

Korea-Japan Grid Symposium, Seoul  
December 1-2, 2004

## P3: Personal Power Plant

Makes over your PCs into power generator on the Grid

Kazuyuki Shudo <[shudo@ni.aist.go.jp](mailto:shudo@ni.aist.go.jp)>,

Yoshio Tanaka,

Satoshi Sekiguchi

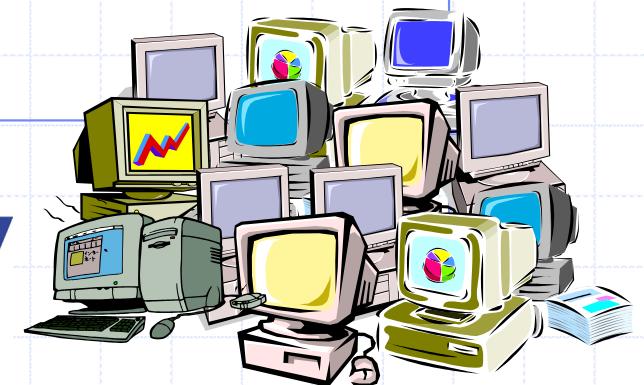
Grid Technology Research Center, AIST, Japan



Personal  
Power  
Plant



Grid  
Technology  
Research  
Center  
AIST



# P3: Personal Power Plant



Personal  
Power  
Plant

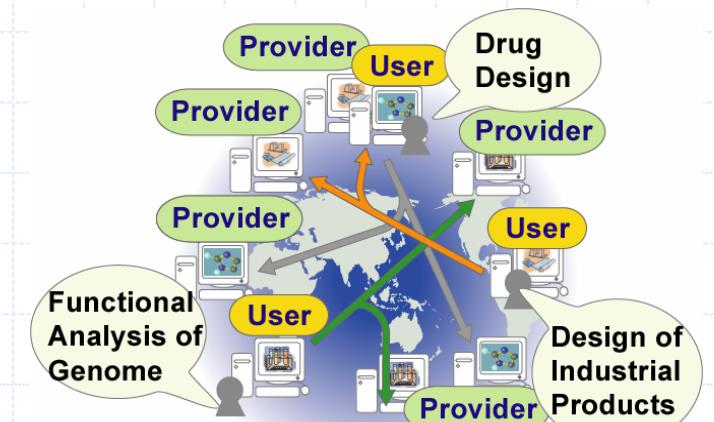
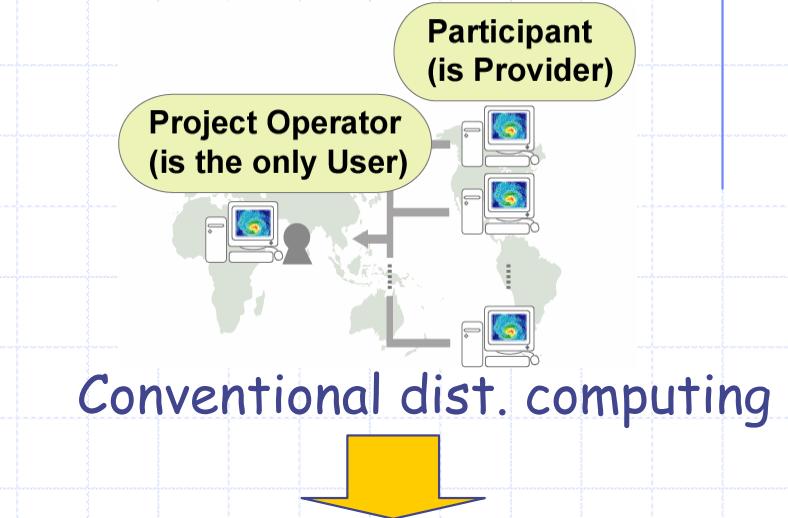
## ◆ Middleware for distributed computation utilizing JXTA.

### ◆ Traditional goals

- Cycle scavenging
  - ◆ Harvest compute power of existing PCs in an organization.
- Internet-wide distributed computing
  - ◆ E.g. distributed.net, SETI@home

### ◆ Challenging goals

- Aggregate PCs and expose them as an integrated Grid resource.
  - ◆ Integrate P3 with Grid middleware ?  
cf. Community Scheduler Framework
- Dealings and circulation of computational resources
  - ◆ Transfer individual resources (C2C, C2B) and also aggregated resources (B2B).
  - ◆ Other resources than processing power.
  - ◆ Commercial dealings need a market and a system supporting it.



Personal  
Power  
Plant

AIST

# P2P way of interaction between PCs

- ◆ P3 uses **JXTA** for all communications
  - JXTA is a widely accepted P2P protocol, project and library, that provides common functions P2P software requires.
- ◆ P2P concepts supported by JXTA efficiently support P3:
  - **Ad-hoc self-organization**
    - ◆ PCs can discover and communicate with each other without pre-configuration.
  - **Discovery**
    - ◆ PCs dynamically discover each other and jobs without a central server.
  - **Peer Group**
    - ◆ PCs are grouped into job groups, in which PCs carry out **code distribution**, **job control**, and **collective communication** for parallel computation.
  - **Overlay Network**
    - ◆ **Peer ID** in JXTA is independent from physical IDs like IP addresses and MAC addresses.
    - ◆ JXTA enables **end-to-end bidirectional communication** over NA(P)T and firewall (even if the FW allows only unidirectional HTTP).
    - ◆ This function supports parallel processing in the message-passing model, not only master-worker model.

# Other Benefits from JXTA

## ◆ Scalability

- JXTA Project set its scalability target as 300,000 peers are active in 1,500,000 peers.

## ◆ Configuration-less

- A P3 peer can discover other peers and submitted jobs with JXTA's discovery function.

## ◆ Multi-protocol

- JXTA relay peers mediate messages between TCP, HTTP, IP multicast and possibly other protocols like Bluetooth.



Personal  
Power  
Plant



# Design Goals

## ◆ Application neutral

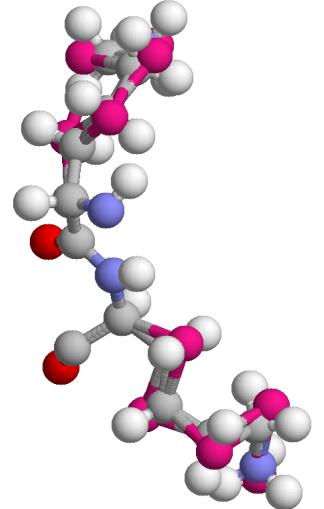
- cf. Client software of traditional dist. comp. projects (e.g. distributed.net) is tightly coupled with a few applications.
- P3 is **decoupled from applications** and users can submit apps into a PC pool.

## ◆ Practical

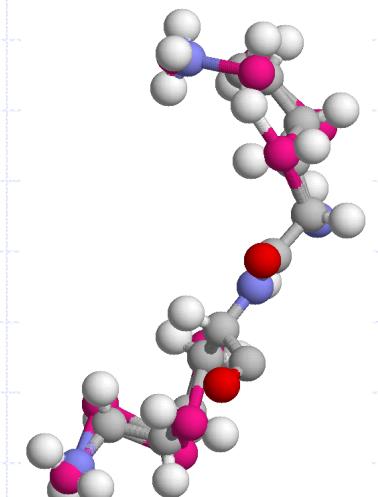
- not only for research.
  - There have been many many middleware for research purpose.
  - Development of P3 is funded by a government agency to promote the development of economy.
- A **Protein-Folding** application is working on P3. Practicality of P3 has been improved with them.

