KOALA-C: A Task Allocator for Integrated Multicluster and Multicloud Environments

Presenter: Lipu Fei
Authors: Lipu Fei, Bogdan Ghiț, Alexandru Iosup, Dick Epema
Institute: TU Delft, the Netherlands
A Multicluster and Multicloud Environment

- Heterogeneous resources
- Efficient resource management
A Multicluster and Multicloud Environment

- Increasing number of short jobs[1][2]

- Goal: optimize global job slowdown by optimizing short jobs

Introduction

Contribution

- KOALA-C scheduler for integrated multicloud and multicloud environments.
- Two new scheduling policies for the new environment.
- A comprehensive experimental evaluation of our architecture and policies.
KOALA: A Co-allocating Grid Scheduler

- **Original goals:**
  1. **processor co-allocation:** parallel applications
  2. **data co-allocation:** job affinity based on data locations
  3. **load sharing:** in the absence of co-allocation

- **Additional goals:**
  - research vehicle for grid and cloud research
  - support for (other) popular application types (e.g. MapReduce)

- **KOALA:**
  - is written in Java
  - is middleware independent (initially Globus-based)
  - **has been deployed** on the DAS2 – DAS4 since Sept. 2004
DAS-4: The Distributed ASCI Supercomputer 4

- System purely for CS research
- OpenNebula installed
- Operational since Oct. 2010
- Institutions and organizations:
  - VU University, Amsterdam (VU)
  - Leiden University (LU)
  - University of Amsterdam (UvA)
  - Delft University of Technology (TUD)
  - The MultimediaN Consortium (UvA-MN)
  - Netherlands Institute for Radio Astronomy (ASTRON)
System Model

- Global and Local resource managers (GRM and LRM).
System Model

- Global and Local resource managers.
System Model

- Global and Local resource managers.
- Compute-intensive jobs, sequential or parallel high-performance applications.
- Rigid jobs that do not change size.
System Model

- Global and Local resource managers.
- Compute-intensive jobs, sequential or parallel high-performance applications.
- Rigid jobs that do not change size.
- Preemptive jobs (restart instead of resume).
Outline

1. A Multicluster and Multicloud Environment
2. KOALA-C: a System for MC + MC Environments
3. Design for Policies without prediction
4. Simulation and Real-World Experiments
5. Conclusion
KOALA-C

User

Job Description File (JDF)

(1) Submit Job

KOALA-C Scheduler

Job Queue
KOALA-C

KOALA-C Scheduler

Policy Framework

Allocation Policy

Provisioning Policy

Job Runtime Predictor

Job Queue

User

(1) Submit Job

Job Description File (JDF)

Resource Pool

(2) Allocate Jobs

Cluster Site 1

Cluster Site 2

Free Sites

Cloud Site 1

Cloud Site 2

Commercial Sites

VMs

Challenge the future
KOALA-C

KOALA-C Scheduler

Job Queue

Policy Framework

Allocation Policy

Provisioning Policy

Job Runtime Predictor

User

(1) Submit Job

Job Description File (JDF)

Resource Pool

Cluster Site 1

Cluster Site 2

Free Sites

Cloud Site 1

Cloud Site 2

Commercial Sites

(2) Allocate Jobs

Lease/Release VMs

VMs
KOALA-C

KOALA-C Scheduler

Policy Framework
- Allocation Policy
- Provisioning Policy
- Job Runtime Predictor

Job Queue

Resource Pool

(1) Submit Job

User

Job Description File (JDF)

(2) Allocate Jobs

(3) Download Results

Result Files

Cluster Site 1
Cluster Site 2

Free Sites

Cloud Site 1
Cloud Site 2

Commercial Sites

Lease/Release VMs

VMs
KOALA-C

KOALA-C Scheduler

Policy Framework
Allocation Policy
Provisioning Policy
Job Runtime Predictor

Job Queue

(1) Submit Job
User

Job Description File (JDF)

