A Single-Kernel Approach to OS Specialization and Node Resource Partitioning for Exascale Computing

Judicael A. Zounmevo, Kamil Iskra, Kazutomo Yoshii, Roberto Gioiosa,
Brian C. Van Essen, Maya B. Gokhale, Edgar A. Leon

Argonne National Laboratory. {zounmevo@, iskra@mcs., kazutomo@anl.gov}
Pacific Northwest National Laboratory. Roberto.Gioiosa@pnnl.gov

*Lawrence Livermore National Laboratory. {vanessen1, maya, leon}@llnl.gov

General purpose OS may not fulfill exascale performance and scalability needs

HPC job

HPC runtime

OS

Device drivers

Hardware

Centralized kernel data

Not speed-scalable

Knowledge of parallelism orchestration

Per-CPU core kernel data

Not resource (memory)-scalable

Information on resource needs

OR

Exascale system hardware

Deep and complex memory hierarchy

Per-CPU core data

Exacheates

Notable hardware

... and there is a strong need for latency propagation avoidance

Node OS and runtime view of the Argo exascale OS architecture

OS specialization and resource partitioning

Application

Vendor RTS

Concurrent RTS

Legacy API

Resource manager

Dynamic adaptation

Storage manager

Call forwarding

Legend

Accelerator

General-purpose CPU core

Special-purpose CPU core

RTS = Runtime system

Some novelties under development

Single kernel; differentiated behaviors

Feature

Feature

Feature

Feature

Feature

Feature

Feature

ServiceOS

ComputeOS

ComputeOS

ComputeOS

Fully-fledged

Lean

New resource controllers for kernel feature activation/deactivation
Cgroup interface is being leveraged

Logical memory nodes

Partitioning into finer-grained logical memory nodes

Static PTE mapping + Free memory pinning + fast RDMA

Smart IRQ migration

Interrupt

Of interest

Process

An IRQ kept close to ComputeOS process