## ◆ Scalable

- We could test P3 with only dozens of PCs so far.
- But we're **measuring other scalability factors** including throughput of workunit-processing by a master.



3D structure  
of a protein



# Design Goals (cont'd)

## ◆ NA(P)T and firewall traversable

- Now, Most PCs are located behind a firewall on the Internet.
- To overcome this restriction, many dist. comp. systems use only HTTP as communication protocol and limit communications to one-way (client → server).
  - ◆ e.g. United Devices' GridMP
- P3 uses JXTA for all communications
  - ◆ All P3 peers can communicate with each other bidirectionally.
  - ◆ P3 provides a message-passing API besides master-worker API.



Personal  
Power  
Plant



# Design Goals (cont'd)

## ◆ Choice of applications by PC providers

- PC providers (participants in a dist. comp. project) should be able to choose jobs to which their PCs are devoted.
  - ◆ It is very important for PC providers to be able to control their own resources.
- In a traditional Internet-wide project, a PC provider has only one choice, install or not.
- Using P3, a PC provider can confirm a digital signature of a job and decide whether to accept it or not.

## ◆ Adaptation to both intra- and Internet

- On the Internet, we have to assume that there are malicious PC providers.
  - ◆ they will try to cheat the software and the operators of the project. E.g. pretending to finish calculation, DoS attack and so on.
- P3 can confirm the correctness of collected results by voting.
  - ◆ Distribute identical workunits and verify the returned results.
  - ◆ This function can be disabled and a verifying logic can be substituted.



Personal  
Power  
Plant



# Design Goals (cont'd)

## ◆ Easy deployment and automatic updating

- The amount of installation and updating labor are proportional to the number of PCs and can be huge.
- Vulnerable client software will be mostly left as it is if the software cannot be updated automatically somehow.
  - ◆ A vulnerability was found in SETI@home client software in April 2003.
- P3 can be **installed by only mouse-clicks** on a web page and **updated automatically**.
  - ◆ cf. Java Web Start (JWS)



Personal  
Power  
Plant



# Structure of P3

- ◆ P3 currently consists from of three subsystems
  - Job management, Parallel programming lib., Job monitor

## ◆ Job management subsystem

- Host jobs (submitted apps) and control their execution.
  - ◆ Host: A daemon program runs on a provided PC.
  - ◆ Controller: by which a resource user submit and control jobs.

## ◆ Parallel programming libraries

- Application programs that use these libraries can run on P3.
  - ◆ Master-worker
  - ◆ Message Passing (like MPI)

## ◆ Job monitor

- Shows the state of jobs and Hosts
- Web-based



Personal  
Power  
Plant

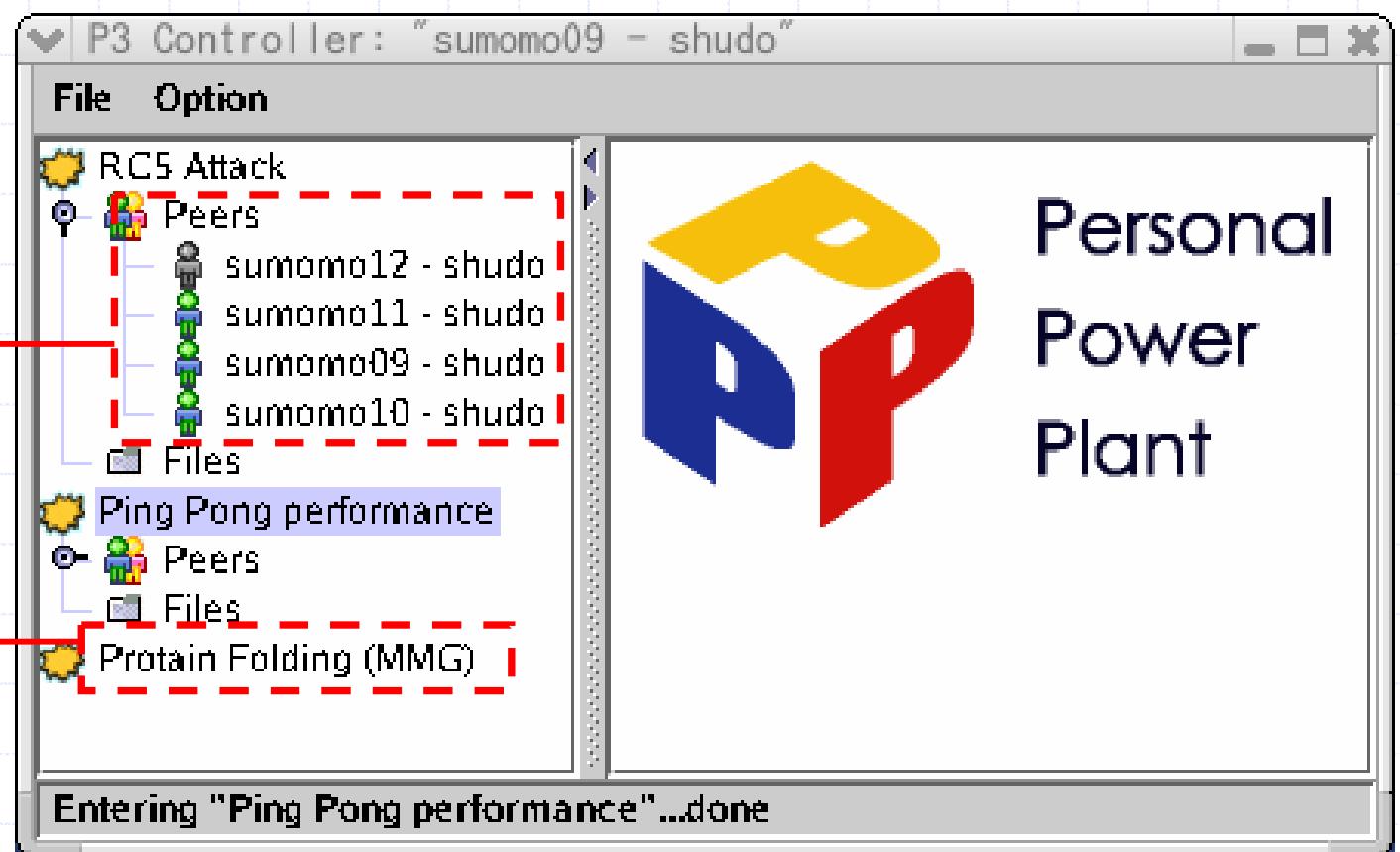
AIST

# Job Management Subsystem: Controller

- ◆ A resource user submits and control jobs with Controller.

