Resource Management in Extreme Scales Distributed Systems

Ioan Raicu
Computer Science Department, Illinois Institute of Technology
Math and Computer Science Division, Argonne National Laboratory

August 12th, 2013
Los Alamos National Laboratory
• **Research Focus**
  – Emphasize designing, implementing, and evaluating systems, protocols, and middleware with the goal of supporting **data-intensive applications on extreme scale distributed systems**, from many-core systems, clusters, grids, clouds, and supercomputers

• **People**
  – Dr. Ioan Raicu (Director)
  – 6 PhD Students
  – 2 MS Students
  – 4 UG Students

• **Contact**
  – [iraicu@cs.iit.edu](mailto:iraicu@cs.iit.edu)
• **Today (2013): Multicore Computing**
  - $O(10)$ cores commodity architectures
  - $O(100)$ cores proprietary architectures
  - $O(1000)$ GPU hardware threads

• **Near future (~2019): Manycore Computing**
  - $\sim 1000$ cores/threads commodity architectures

Pat Helland, Microsoft, The Irresistible Forces Meet the Movable Objects, November 9th, 2007

Resource Management in Extreme Scales Distributed Systems
• **Today (2013): Petascale Computing**
  – $O(100K)$ nodes
  – $O(1M)$ cores

• **Near future (~2018): Exascale Computing**
  – ~$1M$ nodes (10X)
  – ~$1B$ processor-cores/threads (1000X)
Exascale Computing Architecture

- **Compute**
  - 1M nodes, with ~1K threads/cores per node

- **Networking**
  - N-dimensional torus
  - Meshes

- **Storage**
  - SANs with spinning disks will replace today’s tape
  - SANs with SSDs might exist, replacing today’s spinning disk SANs
  - SSDs might exist at every node
Some Challenges to Overcome at Exascale Computing

• Programming paradigms
  – HPC is dominated by MPI today
  – Will MPI scale another 3 orders of magnitude?
  – Other paradigms (including loosely coupled ones) might emerge to be more flexible, resilient, and scalable

• Storage systems will need to become more distributed to scale ➔ Critical for resilience of HPC

• Network topology must be used in job management, data management, compilers, etc

• Power efficient compilers and run-time systems
• **Decentralization is critical**
  – Computational resource management (e.g. LRM)
  – Storage systems (e.g. parallel file systems)

• **Preserving locality is critical!**
  – POSIX I/O on shared/parallel file systems ignore locality
  – Data-aware scheduling coupled with distributed file systems that expose locality is the key to scalability over the next decade

• **Co-locating storage and compute is GOOD**
  – Leverage the abundance of processing power, bisection bandwidth, and local I/O
Critical Technologies Needed to achieve Extreme Scales

- Fundamental Building Blocks (with a variety of resilience and consistency models)
  - Distributed hash tables (aka NoSQL data stores)
  - Distributed Message Queues
- Deliver future generation distributed systems
  - Global File Systems, Metadata, and Storage
  - Job Management Systems
  - Workflow Systems
  - Monitoring Systems
  - Provenance Systems
  - Data Indexing
MTC emphasizes:
- bridging HPC/HTC
- many resources
  - short period of time
- many computational tasks
- dependent/independent tasks
- tasks organized as DAGs
- primary metrics are seconds

Advantages:
- Improve fault tolerant
- Maintain efficiency
- Programmability & Portability
- support embarrassingly parallel and parallel applications
Swift/T and Applications

• Swift/T
  o Active research project (CI UChicago & ANL)
  o Parallel Programming Framework
  o Throughput ~25k tasks/sec per process
  o Shown to scale to 128k cores

• Application Domains Supported
  o Astronomy, Biochemistry, Bioinformatics, Economics, Climate

Swift lets you write parallel scripts that run many copies of ordinary programs concurrently, using statements like this:

```swift
foreach protein in proteinList {
    runBLAST(protein);
}
```

