

The Environments of Geant4 Parallel Execution using GRID Technology

CREST, JST

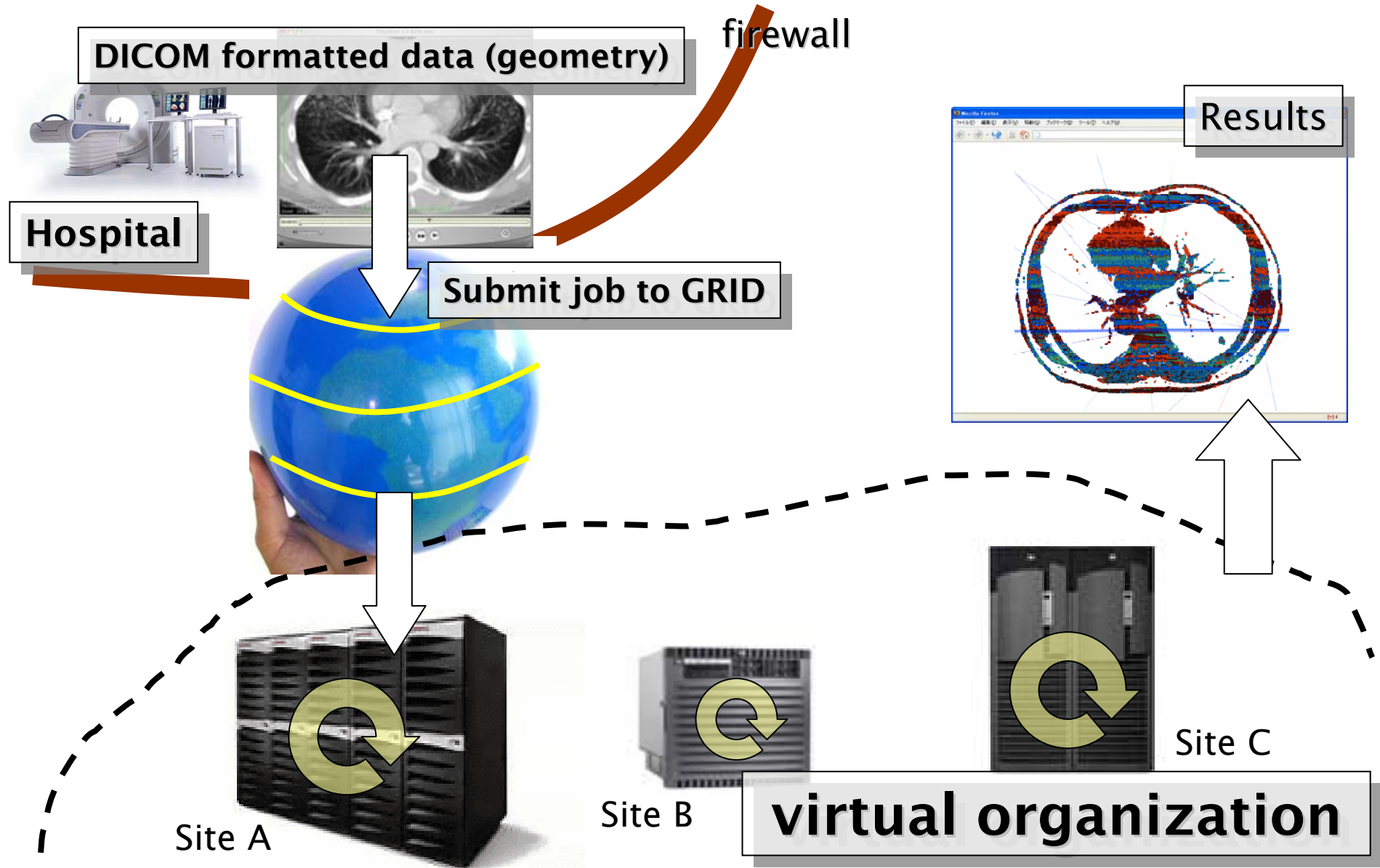
Go IWAI

iwai@post.kek.jp

Introduction

- Purpose
 - Development of software framework for simulation radiotherapy
- Goal
 - provide the framework and software toolkit for simulation in radiotherapy
 - quickly, high-accuracy and securely
- Problem
 - Long term execution
 - Trade off between accuracy and simulation speed
 - Firewall
 - Most of users use our software from hospital, they are strongly protected by firewall

