DynamicCloudSim: Simulating Heterogeneity in Computational Clouds

Marc Bux, Ulf Leser
{bux|leser}@informatik.hu-berlin.de

The 2nd international workshop on Scalable Workflow Enactment Engines and Technologies (SWEET'13)
Meet Sandra
Meet Sandra

DynamicCloudSim: Simulating Heterogeneity in Computational Clouds
Meet Sandra
Meet Paul

- **Small Instance**: 1.7 GB RAM, 1 EC2 Compute Unit, 160 GB local storage
- **Compute Unit**: equiv. CPU capacity of a 1.0-1.2 GHz Opteron or Xeon
- No guarantees wrt. I/O throughput and network delay / bandwidth
Meet Paul

Any one cloud instance is unlike another.
Heterogeneity in EC2 Cloud Instances

- **Different CPUs** on physical host systems [Jackson10, Schad10]
  - Intel Xeon E5430 (2.66 GHz quad)
  - AMD Opteron 270 (2 GHz dual)
  - AMD Opteron 2218 HE (2.6 GHz dual)

- **I/O throughput** varies as well [Dejun10]
  - No correlation between CPU and I/O performance

Source: [Dejun10]
Dynamic Changes of Performance

- Occasional CPU performance slumps and failures during task execution [Dejun10, Jackson10]
- Variance in I/O and network throughput [Zaharia08, Jackson10]
- Performance depends on hour of day and day of week [Schad10]

EC2 Disk performance vs. VM co-allocation [Zaharia08]

CPU performance slumps [Dejun10]
Vision

Adaptive scheduling of scientific workflows

• Exploit heterogeneous resources
• Exhibit robustness to instability
The standard approach for evaluation is **simulation** [Braun01, Blythe05]

- **Cloud simulation toolkits** do not model instability
Agenda

1) Simulating Heterogeneity in Computational Clouds
2) Evaluating Established Workflow Schedulers
3) Summary and Outlook
Agenda

1) Simulating Heterogeneity in Computational Clouds
2) Evaluating Established Workflow Schedulers
3) Summary and Outlook
CloudSim


- More than 250 citations in Google Scholar

- [https://code.google.com/p/cloudsim/](https://code.google.com/p/cloudsim/)
DynamicCloudSim

- Extend CloudSim with models for
  1. Heterogeneous computational resources (Het)
  2. Dynamic changes of performance at runtime (DCR)
  3. Straggler VMs and failed task executions (SaF)
- More fine-grained representation of computational resources
- [https://code.google.com/p/dynamiccloudsim/](https://code.google.com/p/dynamiccloudsim/)
Realism – can we ever get there?

- Simulation can never perfectly resemble reality
- We model inhomogeneity and dynamic changes by sampling from normal distributions
- Default mean and STD/RSD Parameters are obtained from [Zaharia08, Dejun10, Jackson10, Schad10, Iosup11]


Many performance characteristics in EC2 follow a normal distribution [Schad10]
Simulating VM Performance: DCS vs CS

1. Heterogeneous computational resources (\textbf{Het})
2. Dynamic changes of performance at runtime (\textbf{DCR})
3. Straggler VMs and failed task executions (\textbf{SaF})
1) Simulating Heterogeneity in Computational Clouds

2) Evaluating Established Workflow Schedulers
   a) Scheduling Scientific Workflows
   b) Evaluation Workflows
   c) Evaluation Results

3) Summary and Outlook
Agenda

1) Simulating Heterogeneity in Computational Clouds

2) Evaluating Established Workflow Schedulers
   a) Scheduling Scientific Workflows
   b) Evaluation Workflows
   c) Evaluation Results

3) Summary and Outlook
Scheduling of Scientific Workflows

• **Scheduling:**
  – Mapping tasks to the available physical resources
  – Usual goal: minimize overall execution time

• **Static Scheduling:**
  – Schedule is assembled prior to workflow execution
  – Schedule is strictly abided at runtime

• **Adaptive Scheduling:**
  – Monitor computational infrastructure
  – Adjust workflow execution at runtime
Static Schedulers

