SAND-2013-0054P

Emerging High Performance Computing Systems and Next Generation Engineering Analysis Applications

James A. Ang, Richard F. Barrett, Simon D. Hammond, and Arun F. Rodrigues
Sandia National Laboratories
Albuquerque, New Mexico, USA

ABSTRACT

This paper provides a high level overview of the intersection between the broad fields of Infrastructure Engineering and Computer Systems Engineering. The last two decades of technical high performance computing (HPC) have been remarkably stable, with high-end scientific and engineering applications able to leverage the increases in performance of commodity processors in massively parallel supercomputers. But issues began to arise with the advent of the dual core processor in 2004. While many commercial workloads and some technical applications such as materials science can still achieve good performance on multi-core processors and many-core based systems, most finite element engineering analysis applications are sensitive to data locality and data movement and thus have difficulty realizing the performance potential of these systems. This paper describes the HPC co-design methodology we are using to guide the development of advanced concepts for HPC computer architectures and future engineering analysis applications that will execute on them.

Keywords: High performance computing (HPC), HPC co-design, engineering analysis proxy applications, HPC system concepts.

1. INTRODUCTION

Moore's Law is the underlying driving force in microelectronics that has improved performance in scientific and engineering applications for over four decades. In simple terms, it is based on Gordon Moore's observation that the transistor count for a given area of processor silicon will double approximately every two years (Moore, 1965). The exponential growth in transistor count has been utilized to introduce novel processor features such as hardware-based video-decoding or cryptography with enhanced performance through the inclusion of additional arithmetic units. Another important factor has been the effect of Dennard Scaling – the ability to employ higher clock frequency as silicon feature sizes are reduced (Dennard, et al, 1974). The practical effect of both Moore's Law and Dennard Scaling was that from 1970 to 2005 processor performance was doubled every 18 months.

In 2004, the microelectronics community saw Dennard Scaling stall; while Moore’s Law continued to provide a reduction in feature sizes and a doubling of transistor density every two years. Instead of working to increase serial code performance, the processor designers have used higher transistor counts to provide multi-core processors (Fuller and Millet, 2011), requiring the development of parallel algorithms to realize full processor performance. Such hardware changes however present a challenge for finite element engineering analysis applications which typically tend to stress data locality and data movement performance rather than raw calculation rate.

Data movement in the context of high performance computing (HPC) systems requires two considerations: 1) data movement from and to the local system memory which may include multiple levels of processor caches (localized stores of data to reduce access times), and 2) data movement across the system level interconnection network fabric. Since there are many types of commercial computing workloads that do not stress data movement performance, these applications are often able to extract good performance from multi-core processors and interconnection network fabrics that are relatively low performance with correspondingly low costs, e.g., Gigabit Ethernet is the primary interconnection network fabric for cloud computing systems. There are also classes of scientific applications, primarily variants of molecular dynamics materials science applications that can extract significant benefit from cloud computing systems and exploit commodity multi-core processors and many-core accelerators. These applications should be run on cloud computing resources, to free time and space for engineering analysis applications on HPC systems.

Unfortunately current HPC systems that should provide "high performance" on engineering analysis applications fall short of the theoretical peak performance of multi-core processors (Dongarra, et al, 2007), and the shortfall may be even worse on many-core accelerators. The gap between theoretical and realized performance for engineering applications appeared with the first dual-core CPU in 2004 and has been growing with each increase in the core-count for multi-core processors. Part of the solution is a new generation of finite element applications that are re-implemented to match the much higher levels of concurrency that are available in multi and many-core processors. We believe the solution will also require the integration of advanced computer architecture concepts such as user controlled caches, multi-level memory, optical interconnection network fabrics, and dynamic power management, with future generations of multi or many-core processors. The systematic investigation of how our portfolio of applications will need to change with new HPC system architectures is an important part of our co-design strategy.

