INTEGRATING COMPUTING RESOURCES ON MULTIPLE GRID-ENABLED JOB SCHEDULING SYSTEMS THROUGH A GRID RPC SYSTEM

Yoshihiro Nakajima

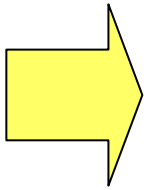
Advisor: Mitsuhisa Sato

University of Tsukuba

2007/10/30

Motivation

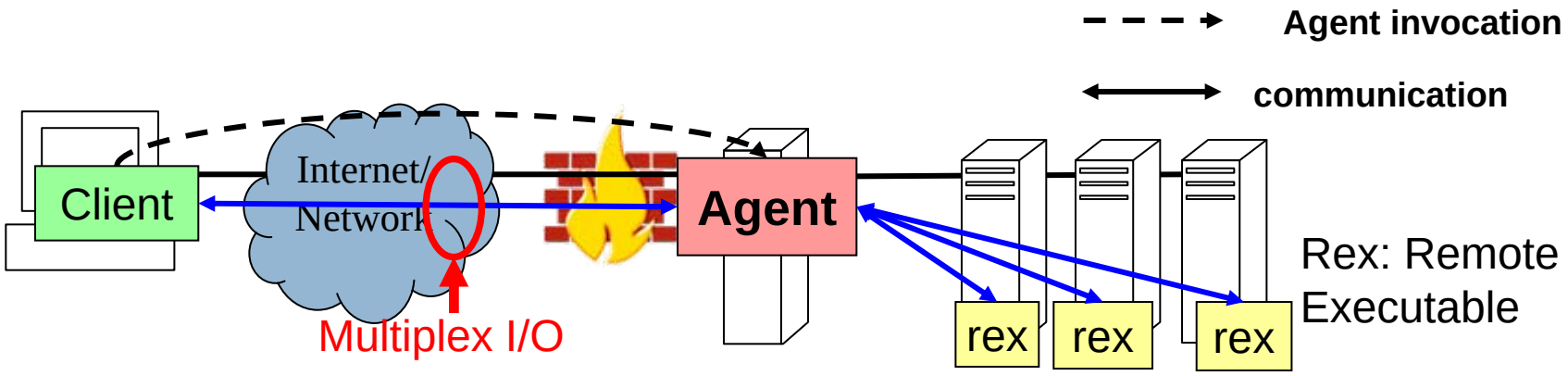
- Need for High Throughput Computing (cf. Simulations for drug design, Circuit design…)
  - Many kinds of Grid-enabled Job Scheduling System (GJSS) have been developed
    - XtremWeb, Condor, Grid engine, CyberGRIP, GridMP…
- User wants to use massive computing resources on different sites easily
  - Different management policy and middleware on each sites
  - User should write extra code to adapt environment
  - User does not want to stop calculation by system faults (Needs for transparent **Fault-tolerance mechanism**)



Provide RPC style programming model on GJSS

Background: OmniRPC as an example of Grid RPC system

- Provide seamless parallel programming for local cluster to multi-cluster in a grid environment
- Make use of remote PC clusters as Grid computing resources
- OmniRPC consists of three kinds of components: Client, Remote executable, **Agent**
- OmniRPC agent works as a proxy of the communication between client program and remote executables

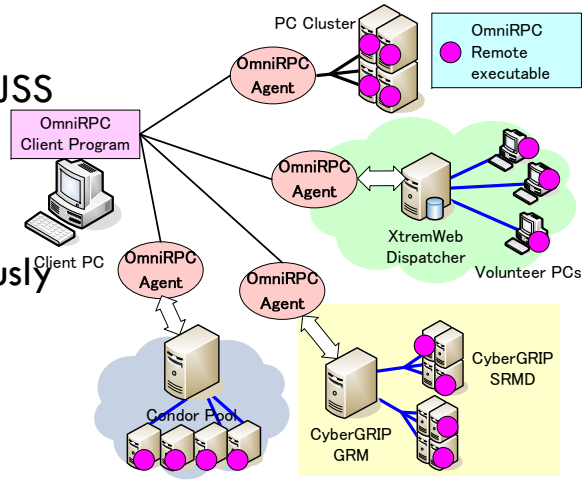


Objective and Design of Grid RPC system for integrating computing resources on GJSS

