A Framework for Adaptable Operating and Runtime Systems

DOE Award/Contract No: DE-FG02-06ER25730

CFDA No. 81.049

Funding Document No. 13SC003119

PI: Thomas Sterling, Indiana University

Period of Performance: 08/15/2006 through 08/14/2009

Executive Summary

The emergence of new classes of HPC systems where performance improvement is enabled by Moore's Law for technology is manifest through multi-core-based architectures including specialized GPU structures. Operating systems were originally designed for control of uniprocessor systems. By the 1980s multiprogramming, virtual memory, and network interconnection were integral services incorporated as part of most modern computers. HPC operating systems were primarily derivatives of the Unix model with Linux dominating the Top-500 list. The use of Linux for commodity clusters was first pioneered by the NASA Beowulf Project. However, the rapid increase in number of cores to achieve performance gain through technology advances has exposed the limitations of POSIX general-purpose operating systems in scaling and efficiency. This project was undertaken through the leadership of Sandia National Laboratories and in partnership of the University of New Mexico to investigate the alternative of composable lightweight kernels on scalable HPC architectures to achieve superior performance for a wide range of applications. The use of composable operating systems is intended to provide a minimalist set of services specifically required by a given application to preclude overheads and operational uncertainties ("OS noise") that have been demonstrated to degrade efficiency and operational consistency. This project was undertaken as an exploration to investigate possible strategies and methods for composable lightweight kernel operating systems towards support for extreme scale systems.

Comparison of Accomplishments with Goals

The original objectives of this work included exploring the realm of new classes of architectures and the potential application of composable lightweight kernels to them, and providing a test runtime environment very different from conventional MPI implementations to stress the abilities of the composable methodology for efficient execution. The strategy for this project has been to devise, specify, and characterize the underlying model of computation that will address the combined challenges of billion-way parallelism and intrinsic system-wide latency hiding to support next generation Petaflops systems that will be available to DOE national laboratories at the end of this decade as well as enabling future Exascale systems in the latter half of the next decade. The interface between the running user application and the computer's operating system can be defined through the needs of the runtime system (provided by the compiler) and this requires a prototype of such a runtime system for a future generation of advanced systems to use as a driver of any future OS. This also has direct implications for possible programming models.

A number of important accomplishments were achieved during the project. In summary these include:

1. Implementation of first distributed version of the ParlleX execution model, "*disPX*", in C++ using existing libraries wherever possible for rapid prototyping and robustness.
2. Demonstration of simple kernel benchmarks running on distPX across multiple nodes of a parallel system.
3. Development of an advanced OS architecture strategy for scalable lightweight kernel in support of distributed global address space execution.
4. Expansion of programming model semantic constructs to the domain of dynamic directed graph data structures as part of a notional "*Agincourt*" Exascale programming language.
5. Implementation of hardware testbed for experiments with heterogeneous computing systems.
6. Educational outreach through introductory course in HPC using internet distribution.
7. Establishment of complementing projects through additional sponsorship.
8. Participation in NNSA External Review of "Roadrunner" and LANL.