Assumed Work Flow

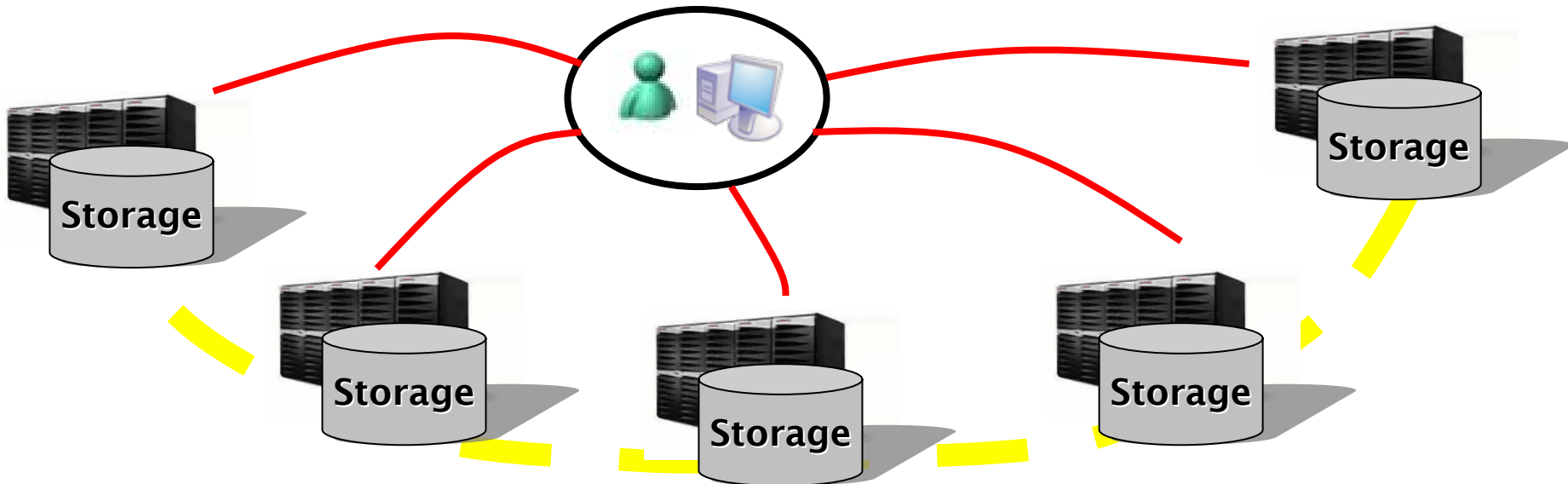


Solution

- Geant4 Parallelization
 - to be short response time of applications
 - to parallelize exist Geant4 applications rapidly and easily
 - with MPI
- System Construction with Grid (LCG-2)
 - for wide-area distributed computing environments
 - to submit job from remote site with strong security
 - to share and/or control distributed files
- Development of web interface is also the important issue