(3) Download Results
Result Files

(2) Allocate Jobs
Resource Pool

Cluster Site 1
Cluster Site 2

Free Sites

(4) Record Information
KOALA-C Database

(2) Allocate Jobs
Resource Pool

VMs

Cloud Site 1
Cloud Site 2

Commercial Sites

Logical Partitioning

Challenge the future
Outline

1. A Multicluster and Multicloud Environment
2. KOALA-C: a System for MC + MC Environments
3. **Design for Policies without prediction**
4. Simulation and Real-World Experiments
5. Conclusion
We propose TAGS-chain and TAGS-sets, which are new in this work.
Traditional TAGS Policy

- Task Assignment by Guessing Size (TAGS)[3]:
  - use no prior information and use no prediction
  - jobs can be aborted
  - optimize short jobs

TAGS-based Policy Design

- **Goal:** to achieve low slowdown *without prediction*
- **Method:** to partition the sites into sets to serve jobs of different runtime ranges

- A number of sets of sites
- Set $i$ allows jobs to run for $T_i$ amount of time ($T_i < T_{i+1}$)
- The last set has a $T$ of $\infty$ (all jobs will finish without being killed)

\begin{align*}
T_1 < T_2 < T_3 \\
T_3 = \infty
\end{align*}
TAGS-based Policy Design

1st Start

T₁

On a site in Set 1

Actual time required for the job to finish
Policy Design

TAGS-based Policies in General

2nd Start

On a site in Set 2

T2

1st Start

On a site in Set 1

T1

Killed (1st time)

Actual time required for the job to finish
Policy Design

TAGS-based Policies in General

$3^{rd}$ Start

On a site in Set 3

$2^{nd}$ Start

On a site in Set 2

$1^{st}$ Start

On a site in Set 1

$T_1$

$T_2$

$T_3$

Finished

Killed (2$^{nd}$ time)

Killed (1$^{st}$ time)

Actual time required for the job to finish
Policy Design

TAGS-chain and TAGS-sets

TAGS-chain:

Arriving Jobs

Set 1

T_1

Site 1

Set 2

T_2

Site 2

Set 3

T_3

Site 3

TAGS-sets:

Arriving Jobs

Set 1

T_1

Site 1

Site 2

Site 3

Set 2

T_2

Site 2

Site 3

Set 3

T_3

Site 3
Outline

- A Multicluster and Multicloud Environment
- KOALA-C: a System for MC + MC Environments
- Design for Policies without Prediction

**Simulation and Real-World Experiments**
- Simulation Results
- Real-World Experiment Results

- Conclusion
Simulation Results

Experimental Setup

- **Resources:**
  - 3 cluster sites (32 nodes each)
  - 1 private cloud site (≤64 VMs) and 1 public cloud site (≤128 VMs)
  - 6 policies: 5 baselines and TAGS-sets

- **Workload:**
  - 8 traces
  - job size limited to 32 nodes

- **6 Policies and 2 Metrics:**
  - FF, RR, SJF, SJF-ideal, and HSDF as baselines + TAGS-sets
  - slowdown, total cost

- *Simulation results agree with real-world experiment results*

- *We have numerous experimental results in the paper*
Simulation Results

Configuration of TAGS-sets

Short Job Set
ALL sites

Long Job Set
ALL without Cluster 1 and 2

![Graph showing simulation results with logarithmic scale](graph.png)
Simulation Results

Configuration of TAGS-sets

More effect on slowdown

Short Job Set
ALL sites

Long Job Set
ALL without Cluster 1 and 2

![Graph showing simulation results with axes labeled as follows:
- Y-axis: Average Job Slowdown
- X-axis: Runtime Limit for Short Jobs [min.]
- Graph lines for Slowdown and Cost]

TU Delft
Challenge the future
Simulation Results

Configuration of TAGS-sets

More effect on slowdown

No common optimal runtime limit, but 40min seems to be suitable for most traces.
Simulation Results

Policy Evaluation

Average Job Slowdown

KTH-SP2  CTC-SP2  SDSC-BLUE  SDSC-SP2  DAS2-FS3  AUVERGRID  LCG  NORDUGRID
Simulation Results

Policy Evaluation

SJF-ideal has lowest slowdowns, but it cannot be achieved in reality.
Simulation Results

