PriDynSim: A Simulator for Dynamic Priority Based I/O Scheduling for Cloud Applications

Nitisha Jain*, J. Lakshmi*
Nikolay Grozev‘, Rajkumar Buyya‘

*Indian Institute of Science, Bangalore, India
‘The University of Melbourne, Parkville, Australia

November 26, 2015

IEEE CCEM 2015, Bangalore

Outline

1. Overview
2. PriDynSim
3. Architecture
4. Implementation
5. Performance Evaluation
6. Conclusions

Nitisha Jain*, J. Lakshmi*
Nikolay Grozev‘, Rajkumar Buyya‘

1. Overview
2. PriDynSim
3. Architecture
4. Implementation
5. Performance Evaluation
6. Conclusions

- Cloud computing - popular model for renting computational resources as per need.
- On-demand availability, scalability of IT resources, lower costs.
- This flexibility is important for enterprises in emerging markets, provides competitive advantages.
- One of the major impediments for the research in the area -> lack of affordable testing environments.

Simulation tools

- Provide a good alternative to testing in large scale and expensive testbeds.
- Controlled environment for hypotheses evaluation.
- Allow emerging enterprises to quickly evaluate novel resource management techniques
- Fostering innovation, putting them at par with competitors.

Nitisha Jain*, J. Lakshmi*
Nikolay Grozev‘, Rajkumar Buyya‘

CloudSim Tookit

- Developed by Rajkumar Buyya et. al. at CLOUDS lab, *University of Melbourne*
- Most widely used tool for Cloud simulation.
- Provides flexible, customizable platform for modeling of Cloud data-centers, services, brokers, virtualized servers.

CloudSim Tookit

- Developed by Rajkumar Buyya et. al. at CLOUDS lab, *University of Melbourne*
- Most widely used tool for Cloud simulation.
- Provides flexible, customizable platform for modeling of Cloud data-centers, services, brokers, virtualized servers.

Limitation

Lacks support for disk I/O operations and policies for allocation of I/O resources to concurrent I/O workloads.

- Suitable I/O performance models extending the widely adopted CloudSim simulator can further research in I/O resource management.
- Need for an I/O resource scheduler, to assign resources to applications based on their specific characteristics and requirements.

PriDynSim (Dynamic Priority Simulator)

Generic *CloudSim* based simulator for priority based I/O resource scheduling.

1. Overview

2. PriDynSim

3. Architecture

4. Implementation

5. Performance Evaluation

6. Conclusions

PriDynSim Toolkit

- Facilitates evaluation of policies for dynamic I/O resource scheduling across co-located heterogeneous applications.
- Consider requirements of wide spectrum of I/O applications, guarantee performance QoS.
- Incorporates the representation of latencies, deadlines of applications in simulation environment
- Facilitate future research efforts in development of time-constrained scheduling policies.

Related Work

- *CloudSim* - most closely related to our work, but lacks support for representation of deadline driven I/O workloads, priority based scheduling policies.
- Extensions of *CloudSim* :
  - *WorkflowSim* - Support for workflow simulations and scheduling algorithms.
  - *DynamicCloudSim* - Extends *CloudSim* to handle heterogeneity of applications, dynamic changes to the performance.
  - CloudReports - Provides GUI for simulating techniques for power optimization in Cloud computing environments.

Related Work

- *CloudSim* - most closely related to our work, but lacks support for representation of deadline driven I/O workloads, priority based scheduling policies.

- Extensions of *CloudSim* :
  - *WorkflowSim* - Support for workflow simulations and scheduling algorithms.
  - *DynamicCloudSim* - Extends *CloudSim* to handle heterogeneity of applications, dynamic changes to the performance.
  - CloudReports - Provides GUI for simulating techniques for power optimization in Cloud computing environments.

None of them address performance modeling of concurrently executing latency sensitive I/O workloads, common in real-life Cloud setups.

- *GreenCloud*, *MDCSim* focus on energy consumption and power optimization for multi-tier Cloud data-centers, but no consideration to application performance.
- *ICanCloud* enables simulation on larger scale spanning multiple machines, but lacks any support for I/O operations or priority scheduling policies.

None of the available simulation platforms have attempted to design priority based I/O scheduling policies to address the performance concerns for I/O intensive workloads in a Cloud environment like *PriDynSim*.