2. BACKGROUND

Sandia National Laboratories (Sandia) is a multi-program U.S. Department of Energy (DOE) National Laboratory that has a deep foundation in applied research. In contrast to our sister DOE National Labs that are focused on various basic science disciplines, Sandia's primary focus is applied science for systems engineering. Sandia's technical capabilities in microelectronics, HPC computer architectures, system software, algorithms and engineering analysis applications are relevant for this paper.

Sandia has a unique perspective on HPC due to several capabilities that do not exist elsewhere in the DOE National Laboratory complex.

- Sandia has a semiconductor fabrication plant. This capability is supported by expertise in electrical and computer engineering, electronics design automation capabilities, electronics packaging and the ability to also access state of the art fabrication plants in the commercial sector.
- Sandia has the largest concentration of computer engineers and system architects that are focused on HPC at any place outside of industry.
- Sandia has a long history in the research and development of system software for large-scale HPC (Wheat, et al, 1994; Greenberg, et al 1997; Riesen, et al, 2009).
- Over 25 years ago, Sandia (Gustafson, Montry and Benner, 1988), helped establish explicit message passing on distributed memory, massively parallel processors (MPP) as the way to move high performance technical computing beyond the then ubiquitous vector-based supercomputers. This was a response to their seminal analysis of Amdahl's Law (Amdahl, 1967), proving that large scale parallelism can be effective when solving large scale problems.

2.1 *Computer Engineering versus Infrastructure Engineering*

Infrastructure engineering typically refers to the design and development of a nation's civil assets such as roads, railways, airports, water, sewer, power grid and other physical infrastructure. In computer science and engineering, computer architecture refers to a description of the structure and relationship among different hardware components and subcomponents of a computer system. As in the architecture of infrastructures, computer architecture can comprise many levels of information. The highest level of the definition conveys the conceptual implementation.

There are also some very important differences between computer and infrastructure engineering. The world of microelectronics is presently working with 22nm feature sizes, die areas of 400mm^2, and large HPC systems have a size on the order of 20m, or 400m^2 of area. In contrast, most infrastructures deal with components that have dimensions on the order of 0.01-1m, and overall sizes of 0.01-1,000km.

Another interesting contrast is found in the design life for microelectronics versus infrastructure engineering systems. The typical central processing unit (CPU), or "processor" has a design life of about 2 years, but for a system-on-chip cell phone processor the design life is now about 1 year. The design life for most infrastructure "products" is 50-100 years. There are well-known examples of physical infrastructures that have significantly longer lifetimes, e.g., the Great Wall of China and Stonehenge. Finally, we note that many of our successful engineering analysis applications have a design life of 20-30 years or more.

3. COMPUTER PLATFORM CLASSES

3.1 *Workstations*

A workstation is a high-end microcomputer designed for technical and scientific applications. Workstations offer higher performance than conventional desktop computers, especially with respect to the CPU and graphics processing unit (GPU), memory capacity, and multitasking capability. They are often tuned for visualization and manipulation of complex data such as 3D mechanical and electrical design, engineering analysis, animation and rendering of images, and mathematical plots. Twenty years ago, workstations were dominated by high performance RISC processors from IBM, HP, SGI, and SUN. Presently, the workstation market is highly commoditized and is dominated by large PC vendors, such as Dell and HP, selling X86 CPUs driven by Microsoft Windows or Linux operating systems. IBM continues to develop and offer its high end line of POWER-7 RISC CPU workstations.

3.2 *Cloud Computing*

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network. Often described as software as a service (SaaS), users are provided on-line access to application software and databases. There are many commercial workloads such as transaction processing, search, database, and data intensive computing that cloud computing is designed to support. As these commercial applications are hosted centrally, updates can be released without users having to reinstall new software. Users typically access cloud-based applications through a web browser while the commercial software and user's data are stored on servers at a remote location.

