FusedOS: Towards Hybrid Workload of HPC and Big Data Analytics

Kyung Dong Ryu, PhD
Manager, Exascale System Software Research Group
IBM T.J. Watson Research Center

Co-authors:
Yoonho Park, Marius Hillenbrand, Bryan Rosenberg, Robert W. Wisniewski, Todd Inglett
Vision: System Convergence for HPC and Big Data Analytics

- **High Performance Computing**
  - 10s - 100s Peta Flops of Compute
  - CPU-oriented
  - Special SW stack (LWK, MPI)

- **High Performance Analytics**
  - 10s - 100s Peta Bytes of Data
  - Storage-oriented
  - Generic SW stack (Linux, Hadoop)

**Data-Centric Deep Computing** for Deep Insight from Big Data
Motivation

- OS Challenges
  - Users want both LWK performance AND FWK functionality (Linux)
  - Emerging Hybrid Workload with HPC and Big Data Analytics coupled through Workflow

- Previous approaches
  - Extend Linux to obtain LWK performance
  - Extend an LWK to provide FWK functionality

- Our approach: FusedOS
  - Fuse LWK and FWK
  - Combine personalities
Outline

- Introduction
- Approach
- Design
- Implementation
- Evaluation
  - Application performance
  - System call performance
  - SLOC
- Related work
- Future work
- Conclusions
Approach

- Specialization - Assign different partitions different roles
- Specialization accommodates heterogeneity
- Partition resources such as cores and memory
- What is possible without hardware virtualization?
• Linux for FWK and CL (CNK in Library) for LWK
• STOC and PEC represent heterogeneity
  o STOC = Single-Thread Optimized Core, PEC = Power-Efficient Core
• Provide PEC apps with direct access to physical resources
Implementation

- CNK apps have direct access to physical resources
Implementation

- CNK apps have direct access to physical resources
- Linux manages physical resources
Implementation

- CNK apps have direct access to physical resources
- Linux manages physical resources
- CL handles CNK system calls
  - CL accesses physical resources through file system interfaces
Implementation

- CL
  - Load application in memory
  - Start application (load TLB & registers)
  - System call/exception
  - Service event
  - Resume/stop

- STOC

- PEC

- Application
Blue Gene Q

Node Card:
32 Compute Nodes
+ Optical Modules, Link Chips;
5D Torus

Compute Node:
Chip module, Memory

System:
e.g. 96 racks (Sequoia)

Chip:
16+2 µP cores

Single Chip Module

I/O drawer:
8 IO cards w/16 GB
8 PCIe Gen2 x8

Rack:
2 Midplanes
1,2,4 I/O drawers

Midplane:
16 Node Cards
Blue Gene/Q Prototype

• Each node has 17 homogeneous cores, each providing 4 hardware threads
  o Simulated STOCs and PECS
  o PEC monitor provides "PEC management interface"
  o 1 core used as a STOC, 16 cores used as PECs
• Each node has 16 GB of physical memory
  o 4 GB allocated to Linux, 12 GB for PECs
Implementation
Sequoia FTQ
Sequoia LAMMPS (OpenMP)

<table>
<thead>
<tr>
<th>Procs x Threads</th>
<th>CNK</th>
<th>FusedOS</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x 1</td>
<td>4753.6 s</td>
<td>-0.1%</td>
<td>+2.7%</td>
</tr>
<tr>
<td>16 x 2</td>
<td>2977.7 s</td>
<td>+3.4%</td>
<td>+9.4%</td>
</tr>
<tr>
<td>16 x 4</td>
<td>1967.9 s</td>
<td>+7.0%</td>
<td>+24.7%</td>
</tr>
<tr>
<td>1 x 8</td>
<td>829.0 s</td>
<td>+2.6%</td>
<td>+7.5%</td>
</tr>
<tr>
<td>8 x 8</td>
<td>1160.5 s</td>
<td>+10.0%</td>
<td>+24.7%</td>
</tr>
<tr>
<td>1 x 16</td>
<td>555.7 s</td>
<td>+5.4%</td>
<td>+7.8%</td>
</tr>
<tr>
<td>4 x 16</td>
<td>761.7 s</td>
<td>+13.5%</td>
<td>+27.9%</td>
</tr>
<tr>
<td>1 x 32</td>
<td>509.5 s</td>
<td>+5.6%</td>
<td>+10.9%</td>
</tr>
<tr>
<td>2 x 32</td>
<td>603.2 s</td>
<td>+10.4%</td>
<td>+24.5%</td>
</tr>
<tr>
<td>1 x 64</td>
<td>557.1 s</td>
<td>+3.2%</td>
<td>+20.1%</td>
</tr>
</tbody>
</table>
System Call Latency

<table>
<thead>
<tr>
<th>Function</th>
<th>Linux</th>
<th>CNK</th>
<th>FusedOS IPI</th>
<th>FusedOS waitrsv</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td>0.028 us</td>
<td>0.028 us</td>
<td>0.027 us</td>
<td>0.027 us</td>
</tr>
<tr>
<td>getpid</td>
<td>0.401 us</td>
<td>0.250 us</td>
<td>8.908 us</td>
<td>2.280 us</td>
</tr>
</tbody>
</table>
It is available now!

- **Ease of Use**
  - Compute node Linux/CNK
  - Linux and CNK applications on same node

- **Open** - [http://github.com/ibm-research](http://github.com/ibm-research)
  - Eclipse Public License and GPL
  - Available for BGQ

- **Free** - Fedora 19 base
Related Work

- Lightweight OS design - Catamount, Kitten
- Improve Linux - K42, LibraOS, Cray Compute Node Linux, ZeptoOS
- System services in user space - Exokernel, SPIN, VINO
- Scalable multicore OS - Corey, fos, Tessellation
- Accommodate heterogeneity - Barrelish, Helios
- VMM for HPC - Palacios
- Close cousin - NIX
Conclusions

• FusedOS, a hybrid approach, can provide LWK performance (CNK) and FWK functionality (Linux)
  o Minimal changes to LWK and FWK
• FusedOS is well positioned to support heterogeneous cores
• FusedOS represents a point in the exploration of exascale operating environments, which are projected to be power constrained and heterogeneous
Backup
<table>
<thead>
<tr>
<th></th>
<th>CNK</th>
<th>CL All</th>
<th>CL New</th>
<th>Monitor</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++</td>
<td>22,390</td>
<td>9,814</td>
<td>796</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>9,532</td>
<td>2,609</td>
<td>127</td>
<td>153</td>
<td>1,215 (18)</td>
</tr>
<tr>
<td>C/C++ Header</td>
<td>4,524</td>
<td>3,308</td>
<td>35</td>
<td>32</td>
<td>143</td>
</tr>
<tr>
<td>Assembly</td>
<td>1,640</td>
<td>0</td>
<td>0</td>
<td>618</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>38,086</td>
<td>15,731</td>
<td>958</td>
<td>803</td>
<td>1,358</td>
</tr>
</tbody>
</table>
Our World

- Multicore
  - Must employ parallelism to obtain performance as processor speeds level off

- Heterogeneity
  - Cores optimized for single-thread performance
  - Cores optimized for power-efficiency
  - Processing pushed to peripherals such as network interfaces, storage, and memory