Proposed Extensions

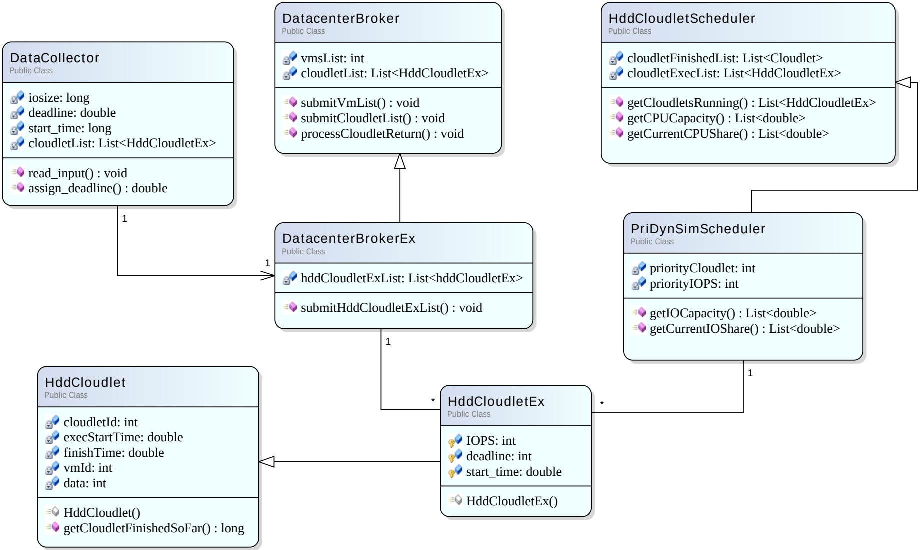
- *HddCloudletEx*
  - *Deadline* - Time by which the cloudlet is expected to finish completion.
  - *Start Time* - Time at which the cloudlet is assigned the disk IOPS for executing I/O operations.
  - *IOPS* - Value of disk IOPS assigned to the cloudlet at a given time.

Proposed Extensions

- *HddCloudletEx*
  - *Deadline* - Time by which the cloudlet is expected to finish completion.
  - *Start Time* - Time at which the cloudlet is assigned the disk IOPS for executing I/O operations.
  - *IOPS* - Value of disk IOPS assigned to the cloudlet at a given time.
- *PriDynSim Scheduler*
  - Replaces existing time shared fair scheduler.
  - Measures requirements of cloudlets as per their I/O operations, deadline values, and assigns IOPS to satisfy deadlines.

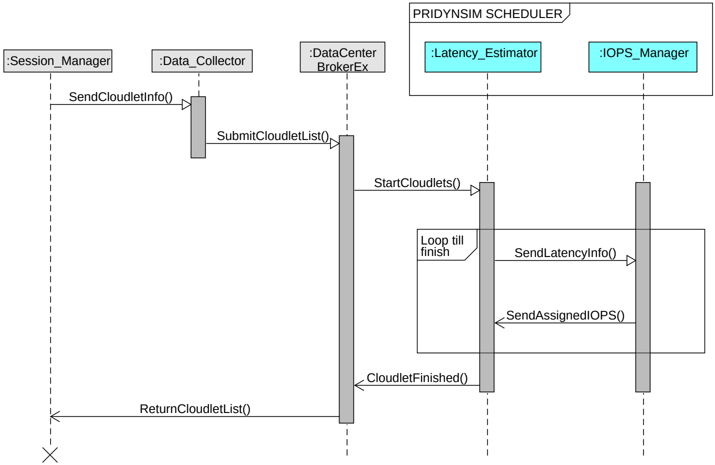
Proposed Extensions

- *HddCloudletEx*
  - *Deadline* - Time by which the cloudlet is expected to finish completion.
  - *Start Time* - Time at which the cloudlet is assigned the disk IOPS for executing I/O operations.
  - *IOPS* - Value of disk IOPS assigned to the cloudlet at a given time.
- *PriDynSim Scheduler*
  - Replaces existing time shared fair scheduler.
  - Measures requirements of cloudlets as per their I/O operations, deadline values, and assigns IOPS to satisfy deadlines.
- *DatacenterBrokerEx*
  - Acts on behalf of the user for creation and destruction of VMs and submission of cloudlets to the VMs.
  - Can handle the submission of *HddCloudletEx* entities with I/O operations, deadlines.