Images from Swift Case Studies - http://www.ci.uchicago.edu/swift/case_studies/
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Characteristics</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Astronomy</td>
<td>Creation of montages from many digital images</td>
<td>Many 1-core tasks, much communication, complex dependencies</td>
<td>E</td>
</tr>
<tr>
<td>Astronomy</td>
<td>Stacking of cutouts from digital sky surveys</td>
<td>Many 1-core tasks, much communication</td>
<td>E (Falkon)</td>
</tr>
<tr>
<td>Biochemistry</td>
<td>Analysis of mass-spec data for post-translational protein modifications</td>
<td>10,000 – 100,000 K jobs for proteomic searches using custom serial codes</td>
<td>D</td>
</tr>
<tr>
<td>Biochemistry</td>
<td>Protein folding using iterative fixing algorithm, also exploring other biomolecule interactions</td>
<td>100s to 1000s of 1-1000 core simulations &amp; data analysis</td>
<td>O</td>
</tr>
<tr>
<td>Biochemistry</td>
<td>Identification of drug targets via computational screening</td>
<td>Up to 1M x 1 core</td>
<td>O (Falkon)</td>
</tr>
<tr>
<td>Bioinformatics</td>
<td>Metagenome modeling</td>
<td>1000’s of 1-core integer programming problems</td>
<td>D</td>
</tr>
<tr>
<td>Business economics</td>
<td>Mining of large text corpora to study media bias</td>
<td>Analysis and comparison of 70M+ text files of news articles</td>
<td>D</td>
</tr>
<tr>
<td>Climate</td>
<td>Ensemble climate model runs and analysis of output data</td>
<td>10s to 100s of 100-1000 core simulations</td>
<td>E</td>
</tr>
<tr>
<td>Economics</td>
<td>Generation of response surfaces for various economic models</td>
<td>1K to 1M 1-core runs (10K typical), then data analysis</td>
<td>O</td>
</tr>
<tr>
<td>Neuroscience</td>
<td>Analysis of functional MRI datasets</td>
<td>Comparison of images; connectivity analysis with SEM, many tasks (100K+)</td>
<td>O</td>
</tr>
<tr>
<td>Radiology</td>
<td>Training of computer aided diagnosis algorithms</td>
<td>Comparison of images; many tasks, much communication</td>
<td>D</td>
</tr>
<tr>
<td>Radiology</td>
<td>Image processing and brain mapping for neurosurgical planning research</td>
<td>1000’s of MPI application executions</td>
<td>O</td>
</tr>
</tbody>
</table>
Active Projects

- **Storage**
  - FusionFS: Fusion distributed File System
    - HyCache, FusionProv, IStore, RXSim
  - ZHT: Zero-Hop Distributed Hash Table
    - NoVoHT

- **Computing**
  - Many-Task Computing
    - MATRIX: MAny-Task computing execution fabRIc at eXascales
      - SimMatrix
    - Falkon: Fast and Light-weight tasK executiON framework
      - FalkonCloud
    - Swift: Fast, Reliable, Loosely Coupled Parallel Computation
  - Many-Core Computing
    - GeMTC: Virtualizing GPUs to Support MTC Applications
  - Cloud Computing
    - CloudBench: Optimizing Cloud Infrastructure for Scientific Computing Applications

Resource Management in Extreme Scales Distributed Systems
Proposed Software Stack in Large-Scale Distributed Systems

Applications

Many-Task Computing
(SwiftScript, Charm++, MapReduce)

High-Performance Computing
(MPI)

Distributed Execution Fabric
(MATRIX)

Resource Manager
(Cobalt, SLURM)

Persistent Distributed Hash Tables (ZHT)

Distributed File Systems (FusionFS)

Parallel File Systems
(GPFS, PVFS)

High-End Computing Hardware
(Petascale to Exascale Systems)

Hardware (Terascale)

Simulator (SimMatrix)
ZHT: A distributed Key-Value store
- Light-weighted
- High performance
- Scalable
- Dynamic
- Fault tolerant
- Strong Consistency
- Persistent
- Versatile: works from clusters, to clouds, to supercomputers
• Many DHTs: Chord, Kademlia, Pastry, Cassandra, C-MPI, Memcached, Dynamo

• Why another?