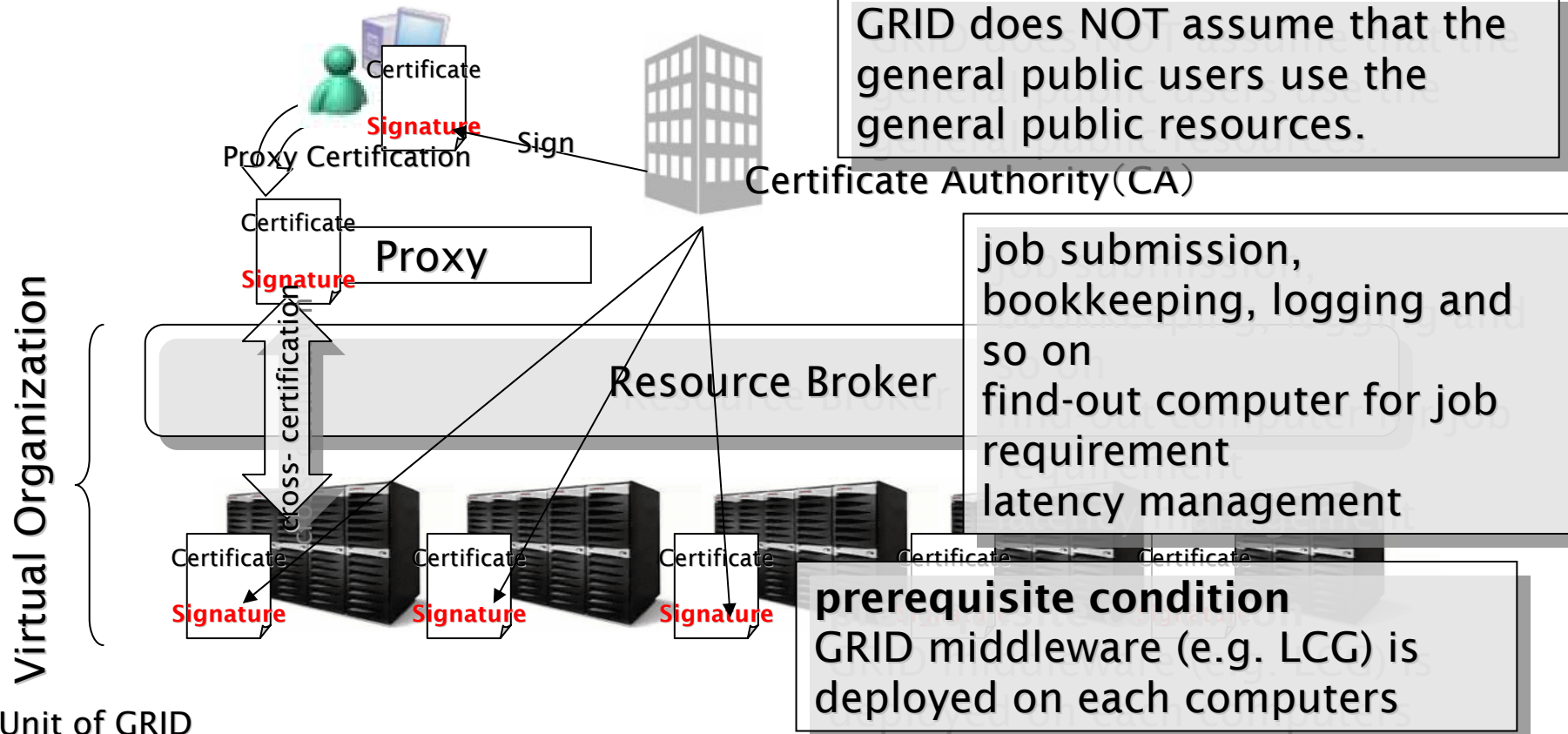
Ideal of GRID

- Sharing computing resource
 - CPU, storage, memory
- Users can use distributed computing resources from their PC



Current Status of GRID

GRID does NOT assume that the general public users use the general public resources.