Most cloud services are set up to provide standardized commodity hardware, open source system software, application software, and databases. Cloud servers are typically clusters with X86 CPUs, and a Gigabit Ethernet interconnect fabric. Since most commercial workloads do not require data movement across the interconnection network fabric, this is a cost effective approach. Some higher end cloud servers are beginning to use a high performance Infiniband interconnection network fabric.

For Sandia's engineering analysis problems, small scale studies are performed on workstations. For larger scale problems, our engineers use Linux clusters – systems that are very similar to cloud servers with the exception that we use Infiniband interconnection networks instead of Gigabit Ethernet.

3.*3 High Performance Computing Systems*

Most current HPC systems have a high performance custom interconnection network fabric. These include massively parallel processor (MPP) systems that integrate large numbers of commodity processors. The challenges of energy-efficient computing are driving some system designs towards heterogeneous nodes that combine multi-core CPUs with many-core accelerators (GPUs). While this can provide energy efficient performance for some applications, it is unlikely such systems will meet the needs for data movement constrained applications. There is also much interest in specialized, low energy, system-on-chip, embedded processors which, in contrast to cell phone processors, are designed for tight integration at large scale.

It is no longer possible to buy a commodity single core processor CPU, and while the shift to multi-core processors has increased the theoretical peak performance of these processors in proportion to Moore's Law, the number of pins to support bandwidth to either the memory subsystem or the interconnection network fabric has experienced much slower growth. This is the data movement bottleneck that constrains performance for engineering analysis applications.

The HPC community is entering an exciting period of architectural diversity. It has similarities to the early 1980's when Cray vector supercomputers were falling behind the steady increases in commodity CPU performance that was driven by the combination of Moore's Law and Dennard scaling. The formula of MPPs with commodity CPUs took the decade of the 1980's to emerge as the winning solution. Many in the HPC community expect the current paradigm of MPP supercomputers will be similarly supplanted within the next decade.

To improve the performance of engineering analysis applications that are constrained by data locality and data movement performance, we need to examine innovative HPC system architectures that integrate new processor designs, new memory technologies such as stacked DRAM, hybrid DRAM and non-volatile memory, and new interconnection network technologies such as optics. In addition, we need a systematic examination of how our traditional finite element engineering analysis applications will map to these advanced computer architectures. This concurrent examination of the evolution of hardware and software is co-design.

4. CO-DESIGN METHODOLOGY

Sandia is leading the definition of the HPC co-design methodology for evaluation and design of advanced architecture concepts as well as guidance for the development of next generation of exascale applications and system software (Geist and Dosanjh, 2009; Alvin, et al, 2010; Ang, et al, 2011). Founded in the co-design principles defined by embedded systems community (Hu, et al, 1994), our approach to co-design is based on three key capabilities: 1) advanced architecture test-beds, 2) architectural simulators with proxy architectures, and 3) a portfolio of proxy applications that represent a good cross section of the computing workload. The relationship among these co-design capabilities is illustrated in Figure 1. The proxy architectures and applications are designed to provide high level models with sufficient fidelity to provide computer architects and application developers with insights into the design space. This figure also represents two capabilities that are beyond the scope of the HPC co-design methodology. These are the full HPC applications that the national labs develop, and the real HPC architectures that industry develops. These real HPC applications and architectures are both too complex to use for practical analysis in high level architectural simulators.

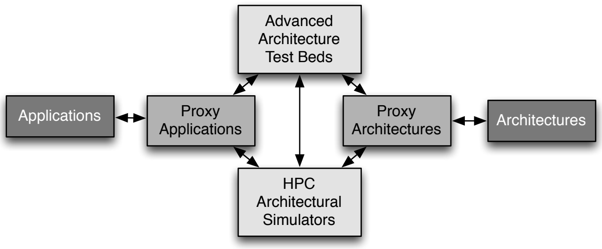


Figure 1 Relationship among co-design capabilities to define and develop advanced concepts for HPC architecture design, and understand how applications need to be re-implemented for HPC system architectures.

