ARGO: An Exascale Operating System and Runtime

**Objective**

- Heterogeneous, massively parallel compute nodes
- High Performance complex network topologies
- Constrained Resources (Power, I/O)
- Resilience, fault and system management
- Need the community to redesign and rethink Operating System and Runtime (OS/R) architectures for extreme-scale systems.

**Exascale Challenges**

- A system-wide global operating system → designed to manage node operating system, provides light-weight concurrency, system-wide power control, runtime resource management, resilience and fault management
- One of the 3 projects funded under the Department of Energy ExaOSR initiative

**Core Ideas**

- ENCLAVES
  - Groups of nodes share the same configuration
  - Enclave-specific Master node handles management of that enclave
  - Enclaves in their lifetime can change in size and be recursively divided into sub-enclaves
  - A Root enclave exists for the global system

- DISTRIBUTED MANAGEMENT
  - Enclaves managed hierarchically and control distributed across masters
  - Masters of parent enclaves have priority over masters of children
  - Reaction to events for fault, environment/configuration changes are distributed across masters
  - Privileged operations (admin, shutdown) are located on root enclaves

**ARGO**

- COMMUNICATION BACKPLANE
  - Part of the Global Information Bus (GIB)
  - BEACON: Backplane scalable communication using publish-subscribe model and is used by all layers of ARGObots software to exchange information
  - EXPOSE Backend provides performance introspection, in situ analysis and feedback mechanisms

- ARGObots
  - Efficient runtime systems to exploit the massive on-node parallelism
  - A new low-level threading/tasking model that exposes hardware characteristics of exascale systems effectively
  - Explore new libraries and high-level tasking frameworks that can take advantage of such low-level model

**STATUS**

- APIs, select software releases and publications can be found at: [www.argo-csr.org](http://www.argo-csr.org)
- Prototype implementation of system-wide Global OS → Built on top of OpenStack services. Implementation uses bare metal provisioning and provides enclave creation and tracking, configuration of system services and job launching
- Distributed enclave and system-wide power management algorithms are a part of Global OS
- Prototype implementation of BEACON (on EVPATH and RIUK KV)
- NodeOS provides partitioning of CPU and memory resources, a prototype implementation of the compute containers and custom scheduling policy for modern HPC runtimes
- Techniques to exploit NVRAM using DI-MMAP will help transparently incorporate NVRAM in memory hierarchies for applications
- Successfully demonstrated Argobots integration with several programming models: MPI, OpenMP, Charm++, Cilk, PIGE
- Collaboration with RIKEN in Japan led to highly scalable OpenMP for nested and irregular loop nests on top of Argobots
- Initial Argobots and Argobots-MPI prototype implementation completed. Development of Cilk + Argobots in progress

**Managing Power in ARGO**

1. User submits several jobs which are launched in their enclaves by Global OS
2. TAU software monitors sensors and publishes this info through BEACON
3. Various components receive this information
4. Global OS decides to reduce power in an enclave and publishes a request via BEACON
5. Node OS components (such as NodeRM: Node Resource Manager) in that enclave receive this command and decide to shutdown a core and ask Argobots for approval
6. Argobots works with higher-level libraries and applications to shutdown an execution stream and inform NodeRM
7. NodeRM shuts down a core

**ACKNOWLEDGMENTS**

This work is supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computer Research, under Contract DE-AC02-06CH11357