PriDynSim Class Diagram

Functionality



Interaction Diagram

1. Overview

2. PriDynSim

3. Architecture

4. Implementation

5. Performance Evaluation

6. Conclusions

PriDyn Algorithm Terminology

Attribute	Notation
Cloudlet	$C =< C_{(1)}, C_{(2)} \dots C_{(N)} >$
Deadline	$D =< D_{(1)}, D_{(2)} \dots D_{(N)} >$
Total Data Size	$R = [R_1, R_2 \dots R_N]$
Number of IO Operations	$IO =< IO_{(1)}, IO_{(2)} \dots IO_{(N)} >$
Start Time	$ST =< ST_{(1)}, ST_{(2)} ... ST_{(N)} >$
Disk IOPS	$IOPS =< IOPS_{(1)}, IOPS_{(2)} \dots IOPS_{(N)} >$
Latency	$L =< L_{(1)}, L_{(2)} \dots L_{(N)} >$
Priority Cloudlet	$C_{Priority}$
Priority IOPS	$IOPS_{Priority}$
Maximum IOPS	$IOPS_{max}$

Algorithm - Priority Scheduler I

Require: $N_{Total}, C, Total\ IO, D, ST$

Ensure: $IOPS$

Initialize $IOPS_{Priority}$ *as* 0
for *each* $C_{(i)}$ *in* $< C_{(1)}...C_{(N)} >$ **do**
    Call $IOPS\_Manager(N, IOPS_{Priority})$
end for
while $(\mathrm{N} >= 0)$ **do**
    for *each* $C_{(i)}$ *in* $< C_{(1)}...C_{(N)} >$ **do**
        Calculate $L_{(i)} = IOPS_{(i)}/IO_{(i)}$
    end for
    if $(exists\ C_{(i)}\ s.\ t.\ L_{(i)} > D_{(i)})$ **then**
        Find $C_{(i)}$ *where* $D_{(i)}$ *is minimum*
        if $((IO_{(i)}/D_{(i)}) - ST_{(i)}) > IOPS_{max}$ **then**
            Continue to next $C_{(i)}$
        else
            $C_{Priority} = C_{(i)}$

Algorithm - Priority Scheduler II

$IOPS_{Priority} = IO_{(i)}/(D_{(i)} - ST_{(i)})$
end if
end if
for *All* $C_{(i)}$ *in* $< C_{(1)} ... C_{(N)} >$ **do**
if $C_{(i)} = C_{Priority}$ **then**
$IOPS_{(i)} = IOPS_{Priority}$
else
$Call\ IOPS\_Manager(N, IOPS_{Priority})$
end if
end for
end while

Algorithm - IOPS Manager

Require: N, $IOPS_{Priority}$
Ensure: $IOPS_{(i)}$

if $(IOPS_{Priority} = 0)$ **then**
    $IOPS_{(i)} = IOPS_{max}/N$
else
    $IOPS_{(i)} = (IOPS_{max} - IOPS_{Priority})/(N-1)$
end if
return $IOPS_{(i)}$

Using Real I/O Workload Traces

- Real world I/O traces used to understand multi-tenanted Cloud setups.[1]
- Block I/O traces from servers at Microsoft Cambridge.
- Consist of diverse applications like web server, media server, research projects etc.
- Trace requests were modelled to represent I/O application stream running on a virtual machine on the server.
- Deadline values assigned to requests based on their characteristics and I/O size.

[1] http://www.iotta.snia.org/

Deadline Assignment for I/O Requests

- *Makespan* - Min time for completing I/O request.
- $Makespan = IO / (IOPS(max))$
- Delay tolerance parameter δ, represents functional characteristics of request.
- $Deadline = rand\ [Makespan, Makespan + (Makespan * \delta)]$

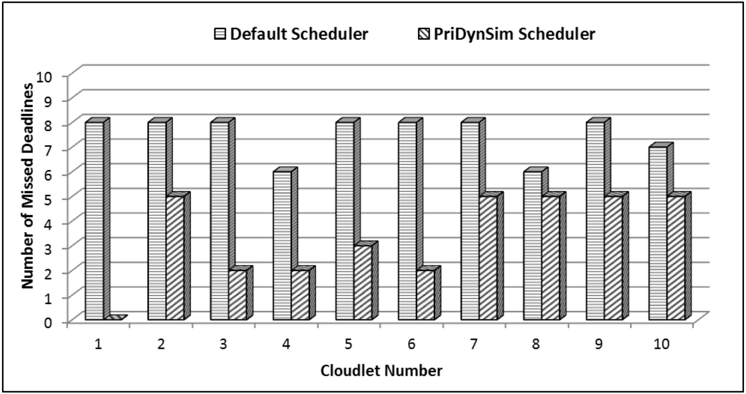
Nitisha Jain*, J. Lakshmi*
Nikolay Grozev‘, Rajkumar Buyya‘