<table>
<thead>
<tr>
<th>Name</th>
<th>Impl.</th>
<th>Routing Time</th>
<th>Persistence</th>
<th>Dynamic membership</th>
<th>Append Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cassandra</td>
<td>Java</td>
<td>Log(N)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>C-MPI</td>
<td>C</td>
<td>Log(N)</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Dynamo</td>
<td>Java</td>
<td>0 to Log(N)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Memcached</td>
<td>C</td>
<td>0</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ZHT</td>
<td>C++</td>
<td>0 to 2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Resource Management in Extreme Scales Distributed Systems
Resource Management in Extreme Scales Distributed Systems
Resource Management in Extreme Scales Distributed Systems

ZHT Project

ZHT on cc2.8xlarge instance
8 s-c pair/instance

<table>
<thead>
<tr>
<th>SCALES</th>
<th>75%</th>
<th>90%</th>
<th>95%</th>
<th>99%</th>
<th>AVG</th>
<th>THROUGHPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>186</td>
<td>199</td>
<td>214</td>
<td>260</td>
<td>172</td>
<td>46421</td>
</tr>
<tr>
<td>32</td>
<td>509</td>
<td>603</td>
<td>681</td>
<td>1114</td>
<td>426</td>
<td>75080</td>
</tr>
<tr>
<td>128</td>
<td>588</td>
<td>717</td>
<td>844</td>
<td>2071</td>
<td>542</td>
<td>236065</td>
</tr>
<tr>
<td>512</td>
<td>574</td>
<td>708</td>
<td>865</td>
<td>3568</td>
<td>608</td>
<td>841040</td>
</tr>
</tbody>
</table>

DynamoDB: 8 clients/instance

<table>
<thead>
<tr>
<th>SCALES</th>
<th>75%</th>
<th>90%</th>
<th>95%</th>
<th>99%</th>
<th>AVG</th>
<th>THROUGHPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>11942</td>
<td>13794</td>
<td>20491</td>
<td>35358</td>
<td>12169</td>
<td>83.39</td>
</tr>
<tr>
<td>32</td>
<td>10081</td>
<td>11324</td>
<td>12448</td>
<td>34173</td>
<td>9515</td>
<td>3363.11</td>
</tr>
<tr>
<td>128</td>
<td>10735</td>
<td>12128</td>
<td>16091</td>
<td>37009</td>
<td>11104</td>
<td>11527</td>
</tr>
<tr>
<td>512</td>
<td>9942</td>
<td>13664</td>
<td>30960</td>
<td>38077</td>
<td>28488</td>
<td>ERROR</td>
</tr>
</tbody>
</table>
NoVoHT Project

• NoVoHT
  – Persistent in-memory hash map
  – Append operation
  – Live-migration
• A distributed file system co-locating storage and computations, while supporting POSIX
• Everything is decentralized and distributed
• Aims for millions of servers and clients scales
• Aims at orders of magnitude higher performance than current state of the art parallel file systems
This gap will grow even larger as parallel filesystems saturate external network – expected gap will be ~4 orders of magnitude faster performance.

Resource Management in Extreme Scales Distributed Systems
• 16K-node scales
  - FusionFS 2500GB/s (measured) vs. GPFS 64GB/s (theoretical)
  - 39X higher sustained throughput

• Full system 40K-node scales
  - Expected Performance: 100X higher I/O throughput
  - Expected Performance: 4000X higher metadata ops/sec
Many sub-projects
- Provenance (FusionProv) – uses ZHT
- Information Dispersal Algorithms (IStore) – uses GPUs

Other relevant Projects (planning to integrate into FusionFS)
- SSD+HHD hybrid caching (HyCache)
- Data Compression

Improvements on the horizon
- Non-POSIX interfaces (e.g. Amazon S3)
- Explore viability of supporting HPC checkpointing
- Deep indexing and search
SimMatrix Project

- Light-weight simulator to study MTC scheduling algorithms at exascale levels and on many-core architectures
Matrix Project

- MATRIX - distributed MTC execution framework for distributed load balancing using Work Stealing algorithm
  - Distributed scheduling is an efficient way to achieve load balancing, leading to high job throughput and system utilization
  - Dynamic job scheduling system at the granularity of node/core levels for extreme scale applications
• Bag of Tasks
• Fan-In DAG
• Fan-Out DAG
• Pipeline DAG
• Complex Random DAG
Resource Management in Extreme Scales Distributed Systems
Resource Management in Extreme Scales Distributed Systems
MATRİX Project

