Running 1 Million Jobs in 10 Minutes via the Falkon Fast and Light-weight task execution framework

Ioan Raicu
Distributed Systems Laboratory
Computer Science Department
University of Chicago

In Collaboration with:
Ian Foster, University of Chicago and Argonne National Laboratory
Mike Wilde, University of Chicago and Argonne National Laboratory
Zhao Zhang, University of Chicago
Yong Zhao, Microsoft
Catalin Dumitrescu, Fermi National Laboratory

Megajob BOF at IEEE/ACM Supercomputing 2008
November 18th, 2008
Obstacles running MTC apps in Clusters/Grids

Running 1 Million Jobs in 10 Minutes via the Falkon Fast and Light-weight task execution framework
Solution

• Falkon: A Fast and Light-weight task execution framework
  – **Goal:** enable the *rapid and efficient* execution of many independent jobs on large compute clusters
  – Combines three components:
    • A *streamlined task dispatcher*
    • *Resource provisioning* through multi-level scheduling techniques
    • *Data diffusion* and data-aware scheduling to leverage the co-located computational and storage resources
Falkon Overview

Running 1 Million Jobs in 10 Minutes via the Falkon Fast and Light-weight task execution framework
Running 1 Million Jobs in 10 Minutes via the Falkon Fast and Lightweight Task Execution Framework
Dispatch Throughput

<table>
<thead>
<tr>
<th>System</th>
<th>Comments</th>
<th>Throughput (tasks/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condor (v6.7.2) - Production</td>
<td>Dual Xeon 2.4GHz, 4GB</td>
<td>0.49</td>
</tr>
<tr>
<td>PBS (v2.1.8) - Production</td>
<td>Dual Xeon 2.4GHz, 4GB</td>
<td>0.45</td>
</tr>
<tr>
<td>Condor (v6.7.2) - Production</td>
<td>Quad Xeon 3 GHz, 4GB</td>
<td>2</td>
</tr>
<tr>
<td>Condor (v6.8.2) - Production</td>
<td></td>
<td>0.42</td>
</tr>
<tr>
<td>Condor (v6.9.3) - Development</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>Condor-J2 - Experimental</td>
<td>Quad Xeon 3 GHz, 4GB</td>
<td>22</td>
</tr>
</tbody>
</table>
Falkon Endurance Test

Running 1 Million Jobs in 10 Minutes via the Falkon Fast and Light-weight task execution framework.

Graph showing the throughput and completed tasks over time.
Falkon Activity History
(10 months)

Max CPUs: 163K
CPU Hours: 1.4M
Num Tasks: 164M
Task Exec: 31 sec

Allocated CPUs
Delivered Tasks

Allocated CPUs (60 sec average)

Completed Tasks
0 30 60 90 120 150 180
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180

1000000 100000 10000 1000 100 10 1
Falkon Demo

- **Workload**
  - 160K CPUs
  - 1M tasks
  - 60 sec per task
  - 17.5K CPU hours in 7.5 min
  - Throughput: 2312 tasks/sec
  - 85% efficiency
MARS Economic Modeling on IBM BG/P (128K CPUs)

- CPU Cores: 130816
- Tasks: 1048576
- Elapsed time: 2483 secs
- CPU Years: 9.3

Speedup: 115168X (ideal 130816)
Efficiency: 88%

Running 1 Million Jobs in 10 Minutes via the Falkon Fast and Light-weight task execution framework
DOCK on the BG/P

CPU cores: 118784
Tasks: 934803
Elapsed time: 2.01 hours
Compute time: 21.43 CPU years
Average task time: 667 sec
Relative Efficiency: 99.7%
(from 16 to 32 racks)
Utilization:
• Sustained: 99.6%
• Overall: 78.3%

Running 1 million Jobs in 10 Minutes via the Falkon Fast and Light-weight task execution framework
Collective IO Model

Application Script

Global FS

ZOID IFS for staging

ZOID on IO node

<-- Torus & Tree Interconnects -->

CN-striped IFS for Data

Large Input Dataset

IFS seg

IFS Compute node

IFS seg

IFS Compute node

LFS

Compute node
(local datasets)

LFS

Compute node
(local datasets)

Running 1 Million Jobs in 10 Minutes via the Falkon Fast and Light-weight task execution framework
Write Performance CIO vs. GFS efficiency

Running 1 Million Jobs in 10 Minutes via the Falkon Fast and Light-weight task execution framework
Data Diffusion

- Resource acquired in response to demand
- Data and applications diffuse from archival storage to newly acquired resources
- Resource “caching” allows faster responses to subsequent requests
  - Cache Eviction Strategies: RANDOM, FIFO, LRU, LFU
- Resources are released when demand drops
All-Pairs Workload
500x500 on 200 CPUs

Efficiency: 75%

Throughput (Gb/s)

Cache Miss %
Cache Hit Global %
Cache Hit Local %
Throughput (Data Diffusion)
Maximum Throughput (GPFS)
Maximum Throughput (Local Disk)
Mythbusting

- Embarrassingly parallel apps are trivial to run
  - Logistical problems can be tremendous
- Loosely coupled apps do not require "supercomputers"
  - Total computational requirements can be enormous
  - Individual tasks may be tightly coupled
  - Workloads frequently involve large amounts of I/O
  - Make use of idle resources from "supercomputers" via backfilling
  - Costs to run "supercomputers" per FLOP is among the best
    - BG/P: 0.35 gigaflops/watt (higher is better)
    - SiCortex: 0.32 gigaflops/watt
    - BG/L: 0.23 gigaflops/watt
    - x86-based HPC systems: an order of magnitude lower

- Loosely coupled apps do not require specialized system software
- Shared file systems are good for all applications
  - They don’t scale proportionally with the compute resources
  - Data intensive applications don’t perform and scale well
More Information

- More information: http://people.cs.uchicago.edu/~iraicu/
- Falkon: http://dev.globus.org/wiki/Incubator/Falkon
- Funding:
  - NASA: Ames Research Center, Graduate Student Research Program
    • Jerry C. Yan, NASA GSRP Research Advisor
  - NSF: TeraGrid
- Check out Falkon:
  - “svn co https://svn.globus.org/repos/falkon“