4.1 *Advanced Architecture Test-Beds*

Sandia has a diverse and growing set of experimental architecture test-beds to guide our HPC technology investment decisions. Our experience with experimental architecture test-beds allows us to become: 1) more informed collaborators with industry in co-design processes, 2) more adaptable to changes in hardware, and 3) able to establish a quantitative basis for making application programming model changes. Perhaps more importantly, our test-beds provide a foundation for decision makers to determine the path to exascale while continuing to meet our mission commitments.

4.2 *HPC Architectural Simulators*

Architectural simulator capabilities are important tools that enable co-design to close the loop back to computer architects and hardware component designers. Without this capability, co-design threatens to retreat to "business as usual," in which new HPC systems are procured and DOE application code teams, algorithm developers, and system software developers are then given the task of extracting the best performance they can from the HPC system we have. The intent behind HPC architectural simulation is to obtain quantitative data to guide the technology development and design of all elements of the integrated HPC system. Our goal is to develop a small set of proxy architectures that have enough fidelity to expose fundamental changes that occur with enhanced data movement capabilities, but are not so detailed as to intrude into proprietary designs, and require cycle-accurate architectural simulation capabilities.

DOE industry partners have a tradition of using simulators to analyze and model processors, interconnection networks, and other features of their proprietary designs. Some simulation capabilities are cycle-accurate but highly proprietary (Bohrer, et al, 2004). To the extent that the DOE HPC program can access and use these simulators, or provide proxy applications to drive these proprietary simulators, important quantitative data can be obtained to inform the co-design process. Processor models can be integrated with memory subsystem and network interface models to provide a node-level model. To support simulations and analysis of large scale systems that integrate thousands to millions of cores, it is also useful to reduce node-level model fidelity to allow simulation of HPC systems consisting of up to hundreds of thousands of nodes.

To encourage interoperability between different simulation models and allow simulation at large scale, Sandia and several other organizations are developing the Structural Simulation Toolkit (SST) (Rodrigues, et al. 2012). The SST is an architectural simulation framework designed to be modular and parallel. By bringing models of different parts of a computer system together, e.g. detailed processor models and network models, it is possible to observe subtle feedback loops that can develop between different subsystems in a way that stand-alone component simulations cannot. This is akin to simulating an entire city at once, instead of only simulating a city's traffic or power grid in isolation.

The SST adopts a simple modular architecture as shown in Figure 2. The core of the SST provides a parallel discrete event simulation interface that allows different component models to interact with each other. It also provides common support services like configuration management, check pointing, and power analysis. This modular architecture presents a common interface, allowing Sandia or DOE's industry partners to create their own proprietary modules without having to expose

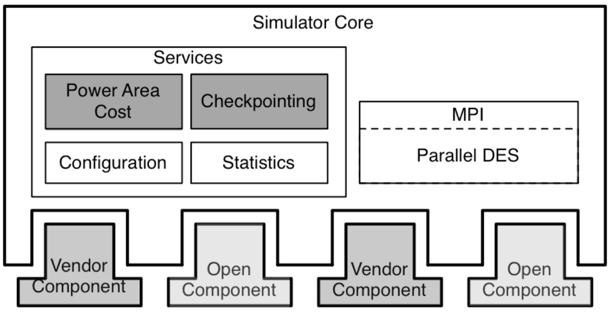


Figure 2 The Modular architecture of the Structural Simulation Toolkit allows easy interaction between a wide array of open and proprietary component simulators.

their inner workings. Using the SST as a common platform for simulation allows results to be exchanged more easily and encourages the types of feedback that are critical to effective co-design.