Case Study

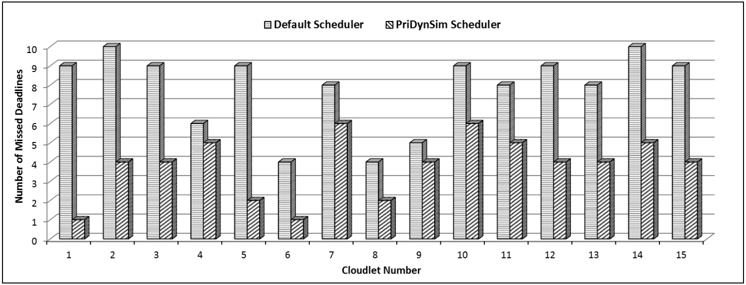
- Experimental evaluation with varying number, combinations of cloudlets.
- Set of cloudlets with equal number of latency sensitive, delay tolerant jobs modeled by the I/O requests belonging to Media server and Research server.

Case Study

- Experimental evaluation with varying number, combinations of cloudlets.
- Set of cloudlets with equal number of latency sensitive, delay tolerant jobs modeled by the I/O requests belonging to Media server and Research server.

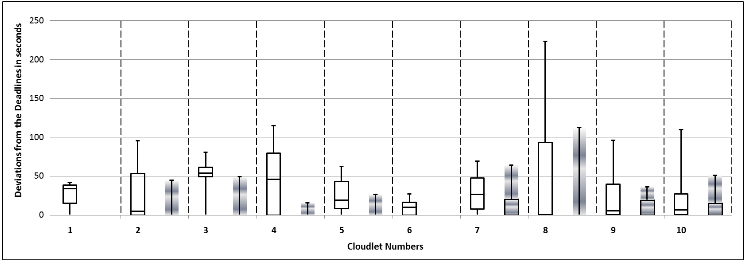


Comparison of Missed Deadlines for 10 Cloudlets



Comparison of Missed Deadlines for 15 Cloudlets

Measure of *deviations* of cloudlets from their deadlines, i.e. the difference between the values of response times (the time at which the cloudlet finished execution) and the value of deadline assigned to it.



Comparison of Deviations from Deadlines

1. Overview

2. PriDynSim

3. Architecture

4. Implementation

5. Performance Evaluation

6. Conclusions

PriDynSim Benefits

- PriDynSim -> Explore policies for dynamic allocation of disk resources to I/O bound applications.
- Gives prioritized disk access to latency-sensitive jobs.
- Guaranteed application performance for a wide variety of typical Cloud workloads modeled by real world I/O traces.
- Designed as a generic simulator -> can be used as a testbed by future researchers for evaluation of own specific scheduling policies.

Nitisha Jain*, J. Lakshmi*
Nikolay Grozev‘, Rajkumar Buyya‘

Future Work..

Extension to data-center level, allocation of cloudlets or job to a VM based on the application requirements -> optimize overall efficacy of the resource allocation for Cloud data-center.

Acknowledgements

We thank Rodrigo Calheiros, Amir Vahid Dastjerdi, Yaser Mansouri, and Chenhao Qu for their comments on improving this work. This work is partially supported by the Melbourne-Chindia Cloud Computing (MC3) Research Network.

Publications

1. Nitisha Jain, J. Lakshmi, "PriDyn : Enabling Differentiated I/O Services in Cloud using Dynamic Priorities", *IEEE Transactions on Services Computing (Special Issue on Cloud Computing)*, vol. PP, no. 99, 2014.

2. Nitisha Jain, J. Lakshmi, "PriDyn : Framework for Performance Specific QoS in Cloud Storage", *Proceedings of the 7th IEEE International Conference on Cloud Computing (IEEE CLOUD 2014)*, June 27 - July 2, 2014, Alaska, USA.

3. Nitisha Jain, J. Lakshmi, "PCOS : Prescient Cloud I/O Scheduler for Workload Consolidation and Performance", *Proceedings of the 6th International Conference on Cloud Computing and Big Data (CCBD 2015)*, November 4-6, 2015, Shanghai, China.

4. Nitisha Jain, Nikolay Grozev, Rajkumar Buyya, J. Lakshmi, "PriDynSim : A Simulator for Dynamic Priority Based I/O Scheduling", under review at the *3rd IEEE International Conference on Cloud Computing in Emerging Markets (CCEM 2015)*, November 25 - 27, 2015, Bangalore, India.

Thank You

Questions ?