Simulator Of A "Weather" Cloud

Khramenkova Ksenia,
Hermant Olivier, Pawlak Renaud

Contents

- Term of Cloud Computing
- Cloud Description
- Cloud Simulator
- Simulation Results
- Conclusion

Term of Cloud Computing

It's a relatively recent term, built of research in virtualization, distributed computing, utility computing, networking, web and software services.

It implies a service oriented architecture, reduced information technology overhead for the end-user, great flexibility, reduced total cost of ownership, on-demand services, etc.

Cloud Description (1/2)

CAP 2020 is a French company that aims at providing various services for the French agriculture.

Due to the extremely large amount of data and to the frequency and complexity of requests, the service is deployed on a cloud computing architecture.

Cloud Description (2/2)

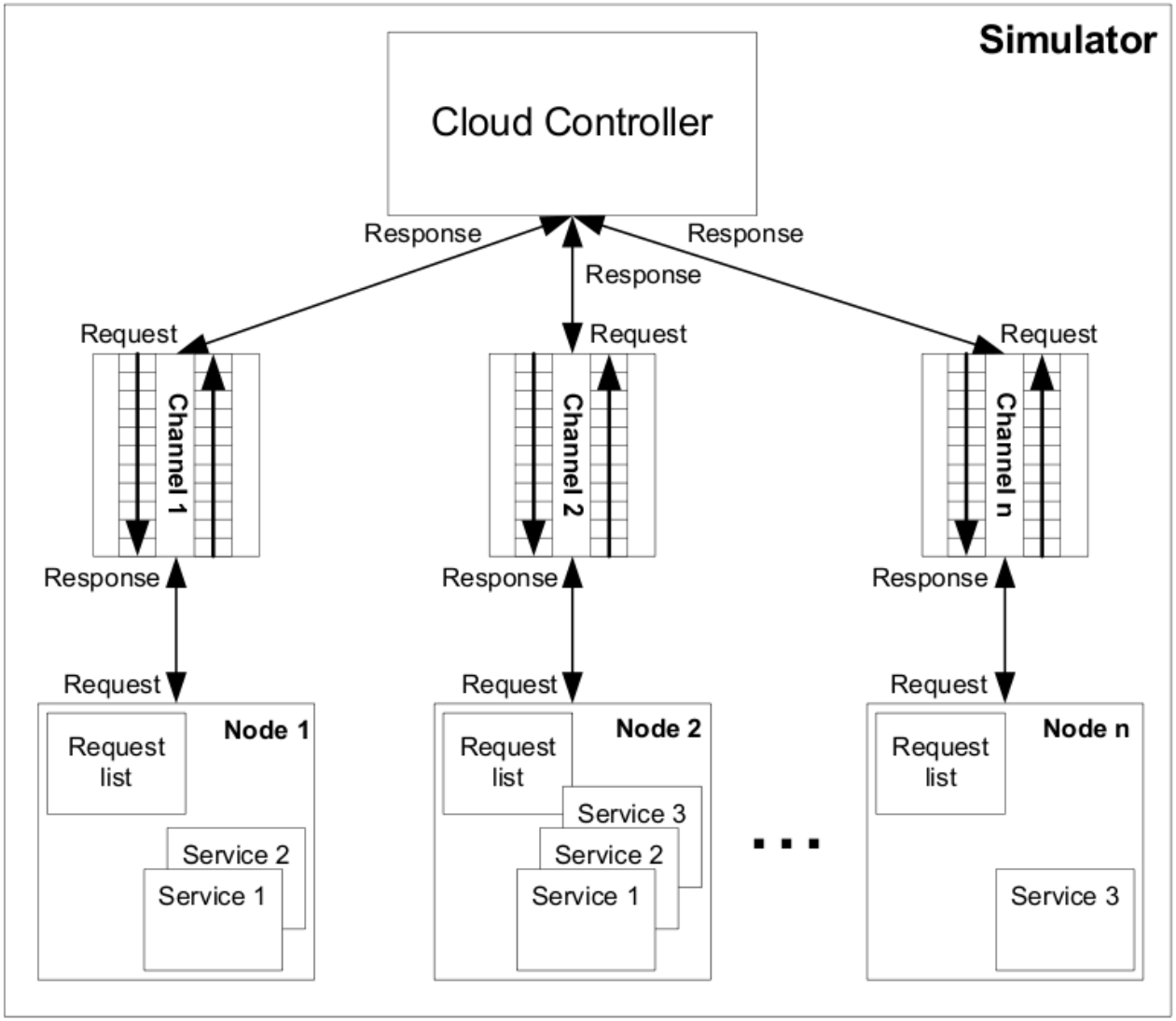
We noticed that the computation time required by the same request may vary drastically: it can be up to 4 times faster in some cases. An analysis of this fact lead us to remark that requests are processed faster when the data files required by the user are already in the cache memory of the system.