The SST allows the construction of proxy architectures that capture key aspects of future technologies. Combining these proxy architectures with proxy applications, it is possible to use simulation to estimate the performance impact on future applications of emerging technologies. For example, 3D stacking of DRAM memory has the potential to dramatically improve the available bandwidth and effective memory latency of future machines (Pawlowski, 2011). Stacked memory designs are still evolving, and physical test-beds will not be available for some time. Additionally, many of the internals of these devices are proprietary. However, by using a generic proxy architecture for a stacked memory part, it is possible to perform experiments that show the potential performance of stacked memories on proxy applications as shown in Figure 3. These experiments show that stacked memory parts have the potential to greatly improve performance, but the performance gains are very dependent on the internal parallelism of the memory stack and the application. This provides early indications of which applications may benefit the most from a new technology, or may indicate if a proposed architecture shows promise.

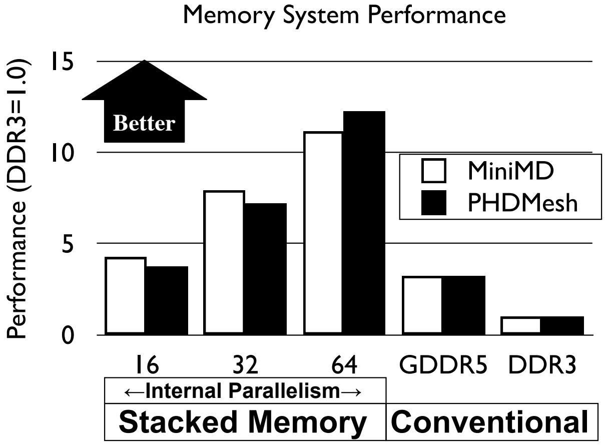


Figure 3 SST Memory Simulation Results

4.3 *Proxy Applications*

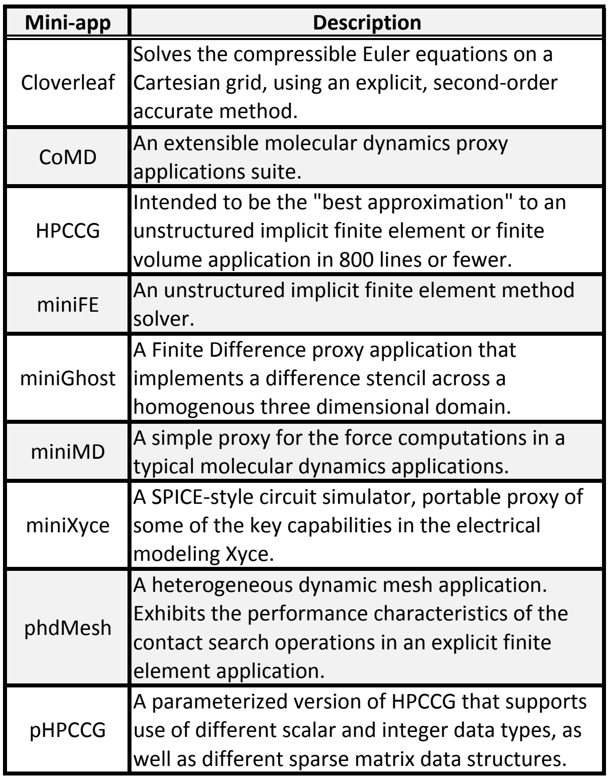
Mission critical engineering analysis application programs used by DOE consist of millions of lines of code, are written using multiple programming languages, link several supporting libraries, and capture significant bodies of knowledge developed over multiple generations of scientists. Some of our engineering analysis applications have roots that trace back to the vector supercomputers of the 1980's, so it is accurate to say that the design life of these applications is measured in decades. The Mantevo project (see http://mantevo.org), initiated at Sandia, was motivated by the need to tractably explore performance-impacting issues of current, emerging, and future architectures. Toward that end, we have developed a set of mini-apps, small self-contained codes that enable agile exploration of a variety of issues that impact performance throughout the co-design space, ranging from low-level hardware capabilities to the application.