Attending Hosts

A submitted job



Personal  
Power  
Plant

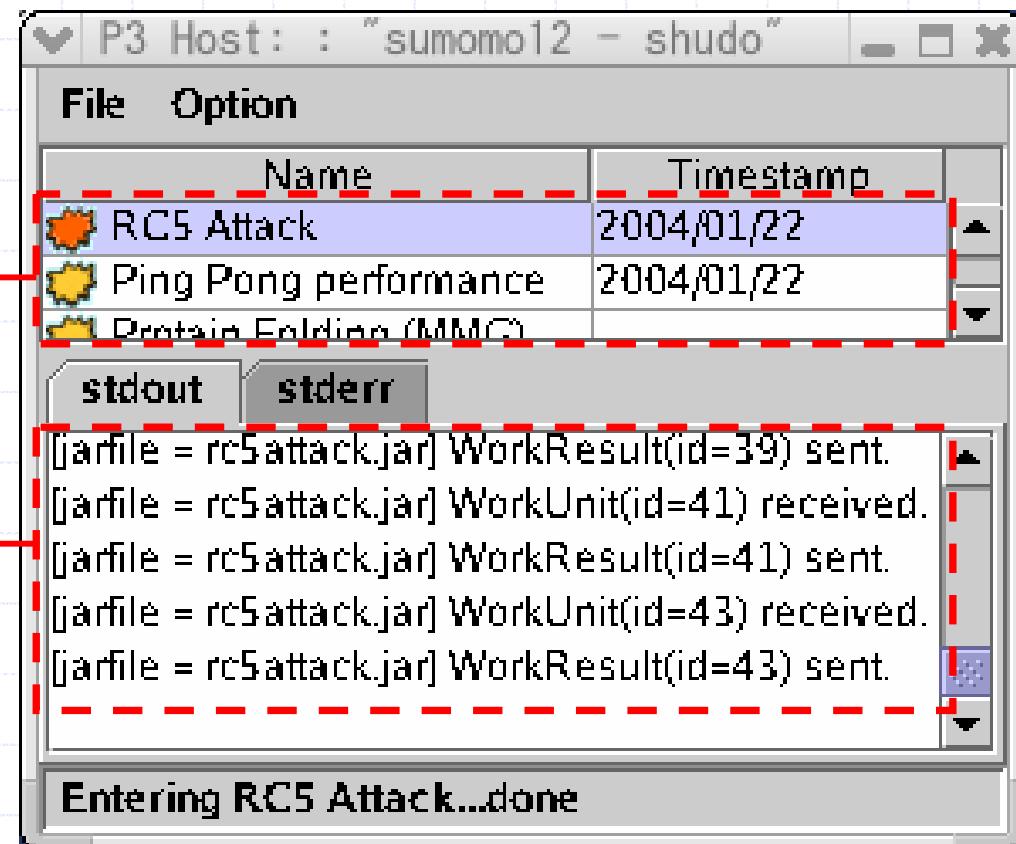


# Job Management Subsystem: Host

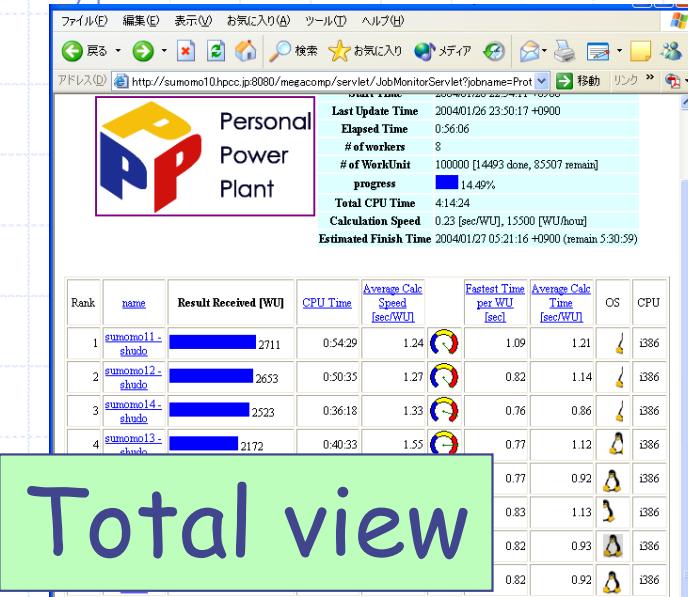
- ◆ A daemon program runs on a provided PC.
  - A Host can run in a head(GUI)-less mode.
    - In that case, the Host decides whether to join a found job or not according to a policy supplied by the PC provider (owner).
  - Host can host multiple jobs simultaneously.