Policy Evaluation

TAGS-sets has the lowest slowdowns overall
Simulation Results

Policy Evaluation

>10x improvement over SJF

>25x improvement over SJF
Simulation Results

Policy Evaluation

**TAGS-sets** can achieve good performance in multicloud and multicloud environments.
Simulation Results

Policy Evaluation

![Graph showing simulation results for different policies and algorithms. The graph is on a log scale, with total cost in dollars on the y-axis and various policies and algorithms on the x-axis.]
Simulation Results

Policy Evaluation

TAGS-sets incurs 1.3 to 4.2x higher costs than the others
Simulation Results

Policy Evaluation

TAGS-sets can achieve a good performance-cost tradeoff in an integrated multicloud environment.
Real-World Experiment Results

Experimental Setup

- **Resources:**
  - 2 cluster sites of the DAS-4 system (32 nodes each)
  - OpenNebula-based private cloud of DAS-4 (up to 32 VMs)
  - Amazon EC2 as public cloud (up to 64 VMs)

- **Workload:**
  - A part of the CTC-SP2 workload (≈12 hours)
  - 70% average utilization on the system (with the max cloud size)

- **Policies:**
  - FF, SJF, and TAGS-sets
  - TAGS-sets setup:

Experimental Results

![Graph showing experimental results with categories: FF, SJF, TAGS-sets. The graph compares slowdown with Overall, Short Jobs (46%), and Long Jobs (54%).](image)
Experimental Results

- Lower and more balanced slowdowns

Overall
- Short Jobs (46%)
- Long Jobs (54%)

Slowdown
Experimental Results

- Overall
- Short Jobs (46%)
- Long Jobs (54%)

<table>
<thead>
<tr>
<th>Method</th>
<th>FF</th>
<th>SJF</th>
<th>TAGS-sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[s]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 - 5000</td>
<td>1000</td>
<td>500</td>
<td>3000</td>
</tr>
<tr>
<td>5001 - 10000</td>
<td>1500</td>
<td>100</td>
<td>5000</td>
</tr>
<tr>
<td>10001 - 15000</td>
<td>2000</td>
<td>150</td>
<td>7000</td>
</tr>
<tr>
<td>15001 - 20000</td>
<td>2500</td>
<td>200</td>
<td>9000</td>
</tr>
<tr>
<td>20001 - 25000</td>
<td>3000</td>
<td>250</td>
<td>11000</td>
</tr>
<tr>
<td>25001 - 30000</td>
<td>3500</td>
<td>300</td>
<td>13000</td>
</tr>
<tr>
<td>30001 - 35000</td>
<td>4000</td>
<td>350</td>
<td>15000</td>
</tr>
<tr>
<td>35001 - 40000</td>
<td>4500</td>
<td>400</td>
<td>17000</td>
</tr>
<tr>
<td>40001 - 45000</td>
<td>5000</td>
<td>450</td>
<td>19000</td>
</tr>
</tbody>
</table>
Experimental Results

Short jobs have shorter wait times
Long jobs have longer wait times

Overall
Short Jobs (46%)
Long Jobs (54%)

Wait Time [s]

FF  SJF  TAGS-sets
**Experimental Results**

TAGS-sets has better optimization for short jobs than the other policies.
Outline

- A Multicluster and Multicloud Environment
- KOALA-C: a System for MC + MC Environments
- Design for Policies without Prediction
- Simulation and Real-World Experiments
- Conclusion
Conclusion

• We designed KOALA-C, a task scheduler for integrated multicluster and multicloud environments.

• We designed two new scheduling policies without prediction for multicluster and multicloud environments.

• Through experiments, we show that two new policies can achieve significant improvement in performance.

• For the future, we plan to extend our policy set.
More Information

• **PDS group:**
  - Parallel and Distributed Systems Group @ TU Delft

• **Publications:**
  - see PDS publication database at:
    - www.pds.ewi.tudelft.nl/research-publications/publications

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

• **Web sites:**
  - KOALA:  www.st.ewi.tudelft.nl/koala
  - DAS-4:  www.cs.vu.nl/das4
  - GWA:    gwa.ewi.tudelft.nl
Thank You
Questions?