Mini-apps are designed to be one of many tools needed to prepare engineering analysis applications for new architectures (Heroux, et al, 2009). Unlike a benchmark, the result of which is a value to be ranked, the output of a mini-app is information, which must be interpreted within some often subjective context. Unlike a compact application, which is designed to capture some sort of physics behavior, mini-apps are designed to capture a key performance issue in the full application. Unlike a skeleton application, which is designed for only focusing on inter-process communication perhaps involving a "fake" computation, mini-apps create an application-relevant context in which to explore the key performance issue.

Developed and owned by application code teams, mini-apps are intended to be modified, and thus are limited to a few thousand source lines of code (SLOC). Once no longer useful for these purposes, a mini-app's life will end. Mantevo mini-apps are freely available as open source software under an LGPL license.

The current set of mini-apps in the Mantevo project is listed in Table 1. Several have been successfully used as part of the co-design of new computer systems and applications during this time of rapid transition to scalable multi-core and accelerator based computer systems. Further, they will play a role in the procurement of our future advanced technology HPC systems, beginning with the joint Los Alamos and Sandia National Laboratories 2015 *Trinity* HPC system procurement for the NNSA/ASC program.

Table 1: Mantevo Project Mini-apps



Mini-app	Description
Cloverleaf	Solves the compressible Euler equations on a Cartesian grid, using an explicit, second-order accurate method.
CoMD	An extensible molecular dynamics proxy applications suite.
HPCCG	Intended to be the "best approximation" to an unstructured implicit finite element or finite volume application in 800 lines or fewer.
miniFE	An unstructured implicit finite element method solver.
miniGhost	A Finite Difference proxy application that implements a difference stencil across a homogenous three dimensional domain.
miniMD	A simple proxy for the force computations in a typical molecular dynamics applications.
miniXyce	A SPICE-style circuit simulator, portable proxy of some of the key capabilities in the electrical modeling Xyce.
phdMesh	A heterogeneous dynamic mesh application. Exhibits the performance characteristics of the contact search operations in an explicit finite element application.
pHPCCG	A parameterized version of HPCCG that supports use of different scalar and integer data types, as well as different sparse matrix data structures.

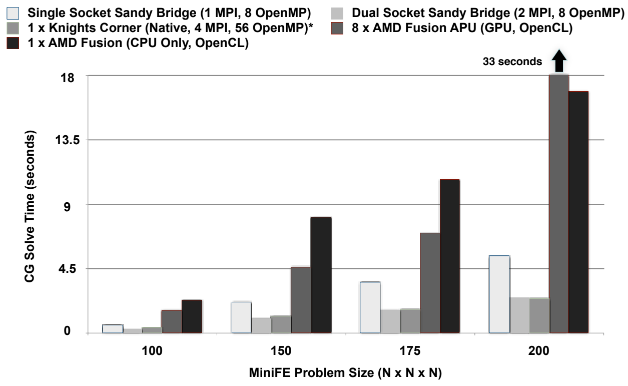
In combination with proxy applications and proxy architectures, simulation enables architectural exploration as well as strategies for tuning future applications through the vehicle of proxy applications. As new architectural concepts like multi-level memory, user controlled caches, non-volatile memory, and new types of accelerators are introduced, it will be increasingly important to understand the impact on applications and to redesign future applications to take advantage of new capabilities.

As implied by the co-design iterations in the center of Figure 1, mini-apps can be used to collect performance data on advanced architecture test-beds for comparison to SST architectural simulations of these same mini-apps used to drive proxy architecture models of these test-beds. This activity helps us understand the fidelity and limitations of our SST architectural simulation capabilities.

5. THE MINI-FE PROXY APPLICATION

Many engineering applications in production use at Sandia require the implicit solution of nonlinear systems of equations. The specific combination of preconditioners and solvers used for these problems varies by application, problem complexity, and user choices, but all rely on a common set of fundamental mathematical operations. The miniFE (mini-Finite-Element) mini-app is an expression of these basic mathematical operations arranged to perform a linear finite-element data assembly phase followed by solution using the conjugate gradient (CG) method. MiniFE is not designed to solve a complex physics problem but instead to be sufficiently representative of performance concerns which we observe have greatest impact on the execution time of real problems – attention is paid in particular to the sparse matrix-vector product kernel that can consume vast amounts of runtime in practical problems.