Discovered jobs

Output from  
a running job

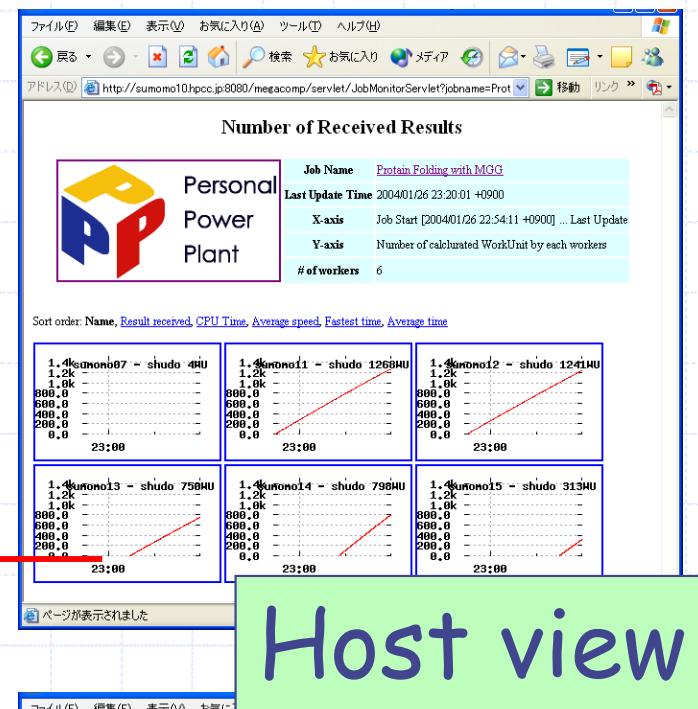


# Job Management Subsystem: Job Monitor

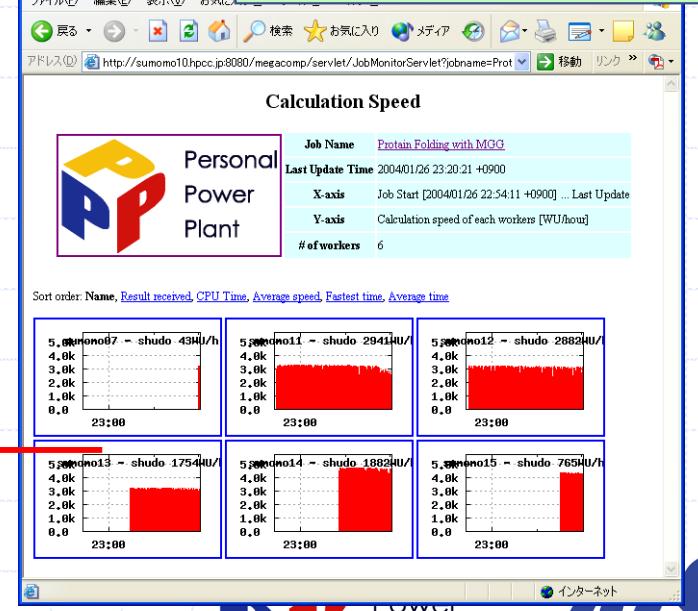


Web browser

Number of processed workunits



Calculation speed



# Job Management Subsystem: Job Monitor (cont'd)



Personal  
Power  
Plant

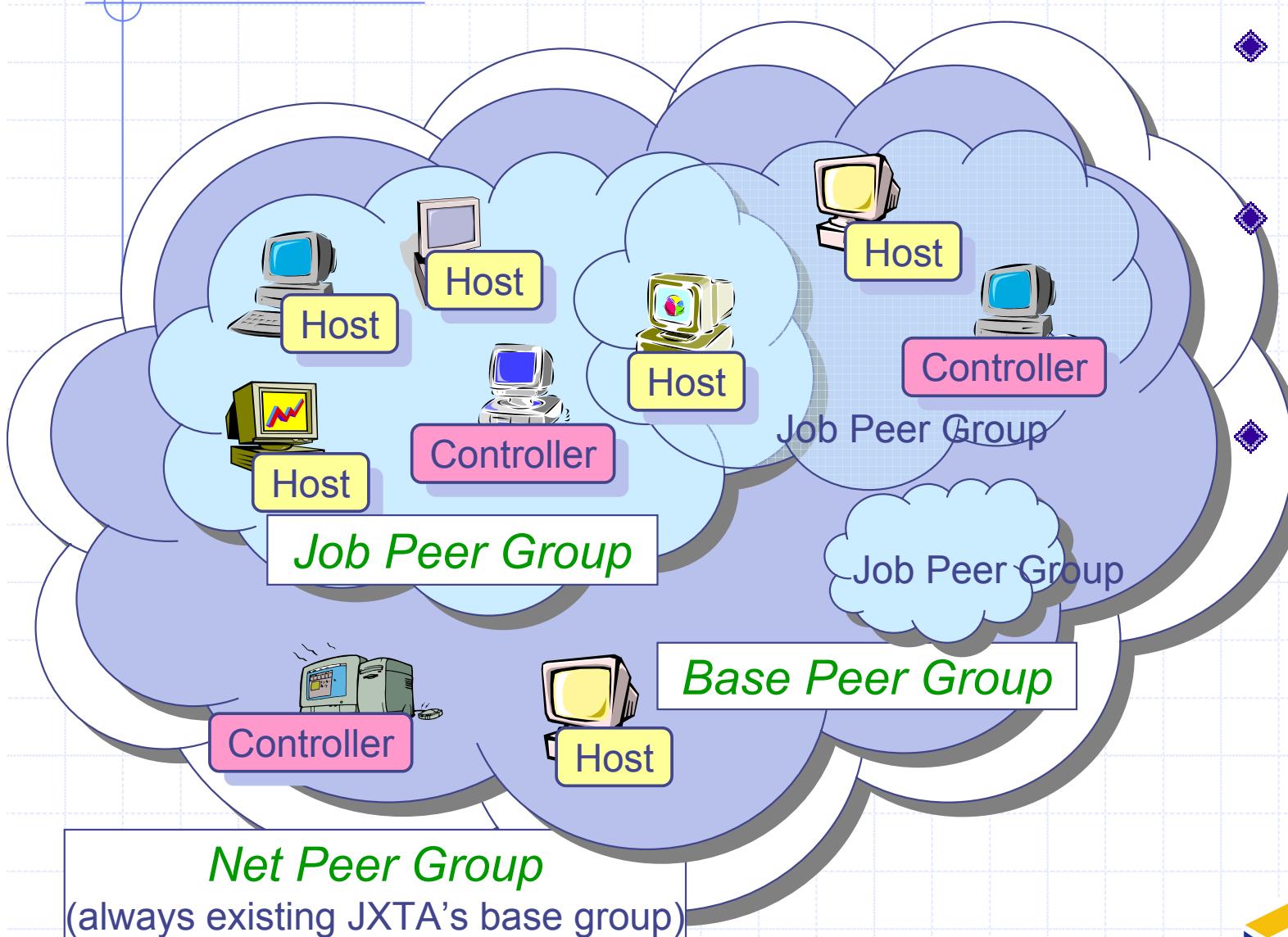
Last Update Time	2004/01/27 05:21:16
Elapsed Time	0:54:29
# of workers	8
# of WorkUnit	100000 [14493 done, 85507 remain]
progress	14.49%
Total CPU Time	4:14:24
Calculation Speed	0.23 [sec/WU], 15500 [WU/hour]
Estimated Finish Time	2004/01/27 05:21:16 +0900 (remain 5:30:59)

## Job Information

## Host Information

Rank	<a href="#">name</a>	<a href="#">Result Received [WU]</a>	<a href="#">CPU Time</a>	<a href="#">Average Calc Speed [sec/WU]</a>	<a href="#">Fastest Time per WU [sec]</a>	<a href="#">Average Calc Time [sec/WU]</a>	OS	CPU
1	<a href="#">sumomo11-shudo</a>	2711	0:54:29	1.24	0.76	1.21		i386
2	<a href="#">sumomo12-shudo</a>	2653	0:50:35	1.27	0.82	1.14		i386
3	<a href="#">sumomo14-shudo</a>	2523	0:36:18	1.33	0.76	0.86		i386
4	<a href="#">sumomo13-shudo</a>	2172	0:40:33	1.55	0.77	1.12		i386
5	<a href="#">sumomo15-shudo</a>	1916	0:29:21	1.76	0.77	0.92		i386
6	<a href="#">sumomo07-shudo</a>	1262	0:23:42	2.67	0.83	1.13		i386
7	<a href="#">sumomo06-shudo</a>	837	0:12:57	4.02	down?	0.82		i386
8	<a href="#">sumomo05-shudo</a>	419	0:06:25	8.03	0.82	0.92		i386

# How Peer Group is utilized



- ◆ **Net Peer Group**
  - A PG always exists in a JXTA apps.

