Panel: Many-Task Computing meets Big Data

Chairs
Ioan Raicu, Illinois Institute of Technology & Argonne National Laboratory
Justin Wozniak, Argonne National Laboratory
Ian Foster, University of Chicago & Argonne National Laboratory
Yong Zhao, University of Electronic Science and Technology of China, China

ACM MTAGS 2013
November 17th, 2013
Panelists

- **Dr. Robert Grossman**
  - Professor and Director, Division of Biological Sciences and Computation Institute, University of Chicago

- **Dr. Xian-He Sun**
  - Chair and Professor, Computer Science, Illinois Institute of Technology

- **Dr. Judy Qiu**
  - Assistant Professor, Computer Science and Informatics, Indiana University

- **Dr. Alexandru Iosup**
  - Assistant Professor, Faculty of Engineering, Mathematics and Science, Delft University of Technology, the Netherlands

MTAGS13: Panel – Many-Task Computing meets Big Data
- We want to compute genomic variants.
- How can this be done as a distributed computation over science clouds?
- What are the APIs?
- What are the key common services?
- What is the governance structure?
- What is the sustainability model?
Big Data require both HPC and HTC, that is MTC, and is mixed compute-intensive and data-intensive components.

1. Human provides information to the machine.
2. Machine learns the appropriate function.
3. Human provides raw data, machine outputs answer, human consumes the information.

MTAGS13: Panel -- Many-Trask Computing meets Big Data
Support Both HPC and HTC: Ours Solution

Decoupled-Execution Paradigm:
- Handle computation- and data- intensive phases separately
- One interface-Two systems, transparent to users
- Integration, scheduling, optimization

Supercomputer or many-core computing system for execution of computing intensive part of an application

Data cloud or storage cluster for execution of data intensive part of an application

High speed network

Interoperability between different file systems

- Enable MPI Apps to access data-intensive file systems
- HPC-Cloud, Data-Cloud

Scalability of MTC: Memory-Parallelisms

- Multi-core
- Multi-threading
- Multi-issue

- Multi-banked Cache
- Non-blocking Cache
- Multi-level Cache

- Out-of-order Execution
- Speculative Execution
- Runahead Execution

- Pipelined Cache
- Data Prefetching
- Write buffer

- Multi-channel
- Multi-rank
- Multi-bank

Input-Output (I/O)

Parallel File System

Disks

MTAGS13: Panel -- Many-Trask Computing meets Big Data
The traditional AMAT:

\[ \text{HitCycle} + \text{MR} \times \text{AMP}. \]

MR is the miss rate of cache accesses; and AMP is the average miss penalty

The Concurrent AMAT:

\[ \frac{\text{HitCycle}}{C_H} + \frac{\text{MR} \times \text{AMP}}{C_M} \]

\( C_H \) is the hit concurrency; \( C_M \) is the pure miss concurrency

Hit is always good, miss may not be necessary bad

Design Choice of memory systems

## Domain of MapReduce and Iterative Extensions

<table>
<thead>
<tr>
<th>(a) Map Only (Pleasingly Parallel)</th>
<th>(b) Classic MapReduce</th>
<th>(c) Iterative MapReduce</th>
<th>(d) Loosely Synchronous</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
</tr>
</tbody>
</table>

**Collective Patterns**

<table>
<thead>
<tr>
<th>MapReduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Wordcount, Grep</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MapReduce-MergeBroadcast</th>
</tr>
</thead>
<tbody>
<tr>
<td>- KMeansClustering, PageRank</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Map-AllGather</th>
</tr>
</thead>
<tbody>
<tr>
<td>- MDS-BCCalc</td>
</tr>
<tr>
<td>- Matrix Mult</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Map-AllReduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>- KMeansClustering, MDS-StressCalc</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Map-ReduceScatter</th>
</tr>
</thead>
<tbody>
<tr>
<td>- PageRank, Belief Propagation</td>
</tr>
</tbody>
</table>

**Applications & Different Interconnection Patterns**

- **MapReduce**
  - Wordcount, Grep
- **MapReduce-MergeBroadcast**
  - KMeansClustering, PageRank
- **Map-AllGather**
  - MDS-BCCalc
  - Matrix Mult
- **Map-AllReduce**
  - KMeansClustering, MDS-StressCalc
- **Map-ReduceScatter**
  - PageRank, Belief Propagation