Project Activities

1. **Advanced Semantics** – The project is focused on the development, implementation, and integration with Sandia/UNM Config-OS of the ParalleX execution model. This is a message-driven multi-threaded global name space model incorporating lightweight futures synchronization and dynamic adaptive resource management. This model integrates a number of earlier ideas while adding some innovative concepts to provide a single unified API to complex parallel computing systems. ParalleX has evolved from the interplay of two classes of work in architecture focusing on processor in memory, and in programming languages exploring how to hide latency. The LSU project has significantly advanced and refined the ParalleX model. Although subtle, three significant semantic advances have been achieved. The first is a unification between suspended threads and local control objects. Previously it was assumed that these two classes of constructs were distinct with the suspended thread being a somewhat odd creature not cleanly integrated in the semantic space of the other model elements. It was discovered that a suspended thread could be naturally converted into a local control object that we call a "depleted thread" that has all the semantic properties of a local control object and is, in fact, one. This has important ramifications for both simplifications of the model and its implementation. A second advance is in the area of continuations related to parcels which is a kind of active message. A unique attribute of ParalleX is its ability to migrate the control flow of execution seamlessly from one hardware node to another. This was to be achieved using "parcels" by had not been specified. This has now been resolved taking advantage of the global name space of the ParalleX system. The third advance is the development of the intra-thread operation flow control. Unlike most thread models, ParalleX has extended its model to provide a natural form of fine grain parallelism that permits compilation to meet a wide range of target architectures. A new form of the old static dataflow has been applied for this purpose including the abstraction of single assignment registers to avoid anti-dependencies.
2. **Future Target Architectures** – The initial advanced class of target architectures identified in the original proposal was the important family of processor in memory (PIM) architectures that may address the memory wall providing substantial improvements in memory bandwidth and latency through the integration of logic on to the memory chip. This space has been significantly

broadened as a result of the project at LSU. One advance is in the area of instruction encoding which is an important overhead in storage space, data and transfer time. A new technique called “precise” (Processor Register Extensions for Collapsed Instruction Set Encoding) has been devised that uses a combination of register tagging, Huffman encoding, and preferential register access patterns to reduce the average instruction word length from a fixed-length of 32 bits to a variable length of 4 to 5 bits (estimated). This can reduce the power consumption for this aspect of computing, reduce the size of small cores for PIM and embedded processors, and permit entire instruction streams to be transferred in messages (e.g., parcels) to remote sites. Another advance is in the use of a class of streaming architectures (e.g., Merrimac, Trips) for regimes of high temporal locality where clock rate and ALU density is important. The Gilgamesh-2 architecture combines this form of accelerator with the previous MIND PIM memory oriented structures to provide a heterogeneous multicore system capable of delivering a sustained Exaflops towards the end of the next decade. Finally, the project explored the use of augmented commodity computing applying the ParalleX model to conventional systems with accelerators and improving the performance of the global overhead mechanisms managing system wide resources through the application of FPGA technology.

3. **Distributed Implementation** – The project completed development of the ParalleX system. As previously reported, a reference implementation was developed (two generations) that ran on sequential systems for the purposes of verifying the semantic validity of the models for application programming. The project began an implementation of the core of the ParalleX system on commodity clusters and MPPs using a set of C++ libraries that have already been developed and proven. This new software system will act as an intermediate step towards a fully functional, high performance parallel ParalleX implementation ported to the Sandia Config-OS lightweight kernel operating system. While not an exhaustive implementation of the full ParalleX semantic model, it captures the key concepts and implements the critical overhead mechanisms that will determine the ultimate effectiveness of the strategy. The distPX system is implemented across system nodes making it a true distributed version of the ParalleX model. It is based in large part on existing C++ libraries like Boost where appropriate for rapid prototyping and robust operation.
4. **Future Architecture Elements**- Specific future architecture elements that were examined by LSU included the following:
    a. Multi-core processor sockets – these have become the principal component from which scalable HPC systems are comprised. While providing many independent threads of action simultaneously on a single die, multi-core chips impose multiple points of resource contention that aggravate rather than ameliorate the memory wall. The shared on-chip communication channels, caches, and pins create such points of contention.
    b. Processor in Memory (PIM) – chips which combine processors and memory blocks in the form of embedded memory processors can greatly increase effective memory bandwidth and reduce latency effects while providing efficient atomic compound sequences. By merging DRAM with logic the average power per unit chip area can be reduced and the shorter latency can reduce energy of data communications.
    c. Stacked Dies – the layering of memory, logic, and communication dies in stacks and providing interconnect among them by “via’s” for higher bandwidth, reduced latency, low power, and compact packaging.
    d. Lightweight Cores – small processors for reduced power consumption and higher density packaging.
    e. Message-driven communication – use of active messages to move work to data, both reduce and hide latency, and improve efficiency.
    f. Global Address Space – to support direct access to memory anywhere in the system but without cache coherence.

   g. GPU Accelerators – heterogeneous architectures that include subsystems optimized for particular computational modalities.

5. **Benchmark Demonstration**- a set of simple benchmark kernels were crafted in the ParalleX Intermediate Form (PXIF) and run using distPX. These demonstrated the correct operation of distPX and provided experience with low level representation of distributed computation in the new ParalleX model. This is not a performance implementation and is not expected to provide performance advantage given that the underlying mechanisms were not optimized for performance but rather for correctness. Nonetheless, this is an important result showing the validity of many aspects of the ParalleX concept.
6. **OS Strategy**- with Sandia national Laboratory and the University of New Mexico, LSu developed a new and powerful strategy for operating system development that will lead to capabilities necessary for future Exascale systems. Leveraging based experiences with such lightweight kernel work as Catamount, a successful and commercially distributed operating system of high efficiency and scalability on conventional MPP systems, a new class of operating system has been conceived that will provide a single virtual image of a highly parallel and widely distributed physical system architecture while delivering the performance efficiency and design simplicity of prior lightweight kernels. This innovation employs a new family of inter-kernel protocols for synergistic operation on a functional basis. For example, XOS would provide a global system layer of distributed shared memory similar to but more advanced than current experimental PGAS models. Similarly layers for a sea of threads and for I/O are also provided. This strategy was the basis for an unsuccessful proposal to the second phase of FAST-OS and is being used for further funding projects.
7. **Programming Model**- A long term activity related to user programming model for Exascale systems is focused on the notional Agincourt programming language, first conceived under the DOE Advanced Programming Models project led by ANL. During this project, work focused on defining the set of programming constructs necessary not only to support the ParalleX runtime system but additionally to represent computations on dynamic directed graphs.
8. **Testbed**- A testbed was deployed at LSU during the project in support of the research on ParalleX and future OS for distributed heterogeneous multicore environments. *Uther* is a 4 compute node, 1 master node system physically packaged to permit easy reconfiguration of system components including multiple system area networks, accelerator cards, and FPGA modules using both PCI and HyperTransport interfaces. The testbed supported the research of a number of students and scientists.
9. **Community Initiatives**- the PI of the project played an important role in community inititiatives to explore research directions towards next generation and Exascale systems and computation. Invited presentations were given at the DOE Town Hall meetings on Exascale computing. The PI was a member of the DARPA blue ribbon panel on the Exascale Study and made substantive contributions on this report. The PI also was a member of an NNSA led external review committee of the Petascale Roadrunner system to be deployed at LANL.

Publications

Thomas Sterling, Chirag Dekate, "Productivity in High Performance Computing", Advances in Computers 2008; Vol.2

Michael, Chris J.; Brodowicz, Maciej; Sterling, Thomas. "Improving Code Compression Using Clustered Modalities." In proceedings of the 46th annual ACM southeast regional conference. Auburn, AL. 2008. Web link: http://eng.aubum.edu/acmse/index.html

Related Follow- On Projects

This research has directly contributed to succeeding research projects. These include:

UHPC X-Caliber Project, DARPA (subcontract to Sandia National Laboratories), PO# 1190769, $156,000, 11/17/2011 – 9/30/2012

eXascale PRogramming and System Software (XPRESS), DOE, DE-SC0008809, $1,100,000, 9/1/2012 – 8/31/2015

Advanced Development of the DOD Extreme Scale Execution Framework, DOD (subcontract to HPTi), $4,136,917, 6/1/2012 – 5/31/2017

Strong Scaling for DoD HPC Applications, DOD (subcontract to HPTi), $500,000, 4/1/2012 – 2/28/2014

SHF: Large: Collaborative Research: PXGL: Cyberinfrastructure for Scalable Graph Execution, NSF, $700,000, 8/1/2011 – 7/31/2014