Unit of GRID

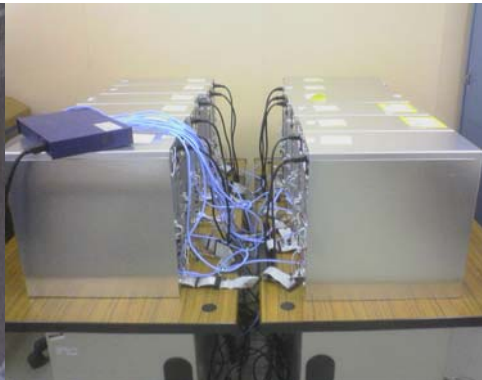
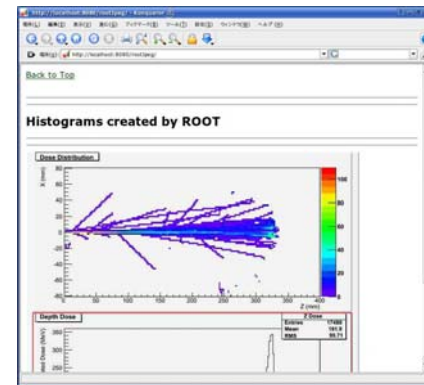
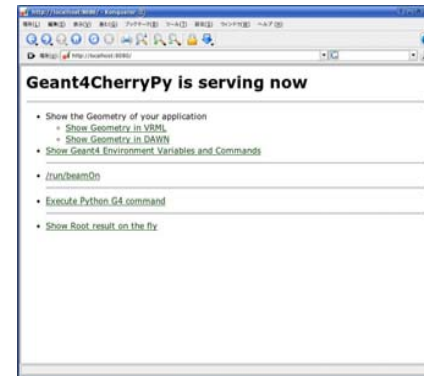
- Virtual Organization (VO)
- like a huge queue

Single sign-on

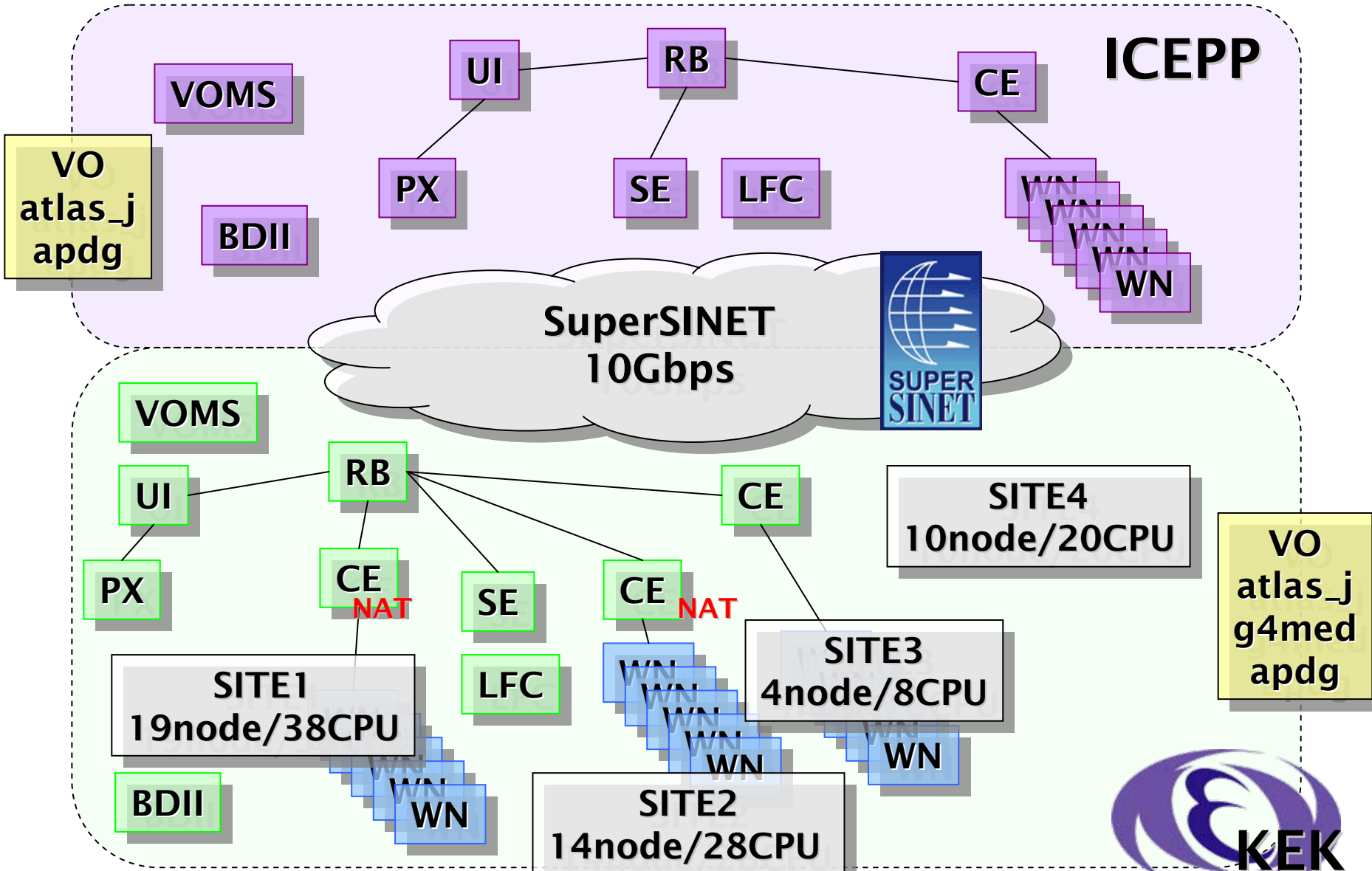
- Proxy certification
- Each certification need signature by CA
- Cross-certification
- based on SSL and PKI