**Collective Communication**

**No Communication**

**MPI**
Alex Iosup
Thanks from the PDS Group at TU Delft.

Alexandru Iosup
Grids/Clouds, P2P systems, Big Data, Online gaming, Gamification

Dick Epema
Grids/Clouds, P2P systems, Video-on-demand, e-Science

Ana Lucia Varbanescu
HPC systems, Multi-cores, Big Data, e-Science

Henk Sips
HPC systems, Multi-cores, P2P systems

Johan Pouwelse
P2P systems, File-sharing, Video-on-demand

Home page
- www.pds.ewi.tudelft.nl

Publications
- see PDS publication database at publications.st.ewi.tudelft.nl
Applications from two worlds

- E-Science (incl. Big Data)
- Massively Multiplayer/Social Online Gaming (incl. Big Data)

10-years research in distributed systems

- System design, development, and evaluation
- Grid->Cloud computing, P2P->? Computing
- Performance measurements, evaluation, modeling, b’marking
- Grenchmark, Koala, Tribler, The Archives, [OpenTTD@large]

10 operational years research in comp. sci.


http://www.pdx.cwi.tudelft.nl/~iosup/
1. In the future, will Small-and-Medium Enterprises use elastic infrastructure running multiple frameworks?
   - Many-Task Big-Data Processing on Clouds—GPUs

2. In the future, should we risk working on scheduling policies?
   - Portfolio Scheduling

3. In the future, what is the role of job throughput, next to task throughput and peak performance (HPC)?

4. In the future, will social awareness be at the core of our shared distributed systems?

5. In the future, will it be possible to rate and rank distributed computing systems (benchmarking, also commercial issue)?
1. How do you see MTC intersecting with MapReduce, HTC, and HPC?

2. Importance of data locality for Big Data ==> how important is data-aware scheduling for Many-Task Computing

3. Supercomputers are designed for HPC applications today; in the future, should they be designed to support both MTC and/or Big Data?

4. With the growing scale of systems, has a centralized MTC system become obsolete? Is distributed MTC management (both scheduling and storage) a necessary next step?
1. How do you see MTC intersecting with MapReduce, HTC, and HPC?

2. Importance of data locality for Big Data ==> how important is data-aware scheduling for Many-Task Computing

3. Supercomputers are designed for HPC applications today; in the future, should they be designed to support both MTC and/or Big Data?

4. With the growing scale of systems, has a centralized MTC system become obsolete? Is distributed MTC management (both scheduling and storage) a necessary next step?
1. How do you see MTC intersecting with MapReduce, HTC, and HPC?

2. Importance of data locality for Big Data ==> how important is data-aware scheduling for Many-Task Computing

3. Supercomputers are designed for HPC applications today; in the future, should they be designed to support both MTC and/or Big Data?

4. With the growing scale of systems, has a centralized MTC system become obsolete? Is distributed MTC management (both scheduling and storage) a necessary next step?
1. How do you see MTC intersecting with MapReduce, HTC, and HPC?

2. Importance of data locality for Big Data ==> how important is data-aware scheduling for Many-Task Computing

3. Supercomputers are designed for HPC applications today; in the future, should they be designed to support both MTC and/or Big Data?

4. With the growing scale of systems, has a centralized MTC system become obsolete? Is distributed MTC management (both scheduling and storage) a necessary next step?
1. How do you see MTC intersecting with MapReduce, HTC, and HPC?

2. Importance of data locality for Big Data ==> how important is data-aware scheduling for Many-Task Computing

3. Supercomputers are designed for HPC applications today; in the future, should they be designed to support both MTC and/or Big Data?

4. With the growing scale of systems, has a centralized MTC system become obsolete? Is distributed MTC management (both scheduling and storage) a necessary next step?
MTAGS 2013 Website:
  – http://datasys.cs.iit.edu/events/MTAGS13/

Panel info:
  – http://datasys.cs.iit.edu/events/MTAGS13/panel.html

Workshop program (7 exciting talks in the PM)
  – http://datasys.cs.iit.edu/events/MTAGS13/program.html

Prize giveaway (win a Google Nexus 7):
  – http://datasys.cs.iit.edu/events/MTAGS13/prize.html

Contact
  – iraicu@cs.iit.edu