Right-Weight Kernels: an off-the-shelf alternative to custom Light-Weight Kernels

ACM SIGOPS Operating Systems Review, Vol. 40 (April 2006)

Minnich, Sottile, Choi, Hendriks, McKie @ Department of Energy's (DOE), Los Alamos National Laboratory (LANL), IBM

January 20, 2010

Introduction

- HPC folks, mission and early result paper

Introduction

- HPC folks, mission and early result paper
- Parallel system with large number of nodes running parallel application
- Desire: application want to own nodes
- Any Interference can cause poor performance

Interference

- Any non-application activity on nodes reducing peak performance
- Delay of one processor out of 1000 wreak havoc to parallel performance

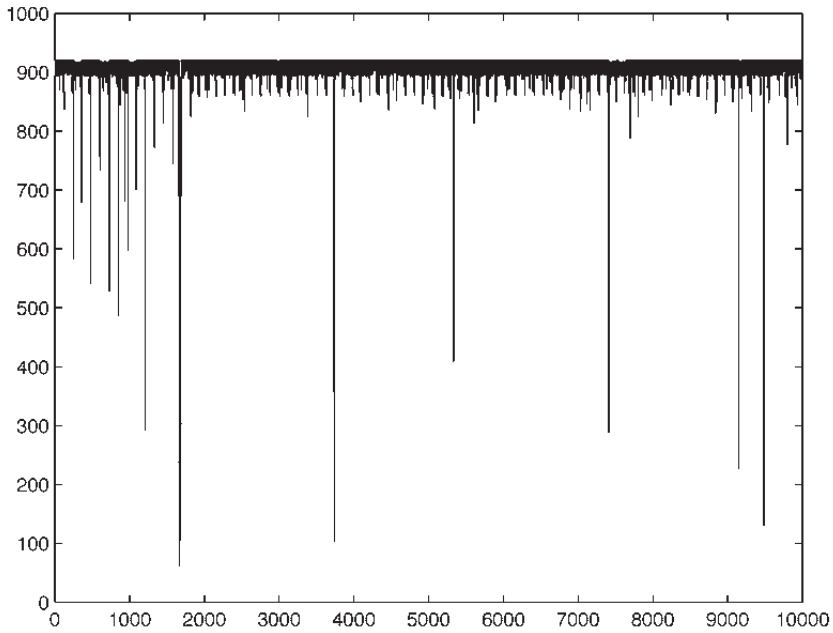

Figure: x: time, y: amout of work

Examples

- ASCI Red: replacing OSF/1-MK ADh from Intel by OS with fewer features
- IBM SP/2: making simple local scheduling decisions
- ASCI Q: removing unnecessary daemons
- IBM, Red Storm, CPlant machines, ...

Clusters and other distributed machines

- Group of nodes dedicated to application
  - Claim: don't eliminate interference
  - Internal activity triggered by clock interrupts
  - Daemons becoming active when network packets arrive

Clusters and other distributed machines

- Group of nodes dedicated to application
  - Claim: don't eliminate interference
  - Internal activity triggered by clock interrupts
  - Daemons becoming active when network packets arrive
- Cluster community approach
  - Bypass parts of kernel regarding networking
  - Application directly access hardware

Clusters and other distributed machines

- Group of nodes dedicated to application
  - Claim: don't eliminate interference
  - Internal activity triggered by clock interrupts
  - Daemons becoming active when network packets arrive
- Cluster community approach
  - Bypass parts of kernel regarding networking
  - Application directly access hardware
- Light-Weight Kernel (LWK)
  - Elimination of almost all capabilities of a kernel
  - Application after start take over node
  - LWK provides basic function for I/O
  - no file system, no sockets, no virtual memory, no security model

Right-Weight Kernels

- LWKs go to far ?! anecdotal evidence
- Discrepancy:
  - Users want the os "out of the way" vs.
  - Convinient: shared lib, reliable file system, sockets, security, app fault management and debug support
- LANL assumptions:
  - LWK not necessary, use adapted off-the-shelf OSes, simulation tools
  - Candidates: Linux and Plan 9
  - Linux: make it more light-weight to avoid interferences
  - Plan 9: Designed as distributed system, Used in hard-real-time environments (routers)

Linux

- Pink, 1024 node BProc cluster
- BProc (Beowulf Distributed Process Space): set of Linux kernel patches
- Single-system process space across entire cluster
- Application processes on slave nodes show up in the process table of the master node
- Per compute(slave) node one BProc daemon running
- (Expected) significant interruptions by:
  - Periodic timer interrupts
  - Kernel threads for internal book keeping, flushing blocks ...

Plan 9

- By Bell Laps since 1990's
- Hybrid of LWK and commodity OS
- Kernel: devices, process management, network protocol stack
- Server: file system, ...
- Customization: server free placeable by user at nodes

Conclusion and Discussion

- "We hope to learn how to configure a kernel that is the right weight for HPC - i.e. a Right-Weight Kernel"

Conclusion and Discussion

- "We hope to learn how to configure a kernel that is the right weight for HPC - i.e. a Right-Weight Kernel"
- Requirements they seem to have ...
  - Customizable OS
  - Strict placing of processes and services
  - Transparent (what is running when and where)
- Questions:
  - General (non-solvable) issue ? Trading performance vs usability/maintain vs security vs ... ?
  - Bad ? : periodic/aperiodic work, batching, daemons, flushing, network ...