Recent Activities around KEK

- Sharing efforts with Atlas/LHC
- Setting test-bed based on LCG-2.6, gLite
 - OS: Scientific Linux 305
 - CPU: Pentium, Celeron, Opteron and Athlon
 - 57node/104CPU
- Testing inter-connectivity between KEK and ICEPP (Univ. of Tokyo)
- Testing Function of middleware
- Measuring performance of data sharing
- Atlas simulation software was installed to demonstrate regional resource sharing
- Developing and providing web service
- Federation in Asia-Pacific region

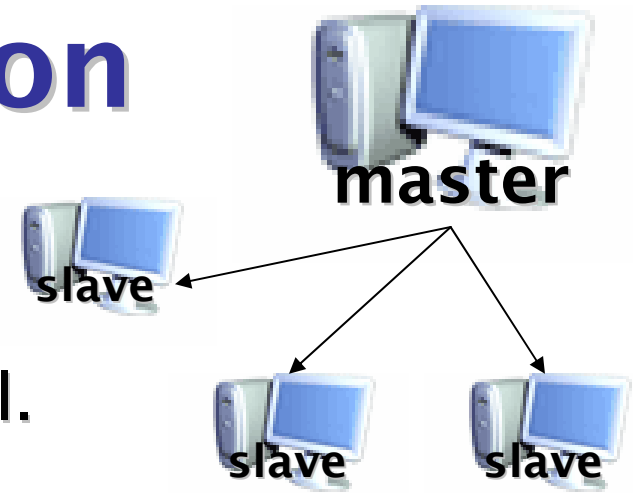


KEK-ICEPP



Geant4 Parallelization

- Parallel software
 - was implemented with MPI.
 - is based on a master/slave model.
 - event-level parallelization
 - Each event is scattered to a slave.
 - Results in each slave are gathered into the master at the end of event/run.
 - A part of Geant4 framework



- 20node/40CPU
- AMD Opteron Processor 250
 - 2CPU on each node
- Memory 4GB
- Gigabit ethernet

