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

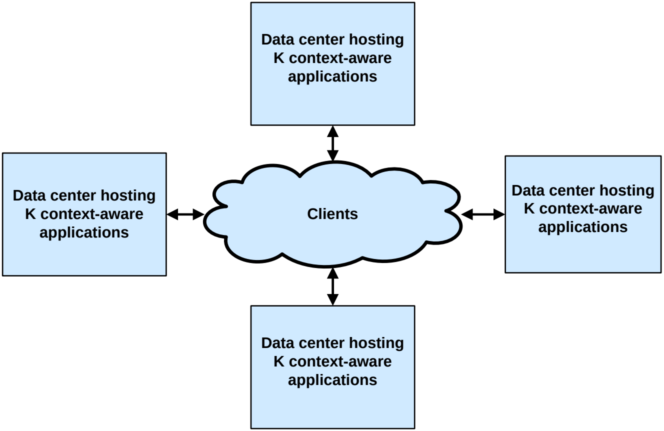
Agenda

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

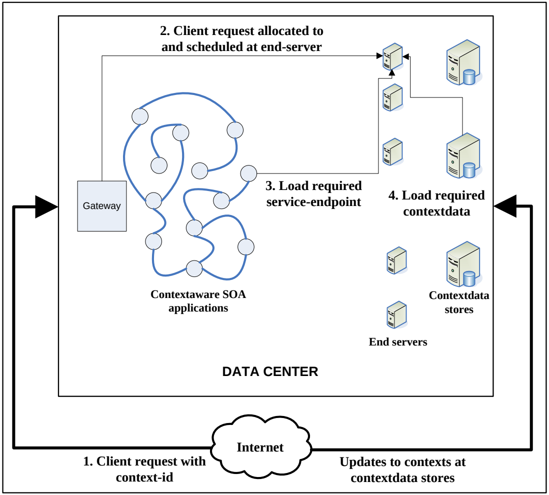
Problem description

- 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?

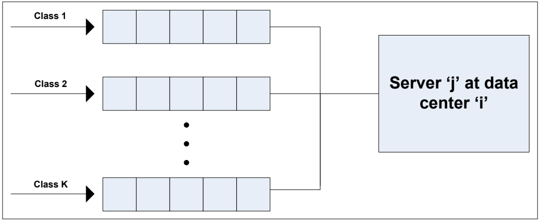
Geographically distributed cloud computing system



SOA based context aware application

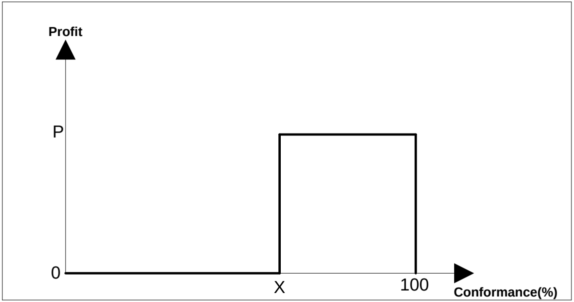


An end-server serving multiple user classes



- 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:

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

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:

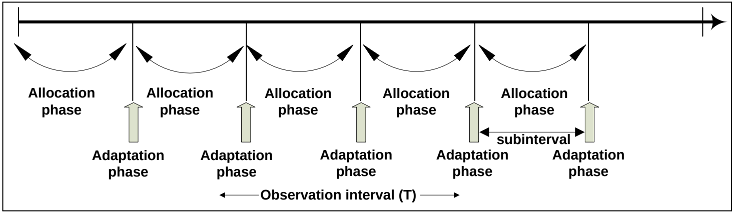
$$\max \sum_{1 \leq j \leq K} \mathit{profit}_j \tag{1}$$

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

This problem is NP-hard!!

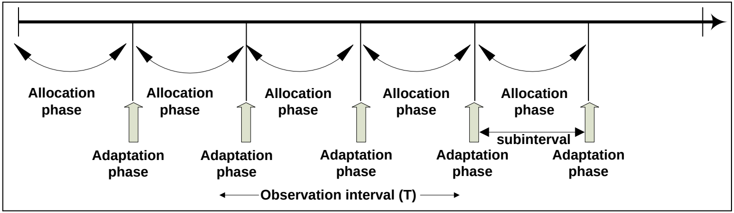
Heuristic-based data-oriented request management scheme

Periodic allocation and adaptation at each datacenter.



