Chameleon & Argo: Experiments in Exascale System Software

Swann Perarnau
Argonne National Laboratory

September 13rd, 2017
What to Expect of an Exascale System

Hardware
Projected for 2020-25.
- Compute Nodes: $O(100\,000)$
- CPU Cores/node: $O(100)$
- Interconnect: $>3$ dimensions
- Power: around 30 MW

Consequences
- High intra-node parallelism
- OS must integrate/abstract new technologies
- High failure rate
- Complex resource management
A Software Stack for Exascale: Argo

Project goal

Design and prototype a system software and runtime stack for exascale.

Project members

D.O.E funded, 3 national labs, 4 universities

- ANL, LLNL, PNNL
- UTK, UIUC, UO, UChicago

About 35 people.
Argo (2)

Argobots

Modern Runtime for high intra-node parallelism.
- User-level task and threading model
- Interactions with OS and Communication libraries

NodeOS

Linux with HPC specializations.
- Compute Containers: partitioning instead of isolation/sharing.
- New scheduler and memory subsystems.
Argo (3)

Global Information Bus

High-level components for complex communications requirements.

- Pub-sub system, data aggregation, ...
- Designed with failures and dedicated networks in mind.

GlobalOS

Distributed management of the entire system.

- Encapsulate configuration and policies inside group of nodes (enclaves).
- Dedicated nodes for system management across the machine.
NOT in Argo

Out of scope

- I/O
- MPI (need to be compatible)
- Sysops

Interaction with other components

- Batch Scheduler
- Power/Boot switches
- Workflows
**My Work at Argonne**

**50 % GlobalOS**
Design, prototype, and evaluate the infrastructure to configure, control, and monitor the resources of an exascale machine.

**50 % NodeOS**
- GlobalOS/NodeOS interactions (containers, resource management),
- Study future memory systems.

**50 % Integration**
Build and maintain our integration platform on top of Chameleon.
Outline

1. Introduction

2. Chameleon Experiments: Power Management

3. Discussion
Plan

1 Introduction

2 Chameleon Experiments: Power Management

3 Discussion
Several Issues

- Limited global power budget,
- Variability in manufacturing process,
- Imbalance in work distribution.

Our Approach

- Distribute power management across the enclaves hierarchy,
- Use feedback from application/job to balance power/performance,
- As many control levers as possible.
Global Power Management

Role
- Receive global limit from outside sources (sysops),
- Use the enclave hierarchy to recursively distribute power.

Features
- Use GIB to monitor power consumption across enclaves,
- Can split power between enclaves, and between nodes in an enclave.
- (Future Work) Closed loop: use performance monitoring to help decision.
Node Power Management

Role

- Receive power budget from enclave manager,
- Apply power budget using available levers.

Features

- Use GIB to monitor power and performance of application.
- Hardware levers: DVFS and RAPL.
- Software levers: can tell the runtime to reduce its core usage.
Control baremetal cluster;
2 Install NFS, Job scheduler, system services;
3 Partition cluster in two separate jobs;
4 Boot on each node our NodeOS;
5 Isolate applications from OS noise;
6 Launch one MPI application per job, using tasking runtime internally
7 Monitor power consumption
8 Adjust power dynamically by asking each node to reduce/increase its consumption;
9 Upon power budget changes, ask the node runtime to remove/add threads to the application
Video: SC15 Demo
Machine Setup

Overall design
- Scripts against OpenStack API for node creation, boot.
- Ansible playbooks for NFS, scheduler, system configuration.
- Job scheduler triggers node boot/shutdown dynamically.
- Whole-disk images for NodeOS installation.
- Containers for resource partitioning.
- Custom MPI launcher to spawn into containers directly.

Allocation Issues
First version on weeks-long node reservation
→ Most compute nodes idle when no job is running.
Second version using KVM cloud for frontend, NFS, scheduler
→ Dynamic reservations for compute nodes.
Plan

1. Introduction
2. Chameleon Experiments: Power Management
3. Discussion
On Chameleon

Argo was a challenging user at the beginning
- Missing low-level controls: whole-disk images, boot configuration.
- Week-long allocations of 30+ nodes.
- NTP/DNS access, highly concurrent node creation/deletion.

We are really happy with it
- Self-managed platform was a nightmare.
- Very responsive and helpful admins.
- Modern hardware, regular software updates.
More automation

- Appliances?
- Would really like to do CI/CD on Chameleon...

What we're still missing

- BIOS reconfiguration
- Deep Memory systems (KNL)
- VPN/direct communication between 3 ‘sites’
Any Questions?