Argonne NATIONAL LABORATORY

Cobalt: An Open Source Platform for HPC System Software Research

Narayan Desai

desai@mcs.anl.gov

04-20-2006

THE UNIVERSITY OF CHICAGO

Office of Science U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by The University of Chicago

Overview

- Motivations
- Design Goals
- Architecture
- BG/L Features
- Development Status
- Future Areas of Work

Motivations

- Needed to support both computational and computer science users
- System software developers have different needs from computational scientists
  - System hangs are common, even desired
  - A large variety of configurations are required, sometimes simultaneously
  - The *“application”* can span all software on a node
- System failures are more common during system software research and development
  - System software must deal gracefully with faults
  - Most resource managers not suitable for system software research environments

Motivations (cont)

- No resource manager shipped with our BG/L system
- How hard could it be?

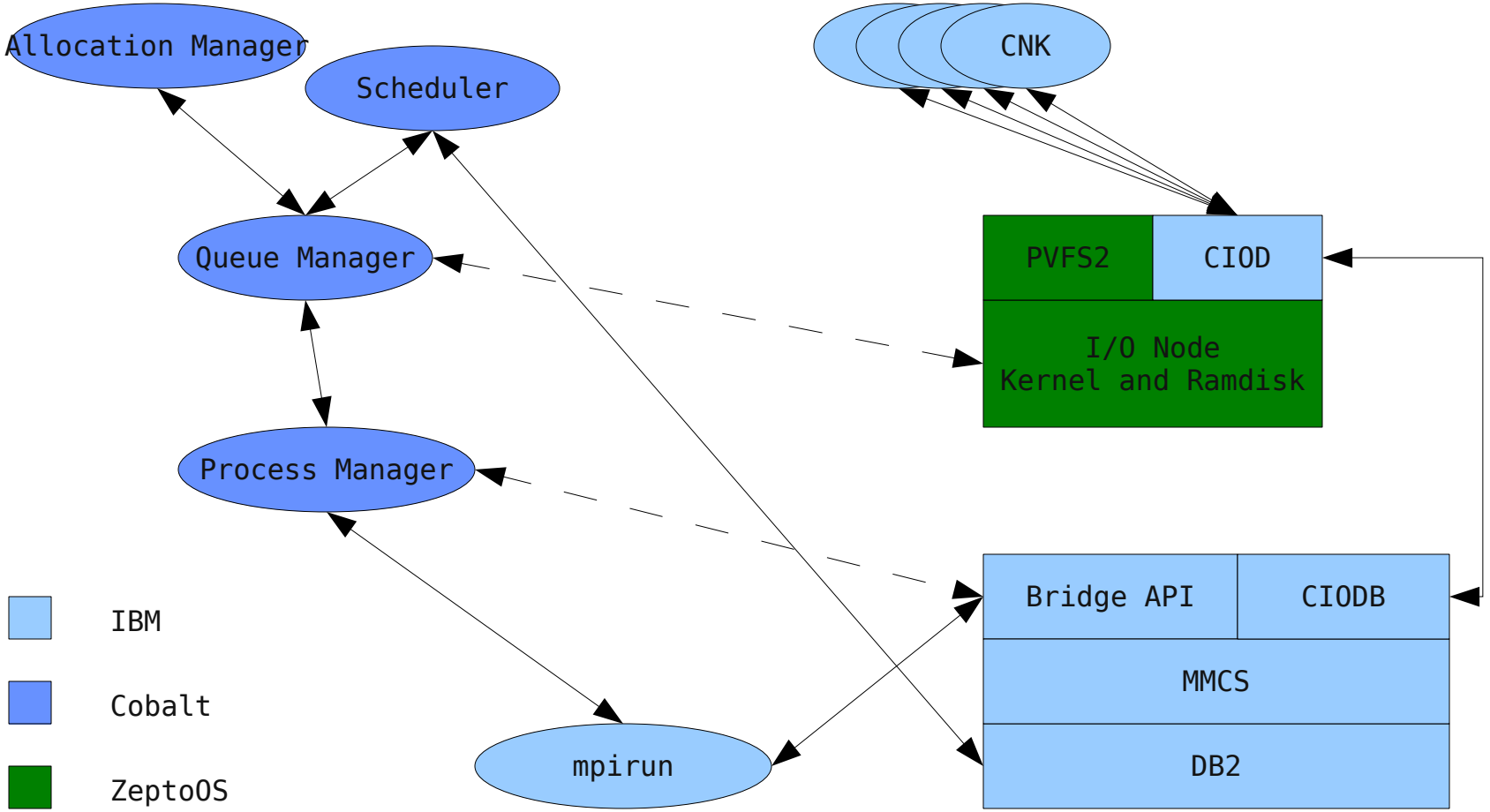
Design Goals

- Suitable for “MCS style” machines
  - Mix of computer science and computational science
  - Neither side dominates the other
- RISC approach to system software
  - As simple as possible, no simpler
- Agility
- Portability
- Extensibility

Architecture

- Component Architecture
  - First revision loosely based on the SciDAC Scalable System Software interfaces
- Smaller and simpler is better
  - somewhat feature poor (but getting better fast)
  - ~2700 lines of python code
  - ~600 lines are BG/L specific
- Its agility makes it the perfect research platform
  - Easy to prototype new ideas
  - Adding new features is easy and quick
    - *2 minutes is the record bugfix time to deployment*
    - *3 minutes is the record feature addition time to deployment*
- If you don't like a component's implementation, write a new one

Cobalt on BG/L



Scheduling on BG/L

- Partitions are defined for scheduling purposes
  - Includes size, queue, etc
- One partition definition per location for user jobs
  - Partitions can overlap, but dependencies need to be defined
- The scheduler effectively packs jobs onto the machine
- Greedy backfill is implemented
- Reservations
- Per-Queue policies
  - default (fifo + backfill)
  - short queue (< 30 minute jobs)
  - easy to implement more

Dynamic Kernel Support on BG/L

- User-setup kernel profiles
  - includes CNK, ION kernel, ION ramdisk, and loader
- Each partition configured with a partition specific boot location
- User jobs include a kernel profile
  - with a default profile of “default”
- The partition specific boot location is a symlink
- Cobalt modifies this link during each job, once execution location has been established
- The partition boots the specified kernel upon job startup

Development Status

- Major rewrite just finished
  - Reduced code from 5K lines of code and a lot of prereqs to 2.8K and one prereq
  - Streamlined deployment and management processes
  - Basic accounting support
- Cobalt has been running in production for over 1 year
  - At ANL and NCAR
- Cobalt is deployed at several sites worldwide
  - Including AIST and NIWS in Japan
- Open development process
  - Suggestions and patches are both welcome
  - NCAR has helped substantially with code and documentation improvements

Active Development Areas

- Scheduler Improvements
  - More sophisticated multi-rack allocation policies
  - More efficient backfill
  - Investigate rule-based scheduling policies
  - Periodic scheduling policies
  - Reading Susan's mind
- Support for user specified ZeptoOS options
- Full allocation management/accounting functionality
- Better user interfaces
  - Dynamic web pages, etc

Results

- Small code base allows easy modifications
  - Usability improvements
  - New features (~3 minutes is the current record)
  - Site-specific customizations
  - Porting to new systems is quite easy
- Properly arbitrating between system software developers and computational science users
  - Allows on-the-fly system configuration changes
  - Ensures that computational jobs get non-development versions of system software
  - Able to protect each user group from the other and its software requirements
- Therapeutic effect on sysadmin blood pressure
  - System software small enough to be readily understood and modified

The End

- Questions?

http://www.mcs.anl.gov/cobalt