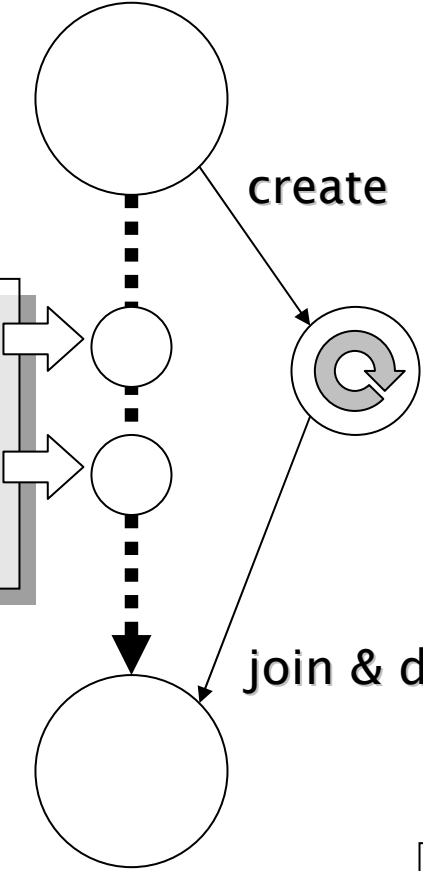


Parallelize Methodology

main thread (session)

Event loop is cause of maximum delay, so user is waited during processing.
A sub-thread is assigned to each "event loop".

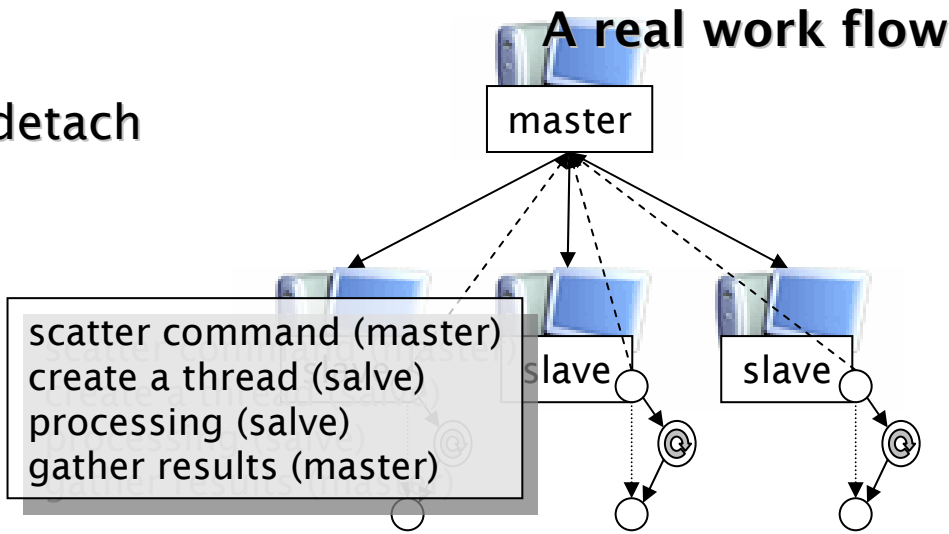
User can interrupt and get the newest information from each slaves in arbitrary timing.



sub-thread (event loop)