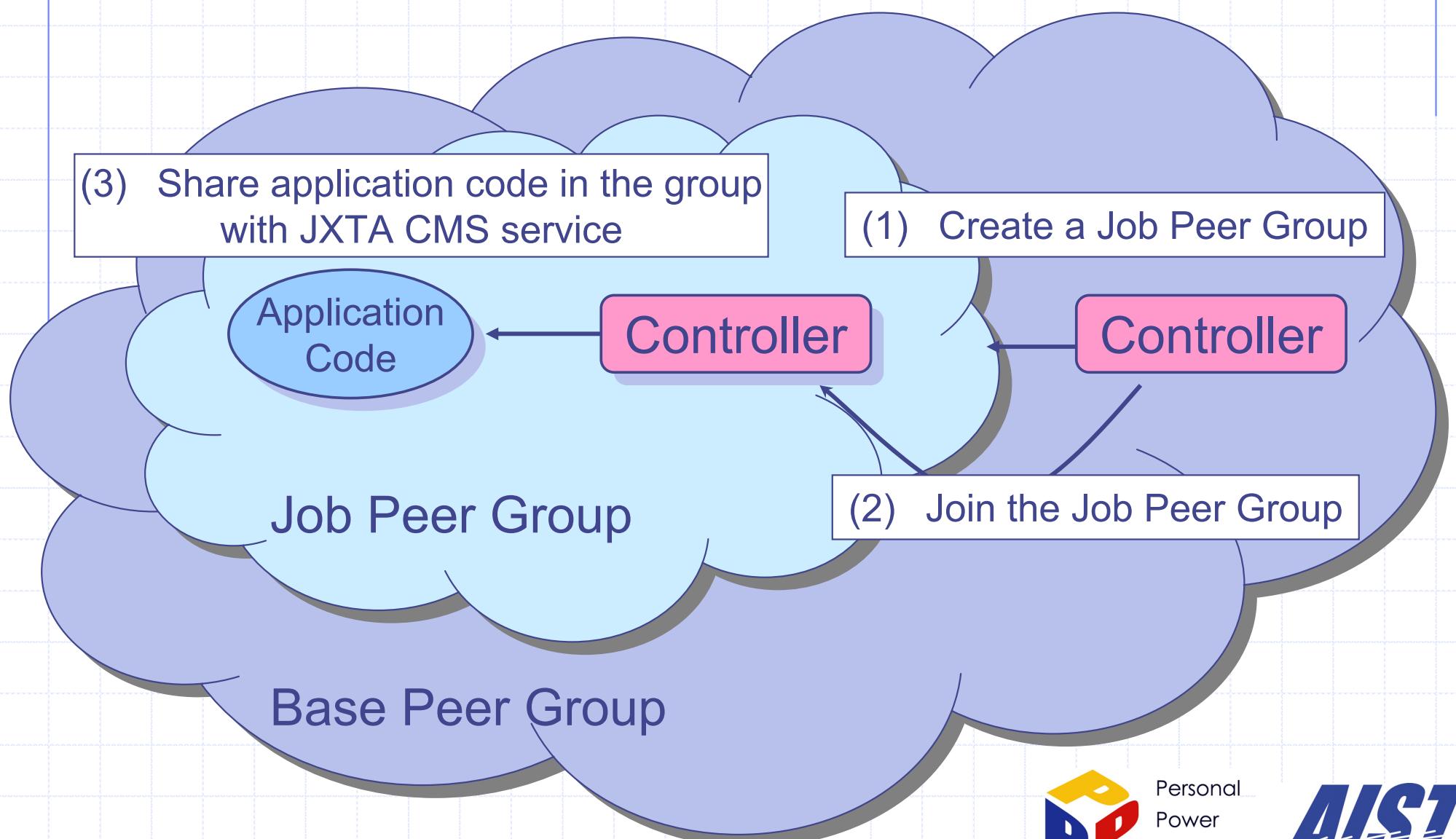
## Base Peer Group

- A PG for P3.
- All Hosts and Controllers join this PG first.

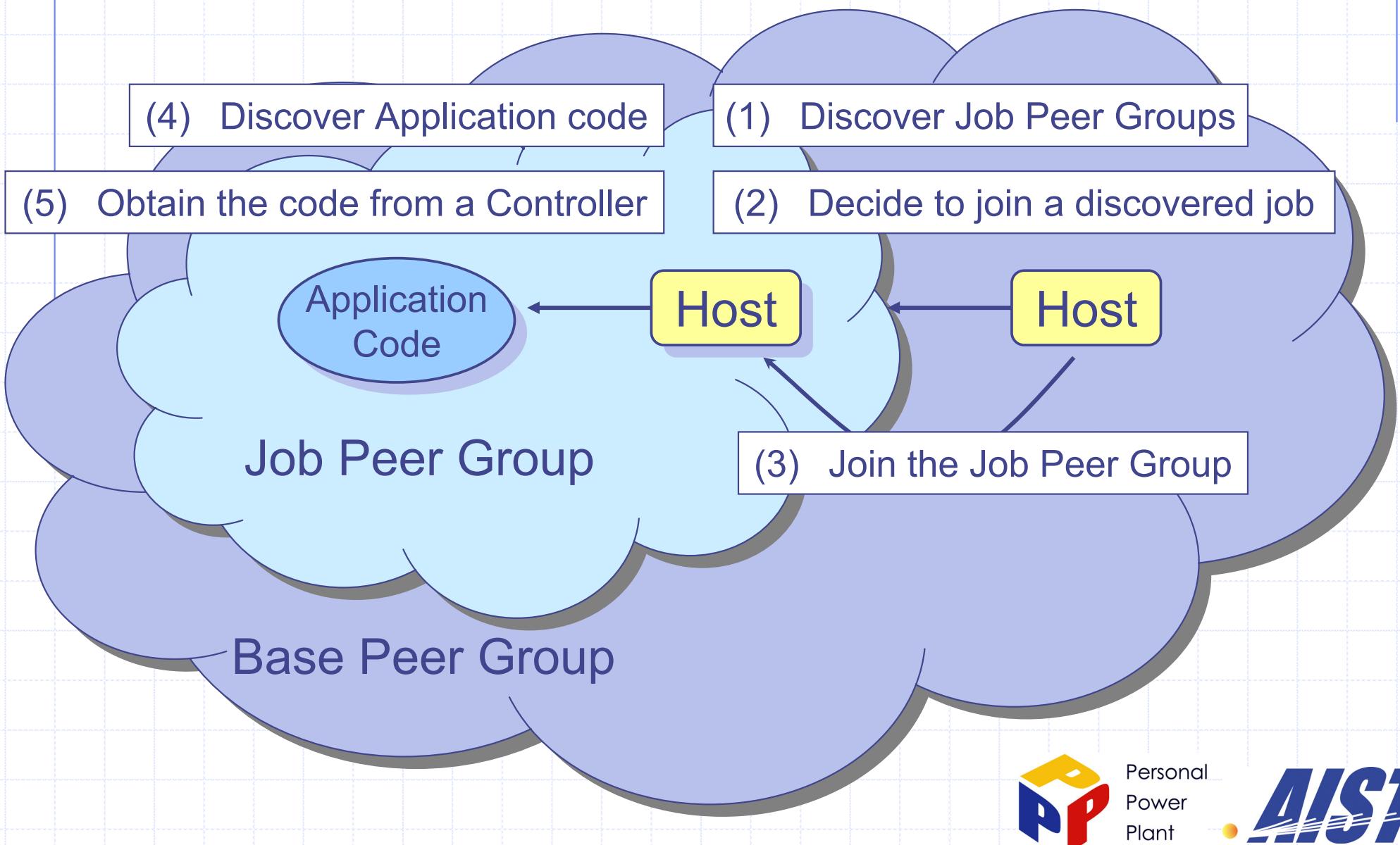
## Job Peer Group

- A PG for each job.
- All job-related comm. are performed in this PG.
  - ◆ Job control
  - ◆ Parallel processing

