Improved Operating System and Resource Management in the Exascale Era

ANL: Pete Beckman (PI), Sridutt Bhalachandra, Nicolas Denoyelle, Rinku Gupta, Kamil Iskra, Florence Monna, Swann Perarnau, Valentin Reis, John Tramm, Brice Videau, Kazutomo Yoshii

LLNL: Maya Gokhale (co-PI), Marty McFadden, Tapasya Patki, Roger Pearce

Developing vendor-neutral, open-source software for OS/R improvements

The goal of Argo is to improve or augment existing OS/R components for use in production HPC systems, providing portable, open source software that improves the performance and scalability and that provides increased functionality to exascale applications.

Global Resource Management

Enclave-based, user-facing facilities
- Application-facing abstraction over resource management infrastructure
- Extendable collection of distributed services
- Encapsulation and configuration of job resources
- Recursive, dynamic (create, destroy, reconfigure)
- ECP goals: integration with Flux, MPI, workflows

Power
- Global Resource Manager: continuously track power needs and consumption; steer power where it can most advance the application’s progress
- Decentralized approach including managers at enclave and node levels
- ECP goals: production-ready GRM, integration across hierarchy levels, collaboration with Flux, collaboration on ERM with ECP power runtime

Node Resource Management

Containers
- Performance isolation through partitioning of physical node resources (CPU, memory)
- Accommodate coupled codes, in situ analytics, workflows
- Separation of system services from applications
- ECP goals: co-scheduling of resources, interfaces for container management, unified Node Resource Manager, integration with power management

Power
- Container-aware application performance monitoring
- Node-wide monitoring of power, temperature, system events and other sensors
- Enforcement of GRM-provided limits using monitoring feedback
- ECP goals: improve power-control policies, use application-reported progress

Memory Management

UMap: Incorporating NVRAM into the memory hierarchy
- Mmap replacement for Out-of-Core Data
- Application (class) specific page fault resolution, page size, page buffer size
- Uses User Fault FD protocol of Linux
  - Lightweight asynchronous messaging from the kernel to user mode handler on page faults
- Runs on LLNL Sierra compute nodes

AML: Exposing deep memory hierarchy to users
- Software-managed scratchpad for multilevel DRAM hierarchy (e.g., HBM and DDR)
- Asynchronous memory migration between hierarchy levels using dedicated CPU threads
- ECP goals: evolve initial proof-of-concept, generalize migration mechanisms (incl. HW), develop suitable user-facing API