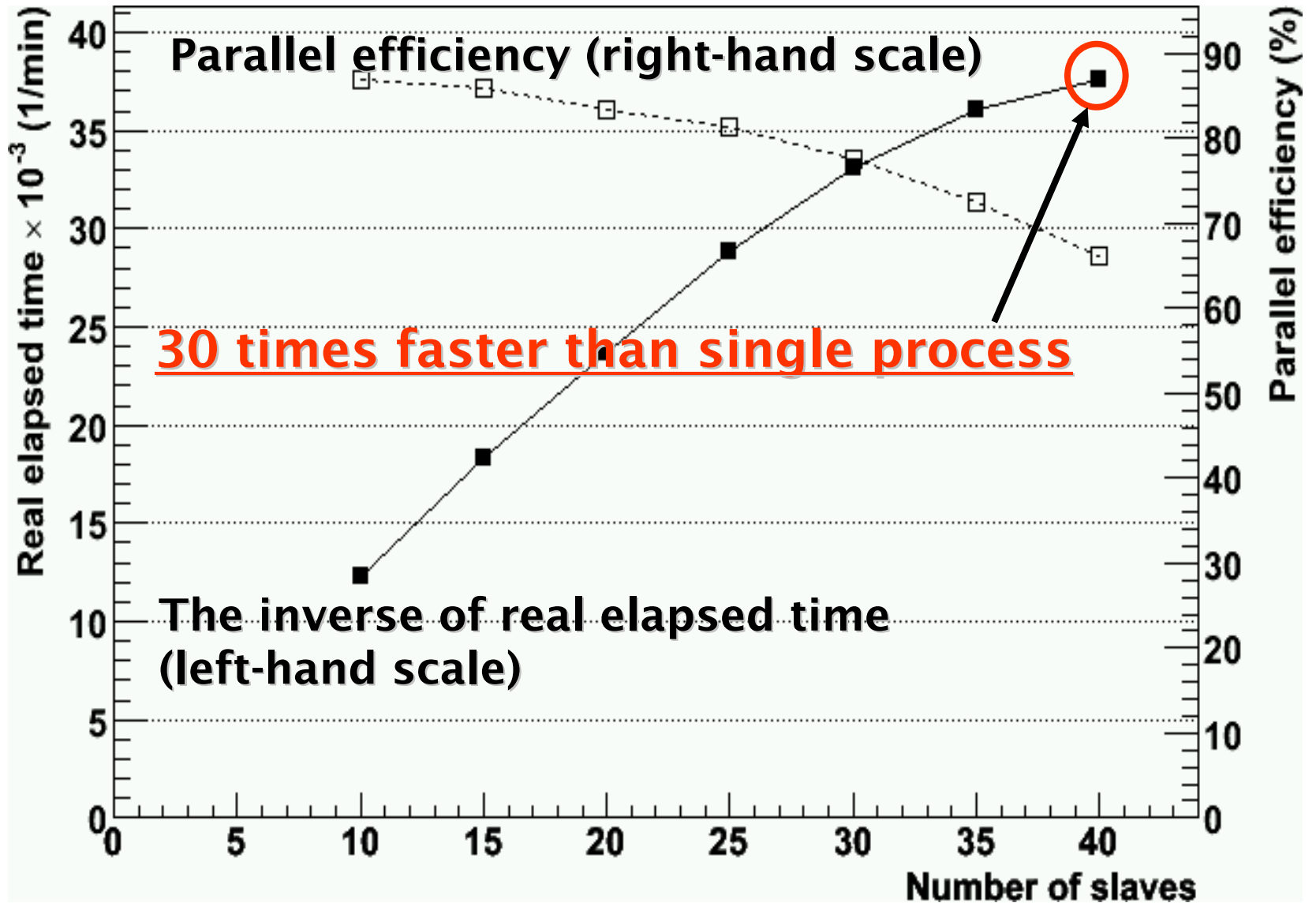
join & detach

A real work flow



scatter command (master)
create a thread (salve)
processing (salve)
gather results (master)

Benchmarking Results



Summary

- We can get a secure and efficient way of sharing computing resources in the context of Grid technology
 - Secure globally-distributed job execution
 - We can use about 100CPU between KEK and ICEPP.
- We are also developing an easy-to-use user interface on our Grid environment.
 - Web application interface
- Geant4 parallel components have been developed using MPI based on master-slave model.
 - Existing applications can be parallelized rapidly and easily because these components are deigned as a part of Geant4 framework.
 - Applications based on these components are executed with one master and one or more slaves.
 - Users can get the newest information anytime through using threads.
- An existing our medical application based on Geant4 has been parallelized with parallel components and executed on PC cluster.
 - If this software is executed as single process, elapsed time is 10 hours and over.
 - When parallelized application was executed with 20 nodes (40 CPUs), its performance was nearly 30 times faster than single process execution.
 - The Parallel efficiency is 67%.
 - There is not a difference between results by parallelized and non-parallelized application.