From Cosmos to Cloud, Bringing Clients closer to Services

Sorry! That’s a lie.

Because, in Service Oriented Approach (SOA) we can’t get Client and Service processes (SPs) closer (or in other words, talk faster) especially with heavy input or output messages because of latency issues.

But that’s the problem!

How to make SOA efficient for communication intensive scientific apps to produce HPC performance? (especially apps like N body problem of planetary systems, which are iterative and communication intensive)

Solution?

Change the paradigm and get my trophy!

My Trophy*

Moderate but significant improvement (when comparing new paradigm with old for the same N body problem).

SOA programming after my Phd

Data Service

Service Proc

Data Service

Service Proc

Data Service

Client

Input data

Sync Command

put

get

Sync Command

get

put

Sync Command

Output data

get

SOA programming before my Phd

Input data

Output data

Input data

Output data

Input data

Output data

* For N-body app, on ANU SoCS internal cluster machines
* Without load balancing efforts (which is my future work to finish Phd :)

It is true..

...that we can reduce communication cost between Client and SPs between many generations of tasks by reducing communications and data transfer with the help of a Data Service (DS). (Please refer the figure).

How does it work?

Data common to all SPs can be initiated by the Client to DS. SPs can update DS data using put. But put is a differed operation which will be reflected only for the next generation of tasks at the (sync) request of the Client.

Result!

In N Body problem, Client app logic could be shifted to Service side with significant improvements in speed. More apps in the making.

The By-Product

(A very happy ending :)

Gosh!

If this were in a “Cloud”?!#@* ..latency would have killed me!.. (without the new paradigm!)

Like this..

Industry

Platform Computing

Jaison Paul Mulerikkal

Peter Strazdins (Supervisor)

Web: memyviews.com

Pictures © NASA, Axel Rouvin

ANU COLLEGE OF ENGINEERING & COMPUTER SCIENCE