Due to the importance of the mathematical operations employed within miniFE, we have focused on developing a broad range of architecture-centric implementations to explore trade offs in programming complexity and achieved solver performance. Figure 4 presents CG solver performance for a fixed number of iterations when running on existing compute nodes (Intel Xeon E5-2670 "Sandy Bridge" oct-core processors) and potential advanced future hardware offerings from Intel Knights Corner, a preproduction version of the Intel Xeon Phi coprocessors, and AMD Trinity A10 Fusion heterogeneous processor, which integrates 4 X86 CPU and 384 Radeon GPU cores into a single socket. These results show approximate parity in runtime between a single



* Performance results are from pre-production 57-core, 1.1GHz Intel Xeon Phi coprocessors (codenamed Knights Corner)

Figure 4 CG Solver Runtime (Seconds) for miniFE running 100^3, 150^3, 175^3 and 200^3 Problem Sizes

Knights Corner coprocessor card and dual-socket Sandy Bridge processors, with the Knights Corner demonstrating the optimization of the design for increased problem scale. AMD’s Trinity Fusion exhibits poor performance with the current design of the algorithm indicating the need for a change in the format of data structures used to represent matrices. Initial investigations point to the sparse structure of the matrices being solved leading to inefficient indirect memory patterns. We are currently investigating matrix format and solver algorithm alternatives that may improve performance on future GPU devices from both AMD and NVIDIA.

CONCLUSIONS

The natural evolution of commodity CPUs and accelerators is predicted to meet the computing requirements for commercial and some scientific applications that will perform well on cloud computing servers. This is not surprising as by definition, the commodity CPU and accelerator designers are targeting mainstream commercial applications for cloud servers.

The performance gap for finite element engineering analysis applications will drive needed changes in both next generation finite element analysis applications and future computer architectures. While there is not enough time to make significant architecture-centric modifications to our engineering analysis applications for the next HPC system procurement, we can achieve significant change in the Exascale timeframe. The coming decade will see development of several generations of processors; if we are able to articulate our priorities for processor designers and system integrators, we have an opportunity to realize significant improvements in engineering analysis capability.

While Moore’s Law still holds, processor designers recognize that using additional transistors to simply increase core count has reached diminishing returns. Our challenge is to collaboratively quantify the benefit of new hardware capabilities for both our engineering analysis applications and commercial applications. This will be the motivation for processor designers to integrate new hardware capabilities into future commodity processors.

The most effective way to design both hardware and software is through co-design. Hardware changes will likely impact future processor designs, memory subsystems and interconnection networks. Software changes will be implemented at the application, algorithm, and system software levels. At Sandia, our co-design strategy combines test-beds with proxy applications and proxy architecture models running on architectural simulators to inform both applications development code teams and computer / system architects. Since the design life of our engineering analysis application “products” is much closer to the design life of Infrastructure products than the design life for the processors and HPC systems that these applications run on, the effort and time spent on co-design is warranted. As various advanced architecture concepts are considered, our co-design tools are also applied to draw a connection to application performance, programmability and portability.

ACKNOWLEDGEMENTS

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy’s National Nuclear

Security Administration under contract DE-AC04-94AL85000. This paper summarizes work that is supported by the U.S. Department of Energy's Advanced Simulation and Computing and Advanced Scientific Computing Research Programs.

REFERENCES

Alvin K., Barrett B., Brightwell R., Dosanjh, S., Geist A., Hemmert S., Heroux M., Koethe D., Murphy R., Nichols J., Oldfield R., Rodrigues A., and Vetter J., 2010. International Journal of Distributed Systems and Technologies, 1(2), 1-22.