- **Objective**
  - Provides unified and parallel programming model by RPC on GJSS
  - Provide Fault-tolerant features for Grid RPC system on the worker programs
  - Exploit massive computing resources on different sites simultaneously
- **Design of proposed system**
  - Decoupling computations and data transmission from RPC mechanism
  - Design the agent mechanism to bridge between Grid RPC and GJSS
  - Using document-based communication, rather than connection-based communication
  - APIs to adopt different GJSS's

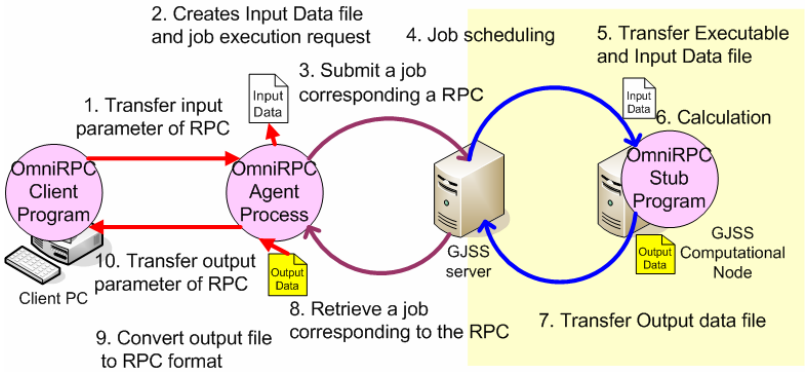
The proposed system can

- **Submit a RPC computation as a job to GJSS**
- **Guarantee transparent fault-tolerant execution on the side of worker programs**



Overview of the proposed system

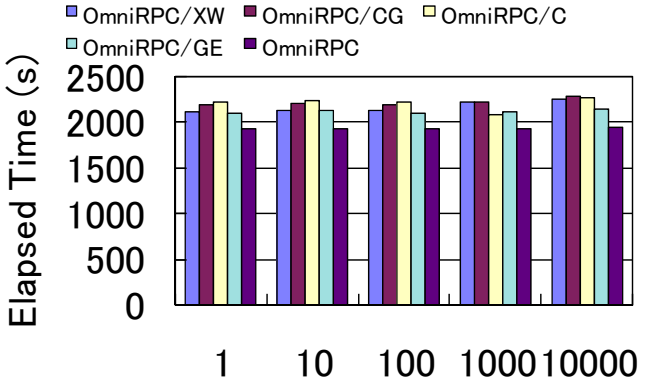
Model of OmniRPC on a GJSS



Implementation and Performance evaluation of the Proposed system

- We have implemented the system to exploit 4 major grid job schedulers
  - Condor (U of Wisconsin)->OmniRPC/C
  - Grid Engine (SUN) -> OmniRPC/GE
  - XtremWeb (INRIA) -> OmniRPC/XW
  - CyberGrip (Fujitsu lab) -> OmniRPC/CG
- The proposed system archives almost the same performance of original OmniRPC
  - OmniRPC/{XW,CG,C,GE} have small performance degradation (10% lower) compared with original OmniRPC
- Fault-tolerance facility of the proposed system works well and user can take advantage of the facility

Elapsed time in case of issuing 128 PRCs (240 sec per RPC) in parallel with 16 nodes [ideal elapsed time takes 1920 sec]



# of running jobs when injecting artificial faults

Discussion: Pros and Cons of the proposed system

Pros

- OmniRPC/{XW,CG,C,GE} can use large-computing resource pools managed by GJSSs on different sites.
- OmniRPC/{XW,CG,C,GE} can take advantage of fault-tolerant facility in worker program's side
  - The original OmniRPC does not support FT

Cons

- Performance of the OmniRPC/{XW,CG,C,GE} might be lower than that of the original OmniRPC
  - If the computation time of a RPC takes longer (more than 10 min), performance degradation will be negligible

Summery and Future Works

- We have proposed a framework for a parallel programming model by RPC for integrating large-scale computing resource pools by GJSSs
  - Agent bridging RPC and job scheduling system by converting connection-based communication to document-based communication
- We have implemented the system as an extension of the OmniRPC system on XtremWeb, CyberGRIP, Condor and Open Source Grid Engine
  - Achieve as approximately same performance as that of original OmniRPC
  - Can takes advantage of FT in worker program side
- Future Works
  - Apply to another GJSS
  - Dynamic RPC scheduling by using computing usage information in each site