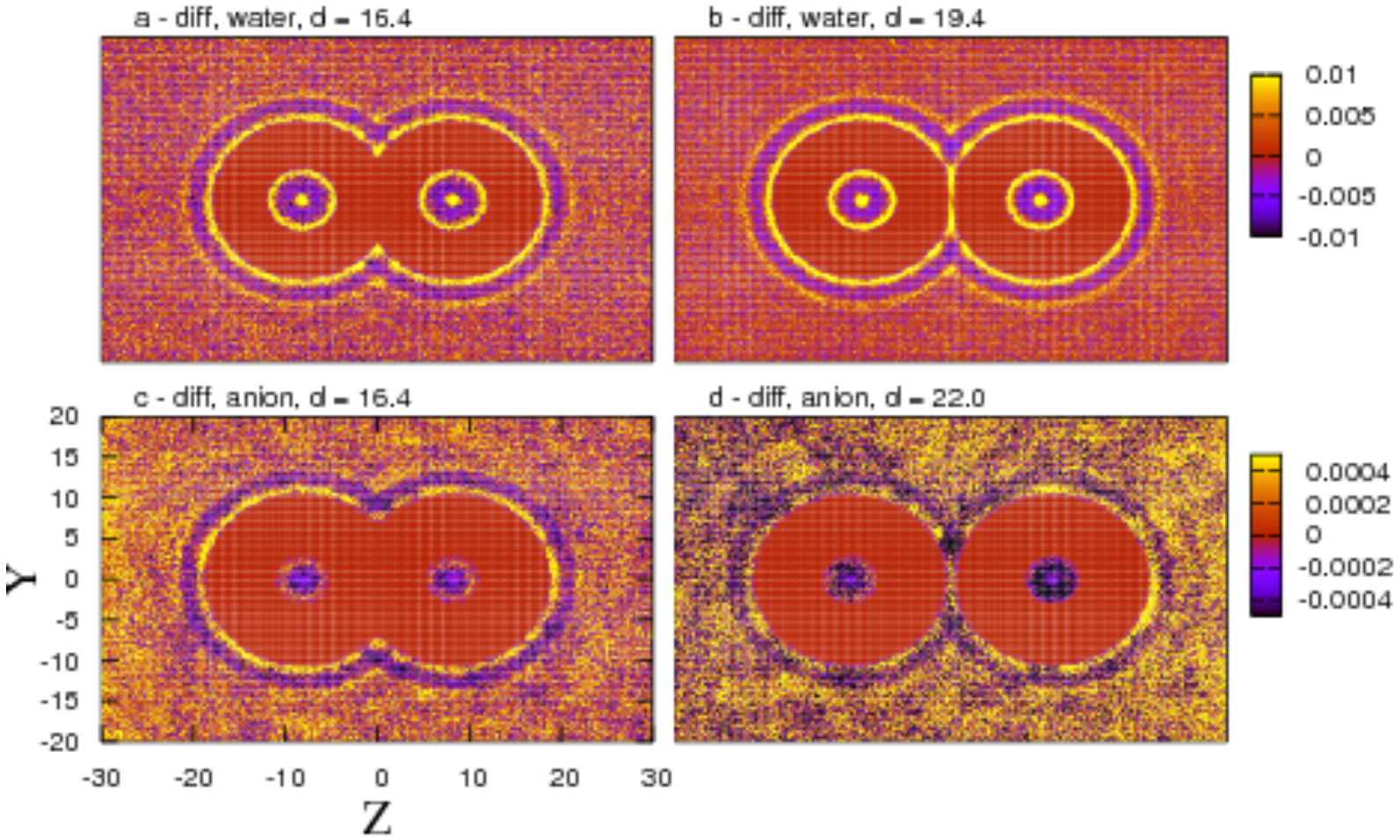
- Run 200ns for each nanotube system – equivalent to ~35 days on 64 nodes of Keeneland, each with 3 GPUs

t_{chkpnt} (hours)	With our approach		W/o our approach	
	sysfail	sysfail appfail	sysfail	sysfail appfail
0.5	99.69%	99.54%	94.07%	90.32%
1	99.64%	99.47%	94.02%	90.24%
3	99.47%	99.23%	93.79%	89.98%
6	99.28%	98.98%	93.61%	89.73%