The goal we identified for task allocation process, is to send requests on the node that, most probably, has already loaded the required data files in its cache memory.

Cloud Simulator (1/3)

A cloud simulator is used for modeling cloud architecture for performance analysis and optimization. During a simulation bugs can be discovered and fixed before moving on to the implementation phase in which debugging will be a lot of work.

Cloud Simulator (2/3)



Cloud Simulator (3/3)

Round Robin (RR)

- The algorithm does not analyze the request.
- The controller selects a node pointed to by a circular counter from a list

The algorithm with an analysis of the contents of the request (AAR)

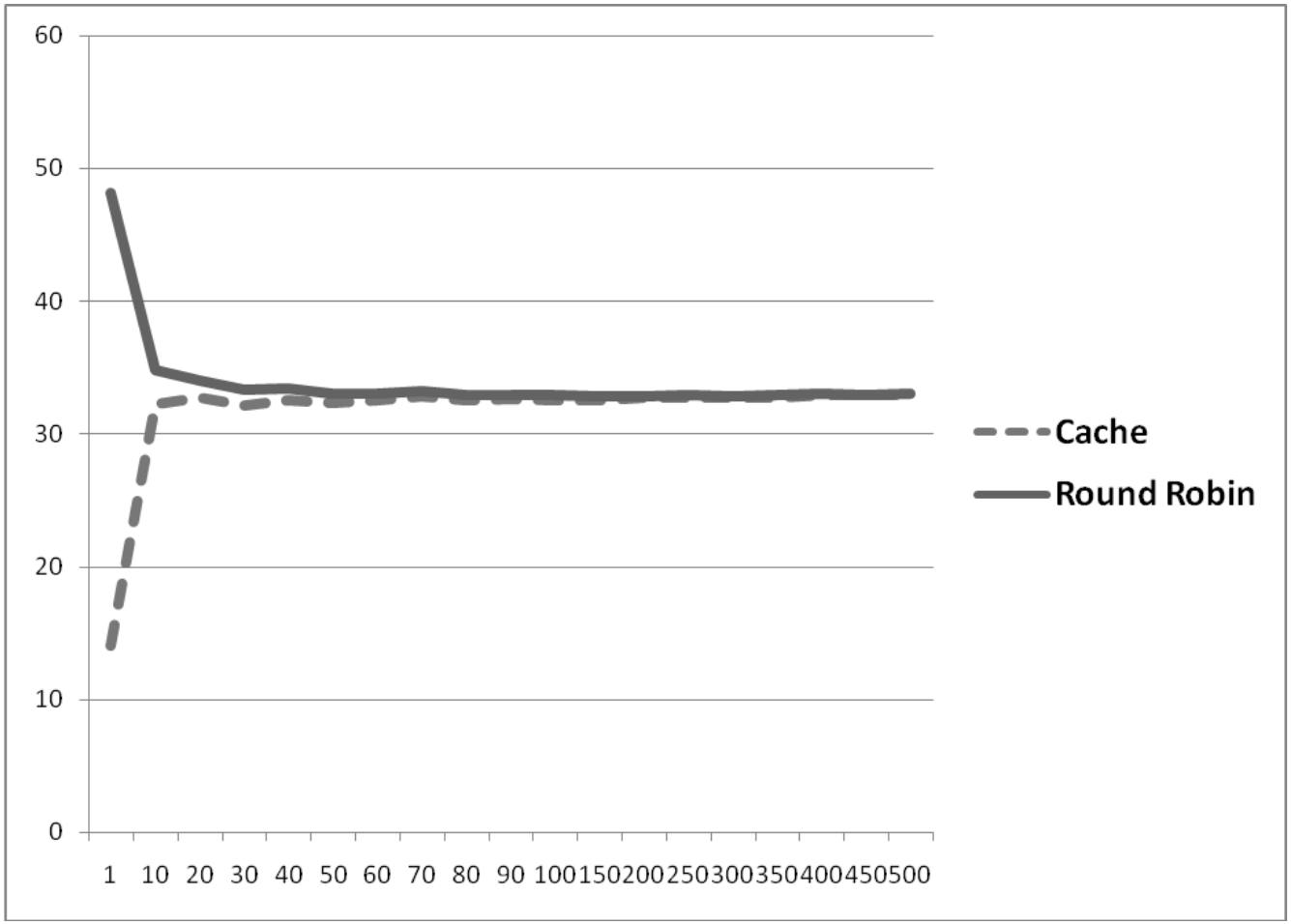
- The algorithm analyzes the content of the request namely the array of documents requested by user.
- Analyzes the percentage of requested documents (cache hit) for each node and chooses by the biggest value where the current request has to go

Simulation Results (1/4)

As a result of the modeling we get a table containing:

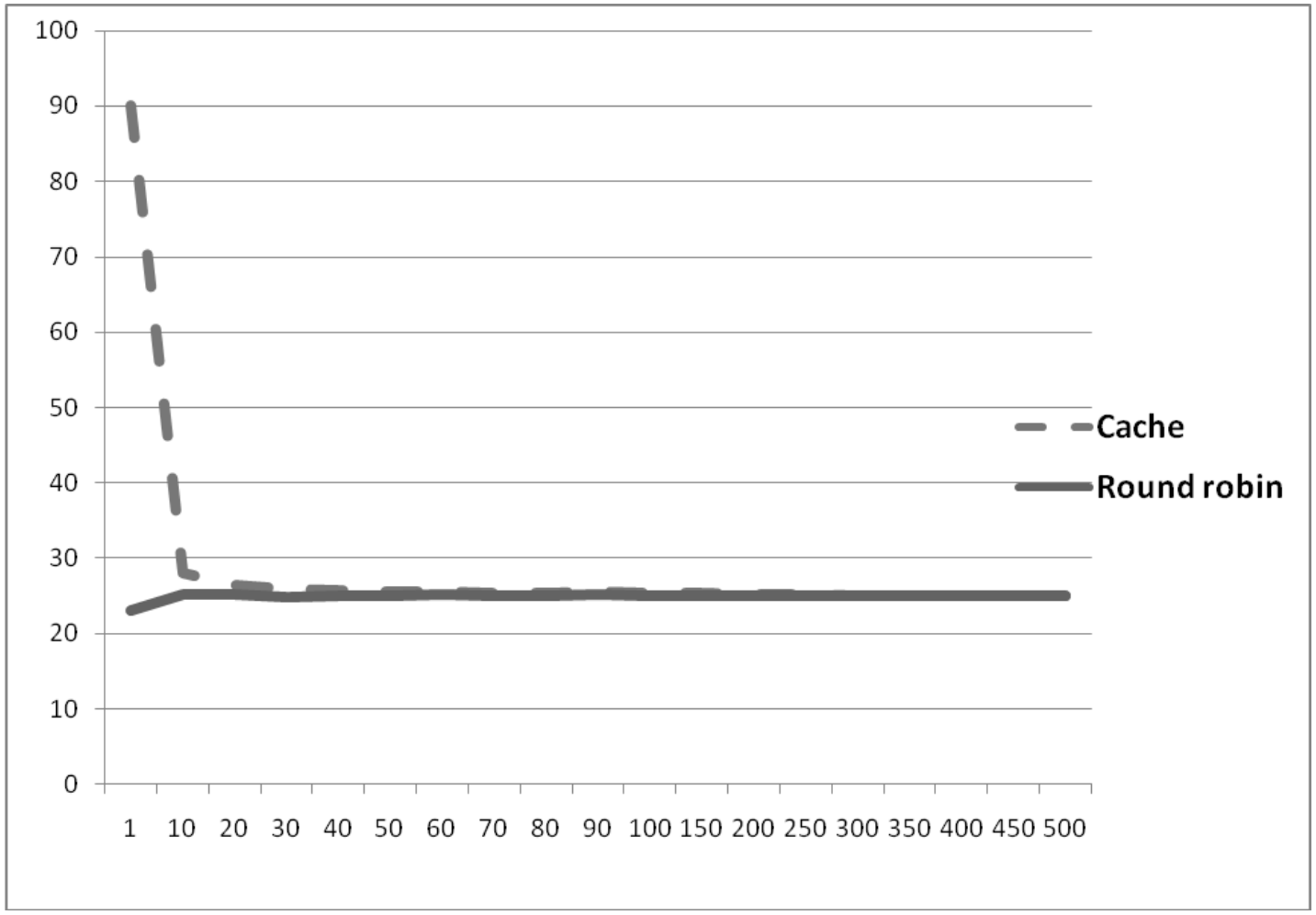
- task identifier,
- service type,
- node identifier,
- start time,
- finish time,
- cache hit percentage,
- input data size,
- output data size

Simulation Results (2/4)



The processing request time

Simulation Results (3/4)



The percentage of cache hit

Simulation Results (4/4)

Some comments for these results:

- The array of documents has been chosen randomly. In the AAR case, more than 25% of the documents are in the target range, but the cache memory of the node is polluted with (at most) 75% of the other documents.
- We can observe in the RR case of figure 2 a decrease of the processing time, while we would expect a stable time. This can be explained by the fact that the cache memory is empty at the beginning, and the more documents there is in a request, the faster it fills, and the faster we obtain a stable 25% cache hit.

Conclusion

The results of the analysis are: the second algorithm gives a small gain in time and cache hit on small amount of requested data. So, we need to improve the algorithm with an analysis of the contents of the request.

When requests are distributed by controller, it has to know dynamically the state of the cache memory of the nodes so as to calculate dynamically the expected cache hit percentage. After analyzing array of requested documents in the current request, the controller can then choose the optimal node for the current request, from the point of view of cache hit.

THANK YOU