# Job Submission by Controller

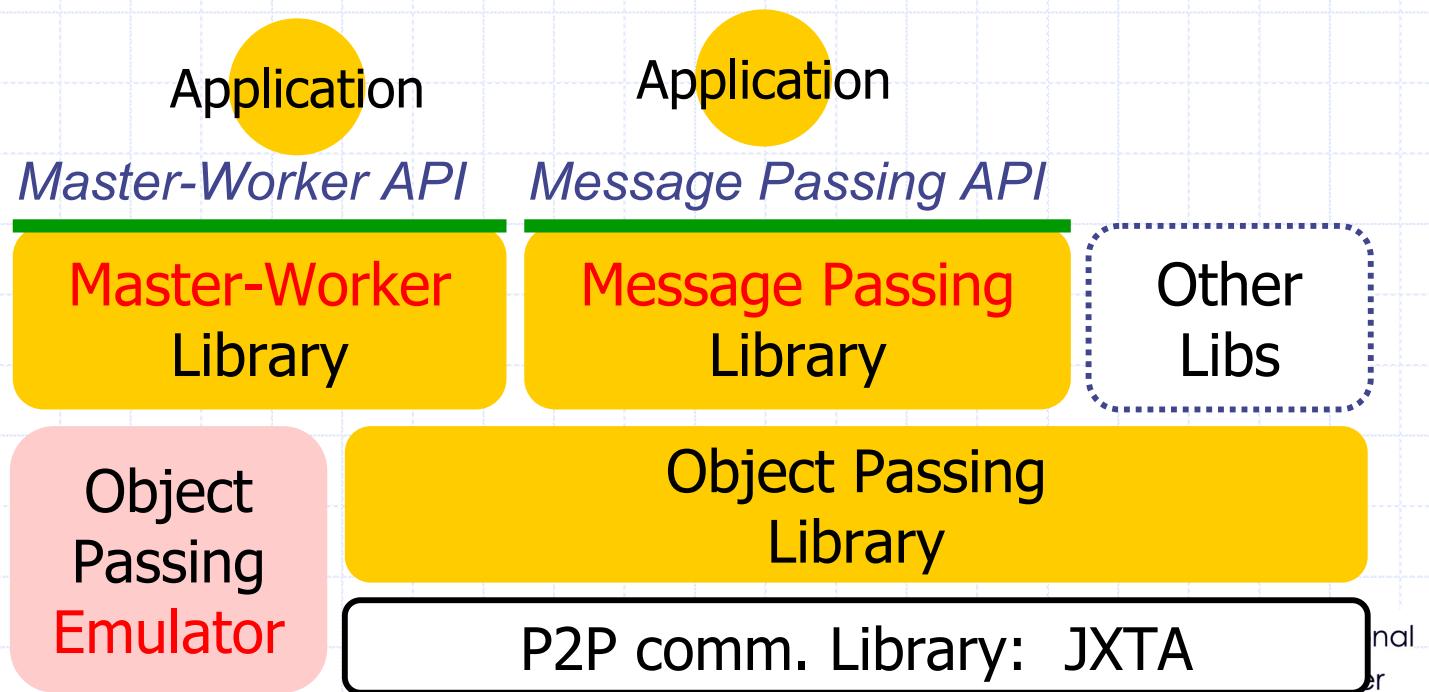


# Participation in a Job



# Parallel Programming Libraries

- ◆ Application programmers can use 2 libraries:
  - Master-worker
  - Message passing (like MPI)
- JXTA-MPI
- ◆ Emulator
  - enables us to run parallel apps on one PC.
  - It is extremely useful to test and debug the application in advance of real deployment.



# Performance Evaluation

- ◆ JXTA provides a rich set of functions, but... Isn't it slow?
  - Certainly, not fast. But enough for many cases.
- ◆ Performance measurements:
  - Basic communication performance
    - ◆ Latency and throughput
  - Application
    - ◆ RC5 attack
- ◆ Environments:
  - 2.4 GHz Xeon PCs, Gigabit Ethernet
  - Linux 2.4.19, Java 2 SDK 1.4.2, JXTA 2.1
  - Rich PC and network compared with today's Internet, but in which limits of P3 software can be measured clearly.



Personal  
Power  
Plant



# Communication Latency

- ◆ 1 byte round-trip communication.  
A one-way comm. takes
  - TCP (in C): 0.062 msec
  - TCP (in Java): 0.064 msec
  - P3's Message passing: 4.5 msec

- ◆ Not fast
  - It can limit the number of workunits that a master can process. One workunit takes several milliseconds.
  - Enough for many situations, but JXTA should be improved.