Amdahl G.M., "Validity of the single processor approach to achieving large scale computing capabilities," 1967. American Federation of Information Processing Societies (AFIPS) Spring Joint Computer Conference Proceedings, vol. 30, 483-485.

Ang J.A., Brightwell R., Donofrio D., Dosanjh S., Hemmert K.S., Rodrigues A., Shalf J., and Wheeler K., 2011. "Exascale Computing and the Role of Co-Design," in *High Performance Computing: From Grids to Clouds to Exascale*, Foster, I. ed., IOS Press.

Bohrer P., Peterson J., Elnozahy M., Rajamony R., Gheith A., Rockhold R., Lefurgy C., Shafi H., Nakra T., Simpson R., Speight E., Sudeep K., Van Hensbergen E., and Zhang L., 2004. "Mambo: A Full System Simulator for the PowerPC Architecture," SIGMETRICS Performance Evaluation Review, 31(4), 8-12.

Dennard R.H., Gaensslen F.H., Yu H.-N. Rideout V.L., Bassous E., and Leblanc A.R., 1974. "Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions," IEEE Journal of Solid-State Circuits, SC-9(5), 256-268.

Dongarra J., Gannon D., Fox G., and Kennedy K., 2007. "The Impact of Multicore on Computational Science Software," Cyberinfrastructure Technology Watch Quarterly, 3(1) 1-8.

Fuller S., and Millet L., eds., 2011. *The Future of Computing Performance: Game Over or Next Level*, National Academy Press.

Geist A., and Dosanjh S., 2009. "IESP Exascale Challenge: Co-Design of Architectures and Algorithms," International Journal of High Performance Computer Applications, 23(4), 401-402.

Greenberg D., Brightwell R., Fisk L.A., Maccabe A.B., Riesen R., 1997. "A System Software Architecture for High End Computing," Proceedings of the 1997 ACM/IEEE Supercomputing Conference, Nov 15-21, San Jose, California.

Gustafson J.L., Montry G.R., and Benner R.E., 1988. "Development of Parallel Methods for a 1,024-Processor Hypercube," SIAM Journal on Scientific and Statistical Computing, vol. 9, 609-638.

Heroux, M.A., Doerfler, D.W., Crozier P.S., Willenbring J.M., Edwards H.C., Williams A., Rajan M., Keiter E.R., Thornquist H.K., and Numrich R.W., 2009. *Improving Performance via Mini-applications*, Sandia Report, SAND2009-5574.

Hu X., D'Ambrosio J.G., Murray B.T., and Tang D.-L., 1994. "Codesign of architectures for automotive powertrain modules," IEEE Micro, vol. 14, 17–25.

Moore G.E., 1965. "Cramming More Components onto Integrated Circuits," Electronics, April 19, 1965, 114-117.

Pawlowski J.T., "Hybrid Memory Cube (HMC)", 2011. Proceedings of Hot Chips-23: Symposium on High Performance Chips, August 17-19, Palo Alto, CA.

Riesen R., Brightwell R., Bridges P.G., Hudson T., Maccabe A.B., Widener P.M., Ferreira K., 2009. "Designing and Implementing Lightweight Kernels for Capability Computing," Concurrency and Computation: Practice and Experience, 21(6), 793-817.

Rodrigues A.F., Cooper-Balis E., Bergman K., Ferreira K., Bunde D., and Hemmert K.S., "Improvements to the Structural Simulation Toolkit," 2012. Proceedings of the 5th International Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering (ICST) Conference on Simulation Tools and Techniques (SIMUTOOLS '12), March 19-23, Brussels, Belgium, 190-195.

Wheat S.R., Maccabe A.B., Riesen R., van Dresser D.W., Stallcup M.T., 1994. "PUMA: An operating system for massively parallel systems," Scientific Programming, vol. 3, 275–288.