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

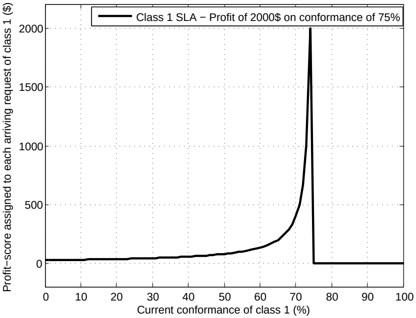
- Profit: p_k
- Required global conformance: c_k
- Current global conformance: cc_k

If $cc_k < c_k$

$$\text{Profit-score} = p_k/(c_k - cc_k)$$

Else

$$\text{Profit-score} = 0$$



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

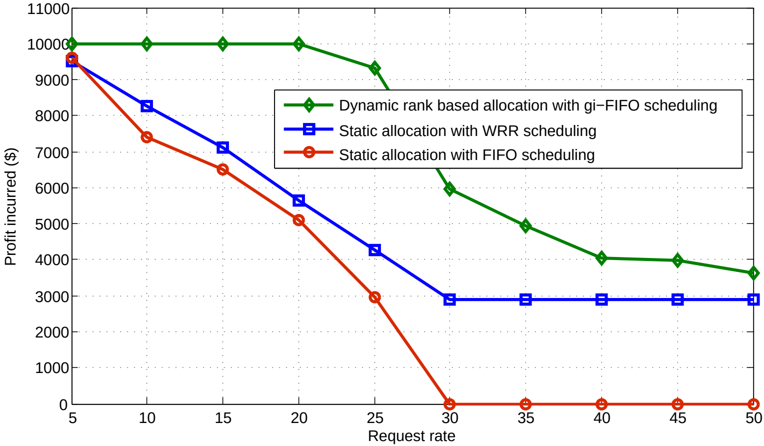
gi-FIFO scheduling

- 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*

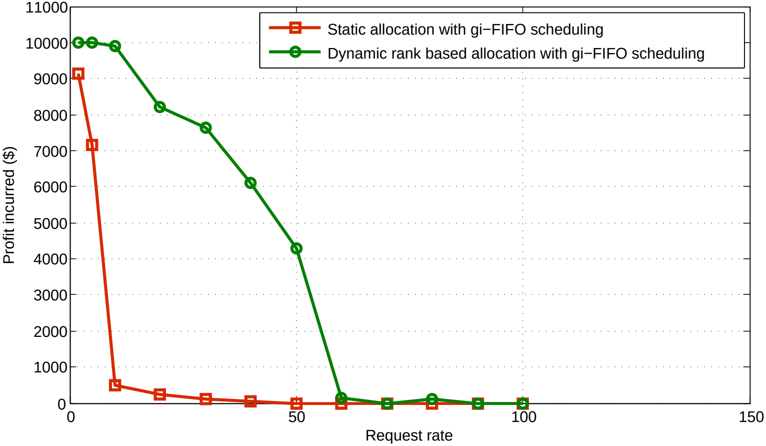
  If no such request exists, choose the request with higher waiting time resulting in a response time greater than *r*
- gi-FIFO has been proven to be the most suitable for percentile SLAs for a single server serving multiple classes.

Evaluation

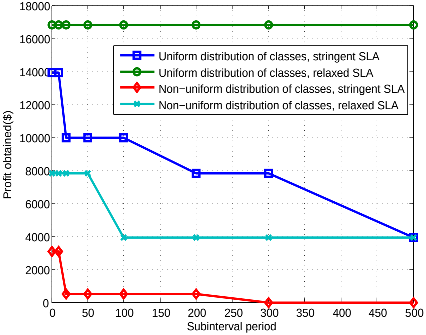
Dynamic scheme vs static schemes



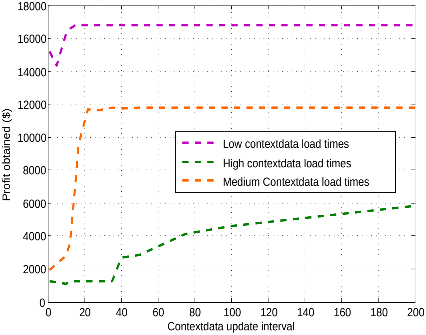
Dynamic rank based allocation vs static allocation scheme



Variation in subinterval length



Variation in context update interval



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??