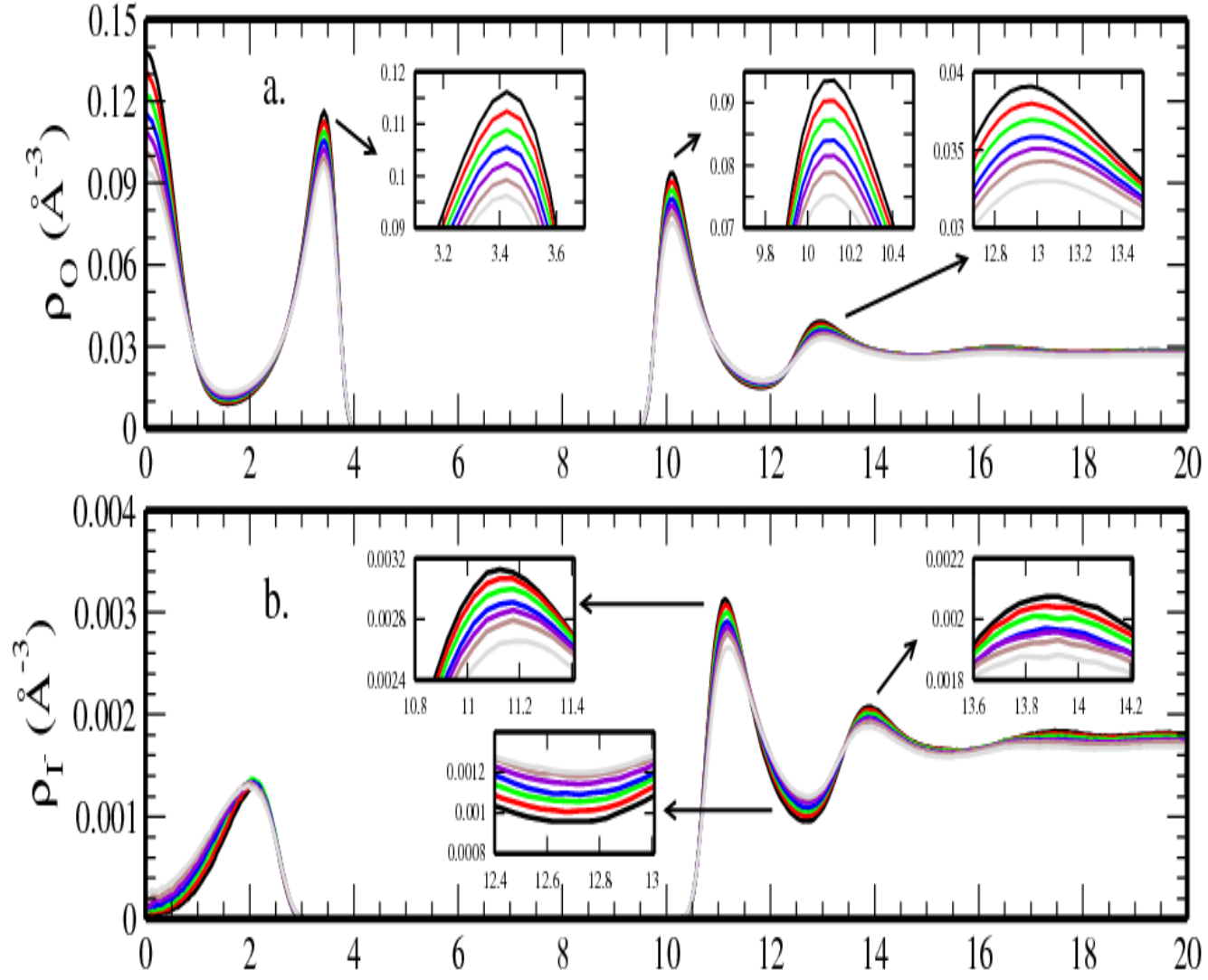
Case study 2: scientific results



Case study 2: scientific results



Case study 2: scientific results



Acknowledgments

Collaborators and contributors to this work:

Students in Taufer's group

Sandeep Patel, U. Delaware

Narayan Ganesan, Stevens Institute of Technology

Sponsors: