Argo: An Exascale Operating System and Runtime

Swann Perarnau
swann@anl.gov

Rinku Gupta
rgupta@anl.gov

Pete Beckman
beckman@anl.gov

Keywords

High-Performance Computing, Supercomputers, Exascale, Operating System, Runtime

1. INTRODUCTION

Exascale supercomputers are expected to comprise hundreds of thousands of heterogeneous compute nodes linked by complex networks. Those compute nodes will have an intricate mix of general-purpose multi-cores and special-purpose accelerators targeting compute-intensive workloads with deep multi-level memory hierarchies. As such, the HPC community expects exascale systems to require new programming models, to take advantage of both intra-node and inter-node parallelism.

The Argo project, funded under the DOE ExaOSR initiative, aims to provide an Operating System and Runtime (OS/R) designed to support extreme-scale scientific computations. With this goal in mind, Argo seeks to efficiently exploit new processor, memory and interconnect technologies while addressing the new modalities, programming environments, and workflows expected at exascale. At the heart of this project are four key innovations: dynamic reconfiguring of node resources in response to workload changes, allowance for massive concurrency, a hierarchical framework for management of nodes, and a cross-layer communication infrastructure that allows resource managers and optimizers to communicate efficiently across the platform. These innovations will result in an open-source prototype system that is expected to form the basis of production exascale systems deployed in the 2020 timeframe.

We provide here a overall description of the project, before highlighting recent achievements in performance and integration with existing systems.

2. THE ARGO PROJECT

Providing a complete software stack for exascale systems, the Argo components span all levels of the machine: a parallel runtime seats on top of a HPC-aware operating system for each node, while a distributed collection of services manages all nodes, using a global communication bus.

NodeOS is the operating system running on each node of an Argo machine. It is a based on the Linux kernel, tuned and extended for HPC use on future architectures. In particular, we leverage the control groups interface and extend it to provide lightweight *compute containers* with exclusive access to hardware resources. To limit OS noise on the node, system services are restricted to a small dedicated share of cores and memory nodes. Additionally, the NodeOS provides custom memory and scheduling policies, as well as specialized interfaces for parallel runtimes.

Argobots is the runtime component of Argo. It implements of low-level threading and tasking framework entirely in user-space, giving users total control over their resource utilization, and provides data movement infrastructure and tasking libraries for massively concurrent systems.

GlobalOS is a collection of services implementing a distributed, dynamic control of the Argo machine. It divides the system into *enclaves*, groups of nodes sharing the same configuration and managed as a whole. Those enclaves can be subdivided, forming a hierarchy, with dedicated nodes (masters) at each level to respond to events. Among the provided services, the GlobalOS includes distributed algorithms for power management and fault management mimicking exceptions across the enclave tree.

The Global Information Bus(GIB) is a scalable communication infrastructure taking advantage of modern high performance networks to provide applications and system services efficient reporting and resource monitoring services.

In its current state, the Argo project has prototype implementations of most of its components:

1. A prototype design and implementation of Global OS built on top of OpenStack services. This current implementation relies on bare metal provisioning of compute nodes, and provides enclave creation and tracking, configuration of system services and job launching.

2. The GlobalOS also includes distributed enclave and system-wide power management algorithms.

3. BEACON, the pub/sub framework of the Global Information bus is available in its 1.0 version. A prototype

SC'15 November 15–20, 2015, Austin, TX, USA

implementation on top of EVPATH and the RIAK key value store is also available.

4. The Argobots runtime has been successfully integrated with several existing programming models: MPI, Open MP, Charm++, Cilk Plus, PTGE.

5. In addition, collaboration with RIKEN in Japan led to a highly scalable OpenMP implementation for nested and irregular loops/tasks on top of Argobots.

6. The NodeOS currently provides partitioning of CPU and memory ressources, a prototype implementation of its compute containers as well as a custom scheduling policy for modern HPC runtimes.

7. DI-MMAP, a tool to integrate NVRAM into the memory hierarchy of the system and use it for parallel application is also integrated.

For the near future, the Argo project will focus on greater integration between its components, aiming for scalability and functionality testing on large scale DOE facilities and applications. In particular, we will focus on the following points:

1. Refining the functionality of Global OS to include: failure management, fault tolerance, recursive enclave management and user customization of enclaves.

2. Add functionality to EXPOSE, the performance monitoring component, and integrate it with TAU.

3. Research new features in NodeOS, Argobots runtime and the GIB that may arise as more information on future systems is made available.

4. Prepare and demonstrate at future conferences a complete integrated software stack on a large scale system.

3. ADDITIONAL AUTHORS

Argonne National Laboratory: Judicael Zounmevo, Huiwei Lu, Kenneth Raffenetti, Sangmin Seo, Pavan Balaji, Franck Cappello, Kamil Iskra, Rajeev Thakur, Kazutomo Yoshii, Marc Snir.

University of Illinois at Urbana-Champaign: Cyril Bordage, Laxmikant Kale, Yanhua Sun, Jonathan Lifflander.

University of Tennessee: George Bosilca, Jack Dongarra, Damien Genet, Thomas Herault.

University of Oregon: Sameer Shende, Xuechen Zheng, Wyatt Spear, Daniel Ellsworth, Allen D. Malony.

Lawrence Livermore National Laboratory: Maya Gokhale, Barry Rountree, Martin Schulz, Brian Van Essen, Edgar Leon.

University of Chicago: Henry Hoffman, Nikita Mishra, Huazhe Zhang

Pacific Northwest National Laboratory: Sriram Krishnamoorthy, Roberto Gioiosa, David Callahan, Gokcen Kestor.

4. REFERENCES

[1] A. Danalis, G. Bosilca, A. Bouteiller, T. Herault, and J. Dongarra. PTG: An abstraction for unhindered parallelism. In *International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing (WOLFHPC)*, New Orleans, LA, 11/2014 2014. IEEE Press, IEEE Press.

[2] D. Ellsworth, A. Malony, M. Schulz, and B. Rountree. POW: System-wide Dynamic Reallocation of Limited Power in HPC. In *24th International ACM Symposium on High-Performance Distributed Computing (HPDC 2015)*, 2015.

[3] H. Hoffmann and M. Maggio. PCP: A generalized approach to optimizing performance under power constraints through resource management. In *ICAC*, 2014.

[4] C. Imes, D. H. K. Kim, M. Maggio, and H. Hoffmann. Poet: A portable approach to minimizing energy under soft real-time constraints. In *RTAS*, 2015.

[5] N. Mishra, H. Zhang, J. D. Lafferty, and H. Hoffmann. A probabilistic graphical model-based approach to minimizing energy under performance constraints. In *ASPLOS*, 2015.

[6] T. Patki, D. Lowenthal, A. Sasidharan, M. Maiterth, B. Rountree, M. Schulz, and B. de Supinski. Practical resource management in power-constrained, high performance computing. In *24th International ACM Symposium on High-Performance Distributed Computing (HPDC 2015)*, 2015.

[7] S. Perarnau, R. Thakur, K. Iskra, K. Raffenetti, F. Cappello, R. Gupta, P. Beckman, M. Snir, H. Hoffmann, M. Schulz, and B. Rountree. Distributed Monitoring and Management of Exascale Systems in the Argo Project. In *15th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS 2015)*, June 2015.

[8] B. Van Essen, M. Jiang, and M. Gokhale. Developing a framework for analyzing data movement within a memory management runtime for data-intensive applications. In *Non-Volatile Memories Workshop*, San Diego, CA, Mar. 2015.

[9] W. Wu, A. Bouteiller, G. Bosilca, M. Faverge, and J. Dongarra. Hierarchical dag scheduling for hybrid distributed systems. In *29th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, Hyderabad, India, 05/2015 2015. IEEE, IEEE.

[10] H. Zhang and H. Hoffmann. A quantitative evalution of the RAPL power control system. In *Feedback Computing*, 2015.

[11] J. A. Zounmevo, K. Iskra, K. Yoshii, R. Gioiosa, B. C. V. Essen, M. B. Gokhale, and E. A. Leon. A single-kernel approach to OS specialization and node resource partitioning for exascale computing. 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI '14), Oct. 2014. (Poster).

[12] J. A. Zounmevo, S. Perarnau, K. Iskra, K. Yoshii, R. Gioiosa, B. C. V. Essen, M. B. Gokhale, and E. A. Leon. A container-based approach to OS specialization for exascale computing. In *1st International Workshop on Container Technologies and Container Clouds (WoC '15), held in conjunction with IEEE International Conference on Cloud Engineering (IC2E '15)*, Tempe, AZ, Mar. 2015.