News on Scheduling Research in Delft (Holland) and Eindhoven (the Netherlands)

Dick Epema

with Bogdan Ghit, Alex Iosup, and Alexy Ilyushkin (TUD) and Aleksandra Kuzmanovska (TU/e)

Parallel and Distributed Systems Group
Delft University of Technology
Delft, the Netherlands

and

System Architecture and Networking Group
Eindhoven University of Technology
Eindhoven, the Netherlands

12 March 2015
Delft – the Netherlands – Europe

Holland (2/12 provinces)
The KOALA multicluster scheduler

- KOALA is our research vehicle for scheduling research
- deployed since 2005
- written in Java
- KOALA is transparent to the LRMs

parallel MPI applications
workflows
cycle-scavenging applications

MapReduce frameworks

Koala-C

LRM
LRM
LRM
LRM

information service
Scheduling frameworks

- **Reduce**
  - scheduling overhead of centralized scheduler
  - complexity of centralized scheduler

- **Provide isolation among frameworks**

- **KOALA**
  - requests large chunk of a cluster and
  - allocates dynamic parts of it to frameworks

- **Two models:**

  ![Diagram of scheduling frameworks]

  - optimal sizing
  - balancing
Types of Isolation

- Performance isolation
- Data isolation
- Failure isolation
- Version isolation
Resizing MapReduce: no data locality

Core nodes

- Classical deployment
- Uniform data distribution
- No removal

INPUT/OUTPUT DATA

Transient nodes (TR)

- No local storage
- R/W from/to core nodes
- Instant removal

12 March 2015
Resizing MapReduce: relaxed data locality

Core nodes

- Classical deployment
- Uniform data distribution
- No removal

INPUT/OUTPUT DATA

Trans-core nodes (TC)

- Local storage, no input
- Only R from core nodes
- Delayed removal

OUTPUT DATA
Performance of no versus relaxed data locality

MapReduce workflow for 15-TB 4-year BitTorrent trace analysis

- single-application performance decrease
- base line: 20 nodes with full HDFS deployment
- 10 core nodes + 10 transient/transient-core nodes

Balancing Allocations with FAWKES

**FAWKES in a nutshell**

1. Updates dynamic weights when:
   - new frameworks arrive
   - framework states change

2. Shrinks and grows frameworks to:
   - allocate new frameworks (min. shares)
   - give fair shares to existing ones

\[ w > w_{\text{min}} \]

12 March 2015
How to differentiate frameworks? (1/3)

By demand – 3 policies:
- Job Demand (JD)
- Data Demand (DD)
- Task Demand (TD)
How to differentiate frameworks? (2/3)

By usage – 3 policies:
- Processor Usage (PU)
- Disk Usage (DU)
- Resource Usage (RU)
By service – 3 policies:
- Job Slowdown (JS)
- Job Throughput (JT)
- Task Throughput (TT)
Our experimental testbed: DAS-4

- VU (148 CPUs)
- TU Delft (64)
- Astron (46)
- Leiden (32)
- UvA (32)
- UvA/MultimediaN (72)

10 Gb/s lambdas

SURFnet6

DAS5 on the way
- Q2 2015
- 400 8-core CPUs
- FDR Infiniband
Performance of FAWKES

<table>
<thead>
<tr>
<th>Nodes</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frameworks</td>
<td>3</td>
</tr>
<tr>
<td>Minimum shares</td>
<td>10</td>
</tr>
<tr>
<td>Datasets</td>
<td>300 GB</td>
</tr>
<tr>
<td>Jobs submitted</td>
<td>900</td>
</tr>
</tbody>
</table>

Up to 20% lower slowdown

Policy

None – Minimum shares
EQ – Equal shares
TD – Task Demand
PU – Processor Usage
JS – Job Slowdown

12 March 2015
Optimal sizing (1)

- **Fluent** is a component-based framework
  - jobs consist of **batches of identical video applications** with identical runtimes
  - **admission control**: jobs require immediate/fast start
  - metric: **reject rate** (of all applications across all jobs)

- **OnDemand** policy:
  - framework **initiative**
  - explicit grow and shrink requests to KOALA
  - **grow** because of new job that doesn’t fit
  - **shrink** after some idle time of resources

- **Proactive** policy:
  - KOALA **initiative**
  - **maintain utilization** (used/allocated) between lower and upper bound (periodic check)
Optimal sizing (3)

<table>
<thead>
<tr>
<th>policy</th>
<th>reject rate (%)</th>
<th>utilization (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td>13</td>
<td>46</td>
</tr>
<tr>
<td>on-demand</td>
<td>13</td>
<td>73</td>
</tr>
<tr>
<td>pro-active</td>
<td>21</td>
<td>65</td>
</tr>
</tbody>
</table>
Workflow scheduling (1)

real workloads

workflows

most workflow scheduling

our workflow scheduling

parallel applications

bag-of-tasks

Workload scheduling (2)

• **Research question**
  o how to schedule *workloads of workflows* with *unknown* task runtimes?

• **Reserving some processors for job(s) at the head of the queue**
  o *reduces* time in service
  o but *increases* wait time
  o is clearly not good at very high utilizations

• **Policies**
  o strict reservation (reserve for *maximum Level of Parallelism*)
  o scaled LoP (reserve only for fraction of max. LoP)
  o future eligible sets (look number of steps into the future)
  o (unrestricted) backfilling

• **Metric**
  o job slowdown

---

12 March 2015
Workload scheduling (3)

- Strict Reservation
- Future Eligible Sets, depth 2
- Scaled LoP ($f = 0.2$)
- Backfilling
Portfolio scheduling

- Create a set of scheduling policies
  - resource provisioning and allocation policies
- Online selection of the active policy, at important moments
  - periodic selection
  - change in pricing model
  - change in datacenter architecture

Next March in Delft

Welcome to the 7th ACM/SPEC International Conference on Performance Engineering

The International Conference on Performance Engineering (ICPE) provides a forum for the integration of theory and practice in the field of performance engineering. ICPE is an annual joint meeting that has grown out of the ACM Workshop on Software Performance (WOSP) and the SPEC International Performance Engineering Workshop (SIPEW). It brings together researchers and industry practitioners to share ideas, discuss challenges, and present results of both work-in-progress and state-of-the-art research on performance engineering of software and systems.

General Chair: Alex Iosup
More information

• Publications
  o see PDS publication database at publications.st.ewi.tudelft.nl

• Home pages:
  o www.pds.ewi.tudelft.nl/epema
  o www.pds.ewi.tudelft.nl/~iosup

• Web sites:
  o KOALA: www.st.ewi.tudelft.nl/koala
  o DAS4: www.cs.vu.nl/das4
Our research tag cloud