• Baseline: **Round Robin**
  – Assign tasks to resources in turn
  – Equal amount of tasks per resource

• Elaborate: **HEFT** (Het. Earliest Finish Time) \[\text{[Topcuoglu02]}\]
  – Implemented in SWfMS **Pegasus**
  – Requires *runtime estimates* for each task on each resource
  – Assign tasks with longest time to finish a fixed timeslot on a suitable (well-performing) resource
  – Exploit heterogeneity in computational infrastructure (**Het**)
Adaptive Schedulers

• Baseline: Greedy Task Queue
  – Assign tasks to resources at runtime in first-come-first-served manner
  – Adapts to changes of performance at runtime (DCR)

• Elaborate: LATE (Longest Approx. Time to End) [Zaharia08]
  – Developed for Hadoop to increase robustness to instability
  – 10% of Tasks progressing at rate below average are replicated and speculatively executed
  – Exploit dynamic changes of performance
  – Robust to straggler VMs and failed task executions (SaF)
Agenda

1) Simulating Heterogeneity in Computational Clouds

2) Evaluating Established Workflow Schedulers
   a) Scheduling Scientific Workflows
   b) Evaluation Workflows
   c) Evaluation Results

3) Summary and Outlook
Evaluation Workflow: Montage [Berriman04]
Abstract Montage Workflow

One task can have many task instances.
Concrete Montage Workflow

- **43,318 tasks** reading and writing **534 GB of data**
- **10 GB** input files which have to be uploaded to the cloud
- Determine avg. runtime over **100 simulations** of workflow exec.
Eval. Workflow: Comparative Genomics

New Data (not in the paper)
Concrete Genomics Workflow
Concrete Genomics Workflow

- Align 10% of the reads produced in a sequencing experiment against the smallest of human chromosomes (chr22)
  - Use about 0.2% of the available data

- 4,266 tasks reading and writing 436 GB of data (2.3 GB upload)
Agenda

1) Simulating Heterogeneity in Computational Clouds

2) Evaluating Established Workflow Schedulers
   a) Scheduling Scientific Workflows
   b) Evaluation Workflows
   c) Evaluation Results

3) Summary and Outlook
Runtime depending on Heterogeneity (Het)
Runtime depending on Dynamic Changes (DCR)
Runtime with Stragglers and Failures (SaF)

Average Runtime in Minutes

<table>
<thead>
<tr>
<th>Method</th>
<th>Static Round Robin</th>
<th>HEFT</th>
<th>Greedy Queue</th>
<th>LATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likelihood</td>
<td>0.00625</td>
<td>0.0125</td>
<td>0.01875</td>
<td>0.025</td>
</tr>
<tr>
<td>Average</td>
<td>368</td>
<td>598</td>
<td>405</td>
<td>321</td>
</tr>
</tbody>
</table>

Likelihood of Straggler VMs and Failed Tasks (SaF)

Average Runtime in Minutes

<table>
<thead>
<tr>
<th>Method</th>
<th>Static Round Robin</th>
<th>HEFT</th>
<th>Greedy Queue</th>
<th>LATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likelihood</td>
<td>0.00625</td>
<td>0.0125</td>
<td>0.01875</td>
<td>0.025</td>
</tr>
<tr>
<td>Average</td>
<td>203</td>
<td>352</td>
<td>262</td>
<td>187</td>
</tr>
</tbody>
</table>
That’s all well and good, but...

• Scheduling in SWfMS: Static or Greedy Task Queue

• HEFT and LATE have a computational overhead and require information not available in real scenarios:
  – HEFT: runtime estimates of each task on each machine
  – LATE: progress rate of each running task

• Untapped optimization potential:
  multiple resource scheduling
  – Find appropriate matches between tasks and machines
Summary and Outlook

• EC2: Heterogeneity and instability in VM performance

• **DynamicCloudSim** introduces several factors of instability into CloudSim

• Simulation experiments reproduce known strengths and shortcomings of established schedulers

• Outlook: Comparative evaluation on real hardware
Thanks for your attention!

https://code.google.com/p/dynamiccloudsim/
Questions
Literature


Literature (cont.)


Literature (cont.)