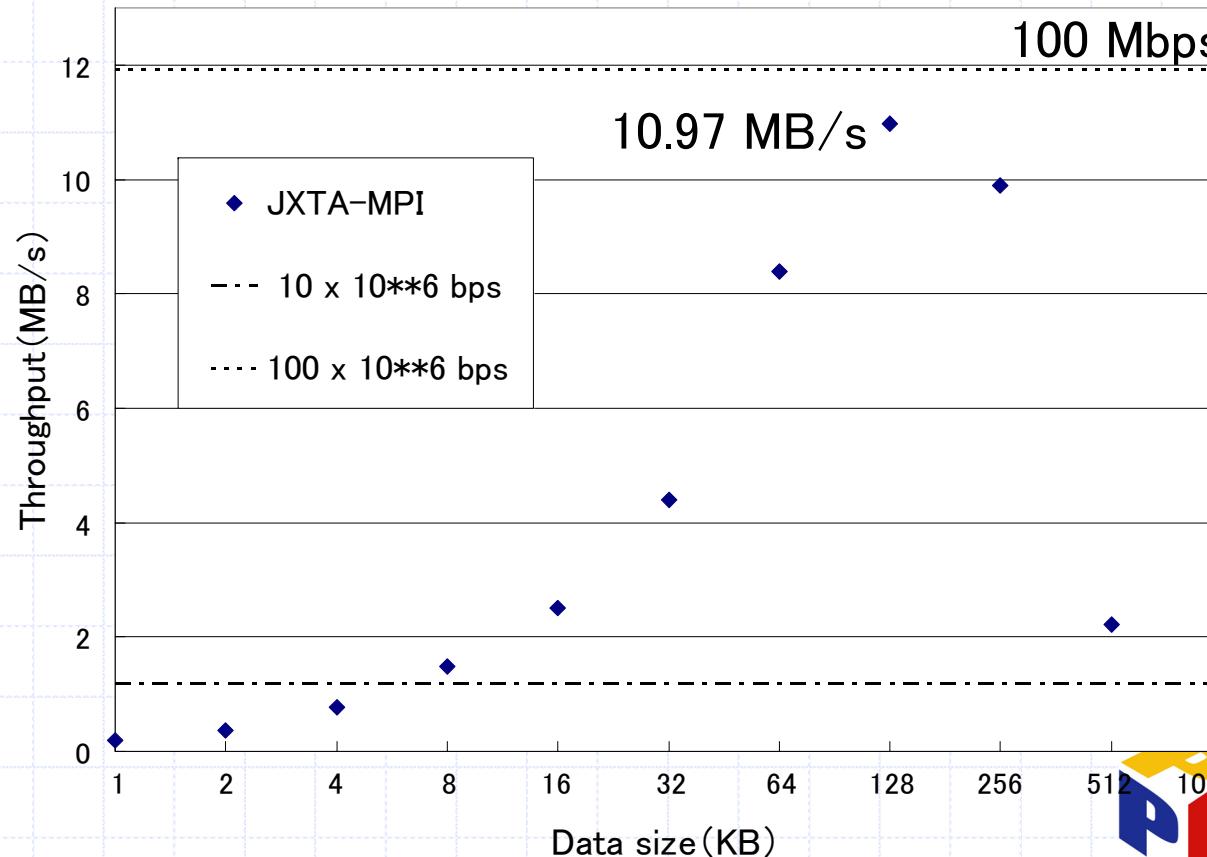


Personal  
Power  
Plant



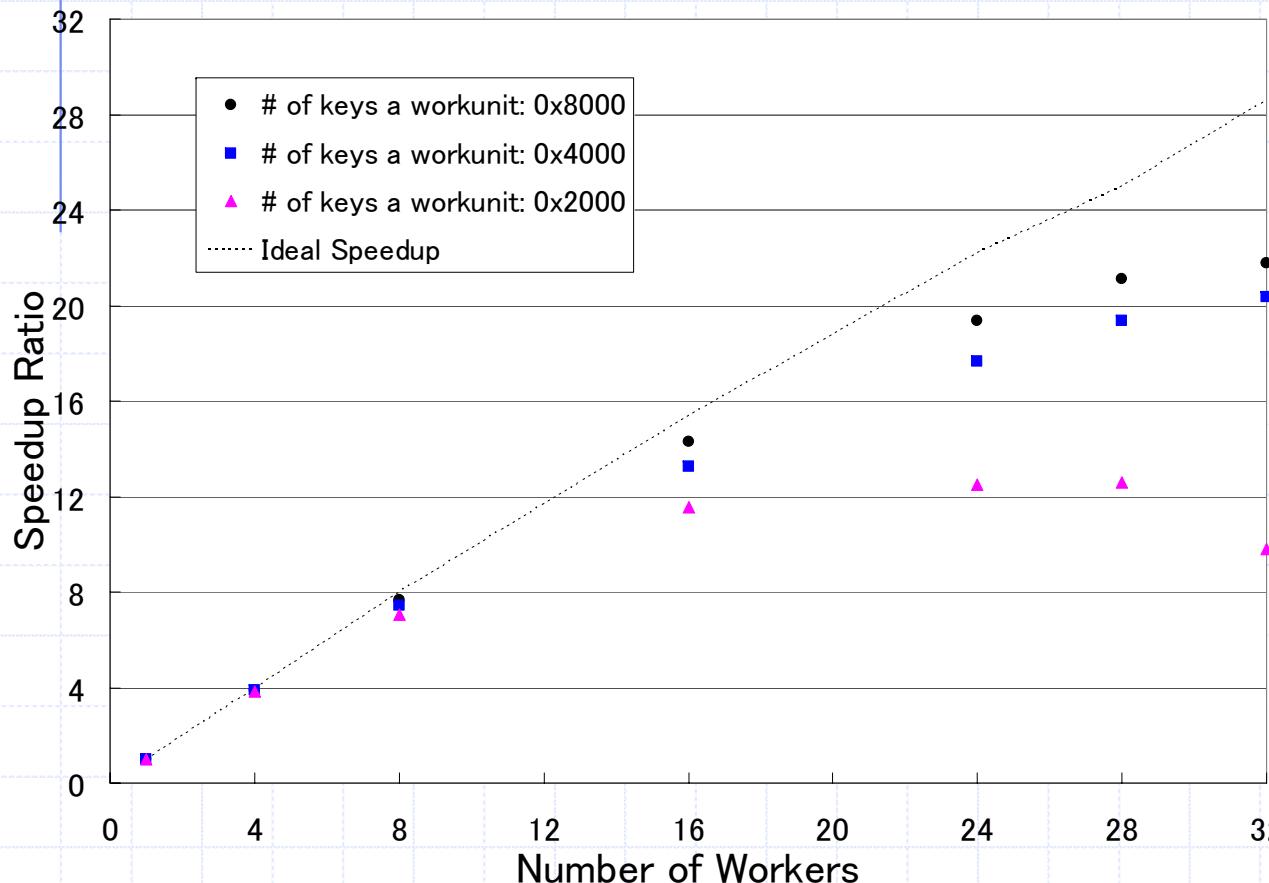
# Communication Throughput

- ◆ Message passing library is used.
- ◆ About 100 Mbps ( $100 \times 10^{**} 6$  bps).
  - Not very fast on Gigabit Ethernet, but P3 can fill Internet connections to small offices and homes.
- ◆ Throughput declines with larger messages.
  - Such a large message should be divided.



# Application Performance

- ◆ A load test with small workunits.
  - Brute-force attack on RC5 cryptosystem.  
same as distributed.net working on RSA RC5 challenge.
  - P3 is tolerant of such granularity of workunits (taking several seconds)  
with dozens of PCs.



- ◆ Granularity of workunit
  - 0x8000: 1.4 sec
  - 0x4000: 0.69 sec
  - 0x2000: 0.36 sec
- Very fine for a load test. Unusual for Internet-wide computation.

# Related Work

## ◆ JNGI

- being developed by Sun Microsystems.
- uses **JXTA**.
- utilizes peer groups to manage many PCs efficiently.
  - ◆ Groups of working peers are established in advance, and a submitted (parallel) job is assigned one of those groups.
  - ◆ cf. while P3 creates peer groups for each job.
- Though a paper has been published (in GRID 2002), most part of the idea has not been implemented (as of Sep 2003).

## ◆ XtremWeb, GreenTea, Javelin, Bayanihan, ...

- Of course, they have their own good points, but
- PC providers cannot choose application programs.
- Programming model is limited to master-worker or divide-and-conquer.
- Firewall are not considered.
  - ◆ use Java RMI, TCP and so on.
- Not tolerant of malicious PC providers or obscure

# Current Status and Future Plan

## ◆ Publicly released

- <http://p-three.sourceforge.net/>
- under Apache License 2.0
  - ◆ Like BSD license and imposes few constraints.
- in September 2004

## ◆ More applications and deployment

- P3 as an application hosting environment  
(Osaka University and NICT).



Personal  
Power  
Plant



KR-JP Grid Symposium, Seoul  
December 1-2, 2004

# Integration Patterns of P2P and Grid Technologies

Kazuyuki Shudo <[shudo@ni.aist.go.jp](mailto:shudo@ni.aist.go.jp)>,  
Grid Technology Research Center,  
AIST, Japan



Personal  
Power  
Plant



Grid  
Technology  
Research  
Center  
AIST



# Synergies between Grid and P2P

- ◆ It seems to be an intuitive feel.
- ◆ There have been activities including:
  - General and conceptual discussion
  - Introduction of activities: projects and implementations
  - Key capabilities of these technologies
  - Comparison in major aspects
- ◆ GGF9 workshop (Oct 2003)
  - Peer-to-Peer and Grids: Synergies and Opportunities
  - <http://www-csag.ucsd.edu/P2P-Grid/>
- ◆ APAN and Internet2 joint meeting (Jan 2004)
  - P2P and GRID: Convergence and Challenges
  - <http://apan.net/meetings/honolulu2004/ws-application.htm#appl3>
- ◆ This talk is a survey of instances of Grid/P2P integration
- ◆ To extract integration patterns

# Grid/P2P Technologies and their Integration

- ◆ Technologies and characteristics regarded as belonging to P2P
  - Resource discovery
    - ◆ Ad-hoc and dynamic grouping/matching
  - Network overlay
    - ◆ Logical ID, NA(P)T traversal, multi-protocol support
  - Scalability, Fault-tolerance, low management cost, ...
- ◆ Grid
  - Resource aggregation
    - ◆ Cluster of computers, global filesystem, ...
  - Scheduling, Resource assignment
  - Parallel/distributed processing
  - (PKI-based) authentication/authorization

# Preliminary Classification toward Taxonomy

## ◆ Instances

- JNGI, P3
  - ◆ Distributed computing system based on JXTA
- P2P Grid Information Service
  - ◆ P2P GIS (ISTI-CNR, Italy), GAIS (KISTI)
- Sun ONE Grid Engine (SGE) + JXTA
  - ◆ SGE is a job management software for PC/WS cluster like PBS and Platform's LSF
- P2P Grid system + Grid middleware
  - ◆ cf. Community Scheduler Framework (CSF): OGSI-compliant meta-scheduling framework built upon GT3
    - Platform's LSF + Globus Toolkit 3.x (GT3)
    - United Devices' Grid MP was planned to be compliant with the CSF ?

