Dynamic request allocation and scheduling for context aware applications subject to a percentile response time SLA in a distributed cloud

Keerthana Boloor*, Rada Chirkova*, Tiia Salo, and Yannis Viniotis*

*Department of Electrical and Computer Engineering
*Department of Computer Science
North Carolina State University

◊IBM Software Group
Research Triangle Park
Agenda

- Problem description
- Dynamic request allocation and scheduling scheme
- Comparison with static allocation and FIFO/Weighted Round Robin scheduling scheme
- Conclusion
Problem description

- More web applications are designed to be context aware.
- Most context aware applications are built on SOA principles.
- Cloud computing systems - the most preferred platform for deployment.
- Service Level Agreements (SLA) - terms of service and pricing model.
- What is this presentation about?
Geographically distributed cloud computing system

Diagram:

- Data center hosting K context-aware applications
- Clients
- Data center hosting K context-aware applications
- Data center hosting K context-aware applications
SOA based context aware application

1. Client request with context-id
2. Client request allocated to and scheduled at end-server
3. Load required service-endpoint
4. Load required contextdata
5. Updates to contexts at contextdata stores

DATA CENTER

Cloudcom 2010, Indianapolis, Indiana, USA
An end-server serving multiple user classes

- Each context aware application services multiple classes of users
- Each user class is guaranteed different quality of service based on economic considerations
- SLA specifies different service levels and service charges for the different user classes
Percentile Service Level Agreements

$X\%$ - the fraction of requests of a particular user class which need to have a response time less than $r$ seconds

$P$ - The profit charged by the cloud, if the percentile of requests that have response time less than $r$ seconds is greater than or equal to $X\%$
Problem statement

Allocate and schedule service requests locally at the end-servers so as to globally:

$$\max \sum_{1 \leq j \leq K} profit_j$$  \hspace{1cm} (1)

where $profit_j$ is the profit charged for conformance of the requests from users of class $j$. 
Problem statement

Allocate and schedule service requests locally at the end-servers so as to globally:

$$\max \sum_{1 \leq j \leq K} profit_j$$

where $profit_j$ is the profit charged for conformance of the requests from users of class $j$.

This problem is NP-hard!!
Heuristic-based data-oriented request management scheme

Periodic allocation and adaptation at each datacenter.

Cloudcom 2010, Indianapolis, Indiana, USA
Heuristic-based data-oriented request management scheme

Periodic allocation and adaptation at each datacenter.

- **Adaptation phase**
  Datacenters exchange conformance levels.

- **Allocation phase**
  Rank-based request allocation and gi-FIFO scheduling.

Aim at increasing global profit.
Rank-based allocation and gi-FIFO scheduling

Profit-score calculation

- Profit: $p_k$
- Required global conformance: $c_k$
- Current global conformance: $cc_k$

If $cc_k < c_k$

Profit-score = $p_k / (c_k - cc_k)$

Else

Profit-score = 0
Rank-based request allocation

1. Query hash-based lookup table ([context-id,machine-id] or [service-id,machine-id])

2. Rank-based compatibility test
   
   1. The arriving request is assigned a rank based on its profit-score and deadline.
   
   2. Does the arriving request meet its deadline? - Machine compatible!!!

3. Compatible machine not found? - Choose least loaded closest to context DB
gi-FIFO scheduling

- Choose the request of user class with the highest current profit-score

- Choose one with maximum waiting time but which results in a response time less than or equal to $r$
  
  If no such request exists, choose the request with higher waiting time resulting in a response time greater than $r$

- gi-FIFO has been proven to be the most suitable for percentile SLAs for a single server serving multiple classes.
Evaluation

Dynamic scheme vs static schemes

- Dynamic rank based allocation with gi–FIFO scheduling
- Static allocation with WRR scheduling
- Static allocation with FIFO scheduling

Graph shows the profit incurred ($) against request rate.

Cloudcom 2010, Indianapolis, Indiana, USA
Dynamic rank based allocation vs static allocation scheme

![Graph showing comparison between static allocation and dynamic rank based allocation]

- **Static allocation with gi-FIFO scheduling**
- **Dynamic rank based allocation with gi-FIFO scheduling**

Cloudcom 2010, Indianapolis, Indiana, USA
**Variation in subinterval length**

- Uniform distribution of classes, stringent SLA
- Uniform distribution of classes, relaxed SLA
- Non-uniform distribution of classes, stringent SLA
- Non-uniform distribution of classes, relaxed SLA

**Variation in context update interval**

- Low context data load times
- High context data load times
- Medium Context data load times
Conclusion

- Identified the need for dynamic request scheduling and allocation for context aware applications in a distributed cloud.

- Proposed a novel rank-based request allocation and gi-FIFO scheduling scheme for managing percentile SLAs with an aim to maximize profit obtained by the cloud.
Questions??