Resource Management in Extreme Scales Distributed Systems

![Graph showing efficiency trends for different scales and methods: Bag of Tasks, Fan In, Fan Out, Pipeline. The efficiency decreases as the scale increases.](image-url)
**GPU**
- Streaming Multiprocessors (15 SMXs on Kepler K20)
- 192 warps * 32 threads

**Coprocessors**
- Intel Xeon Phi
- 60 cores * 4 threads per core = 240 hardware threads

**GeMTC**
- Efficient support for MTC on accelerators
GeMTC Project

![Graph showing efficiency vs. sleep time for different processors and worker counts.](image)
Active Collaborations
National Labs and Industry

• National Laboratories
  – ANL: Kamil Iskra, Rob Ross, Mike Wilde, Marc Snir, Pete Beckman, Justin Wozniak
  – FNAL: Gabriele Garzoglio
  – LANL: Mike Lang
  – ORNL: Arthur Barney Maccabe
  – LBL: Lavanya Ramakrishnan

• Industry
  – Cleversafe: Chris Gladwin
  – EMC: John Bent
  – Accenture Technology Laboratory: Teresa Tung
  – Microsoft: Roger Barga
  – SchedMD: Morris Jette, Danny Auble
  – Oracle: Hui Jin
  – INRIA: Gabriel Antoniu
  – IBM: Bogdan Nicolae
• **Academia**
  – **IIT**: Xian-He Sun, Zhiling Lan, Shlomo Argamon
  – **UChicago**: Ian Foster, Tanu Malik, Zhao Zhang, Kyle Chard
  – **UEST China**: Yong Zhao
  – **SUNY**: Tevfik Kosar
  – **WSU**: Shiyong Lu
  – **USC**: Yogesh Simmhan
  – **Georgia Tech**: Jeffrey Vetter
  – **Columbia**: Glen Hocky
Active Funding ($)

- **NSF CAREER 2011 – 2015: $486K**
  - “Avoiding Achilles’ Heel in Exascale Computing with Distributed File Systems”, NSF CAREER

- **DOE Fermi 2011 – 2013: $84K**
  - “Networking and Distributed Systems in High-Energy Physics”, DOE FNAL

- **DOE LANL 2013: $75K**
  - “Investigation of Distributed Systems for HPC System Services”, DOE LANL

- **IIT STARR 2013: $15K**
  - “Towards the Support for Many-Task Computing on Many-Core Computing Platforms”, IIT STARR Fellowship

- **Amazon 2011 - 2013: $18K**
  - “Distributed Systems Research on the Amazon Cloud Infrastructure”, Amazon

- **NVIDIA 2013 – 2014: $12K**
  - “CUDA Teaching Center”, NVIDIA
Funding (Time)

- **DOE 2011 – 2013: 450K hours**
  - “FusionFS: Distributed File Systems for Exascale Computing”, DOE ANL ALCF; 450,000 hours on the IBM BlueGene/P

- **XSEDE 2013: 200K hours**
  - “Many-Task Computing with Many-Core Accelerators on XSEDE”, NSF XSEDE; 200K hours on XSEDE

- **GLCPC 2013: 6M hours**
  - “Implicitly-parallel functional dataflow for productive hybrid programming on Blue Waters”, Great Lakes Consortium for Petascale Computation (GLCPC); 6M hours on the Blue Waters Supercomputer

- **NICS 2013: 320K hours**
  - “Many-Task Computing with Many-Core Accelerators on Beacon”, National Institute for Computational Sciences (NICS); 320K hours on the Beacon system
Service Activities

- IEEE Transactions on Cloud Computing
  - Special Issue on Scientific Cloud Computing
- Springer’s Journal of Cloud Computing: Advances, Systems and Applications
- IEEE/ACM MTAGS 2013 @ SC13
- IEEE/ACM DataCloud 2013 @ SC13
- ACM ScienceCloud 2014 @ HPDC14
- IEEE CCGrid 2014 in Chicago
- GCASR 2014 in Chicago
- Others:
More Information

• More information:

• Contact:
  – [iraicu@cs.iit.edu](mailto:iraicu@cs.iit.edu)

• Questions?