## ◆ Patterns

- P2P distributed computing, Desktop Grid
- P2P over Grid
- Grid over P2P



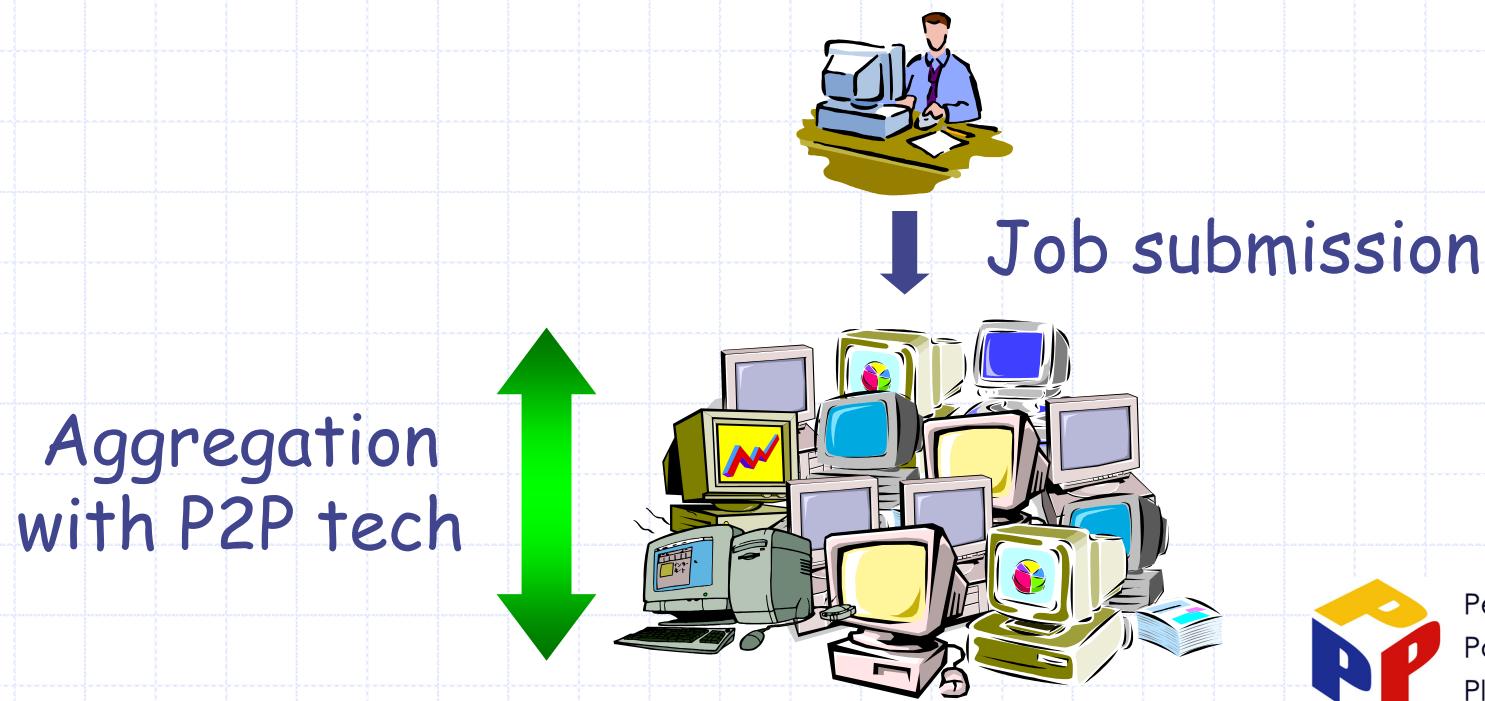
Personal  
Power  
Plant



# P2P distributed computing

◆ Aggregate computational resources with P2P technologies for distributed computation

- ex. JXTA-based dist. comp. middleware
  - ◆ JNGI, P2P
- (XtremWeb, SETI@home, distributed.net, ...)



Personal  
Power  
Plant

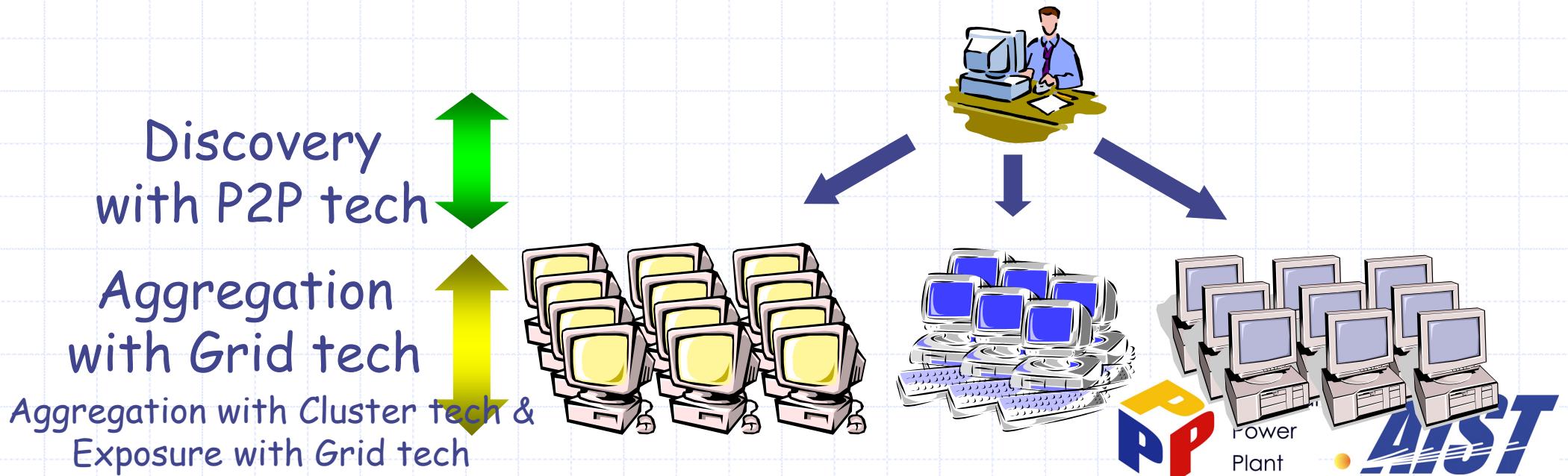


# P2P over Grid

◆ Aggregate resources with Grid technologies  
and discover the resources with P2P  
technologies

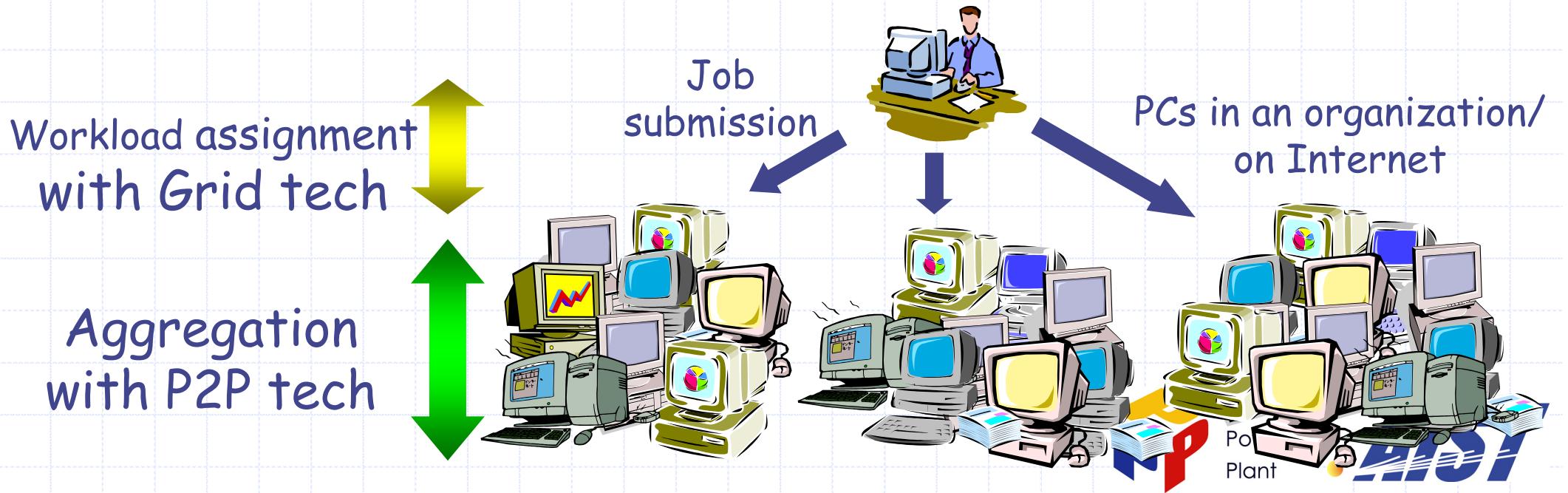
◆ ex.

- ◆ P2P-based Grid Information Service
  - P2P GIS (ISTI-CNR, Italy), GAIS (KISTI), ...
- ◆ Sun One Grid Engine (SGE) + JXTA

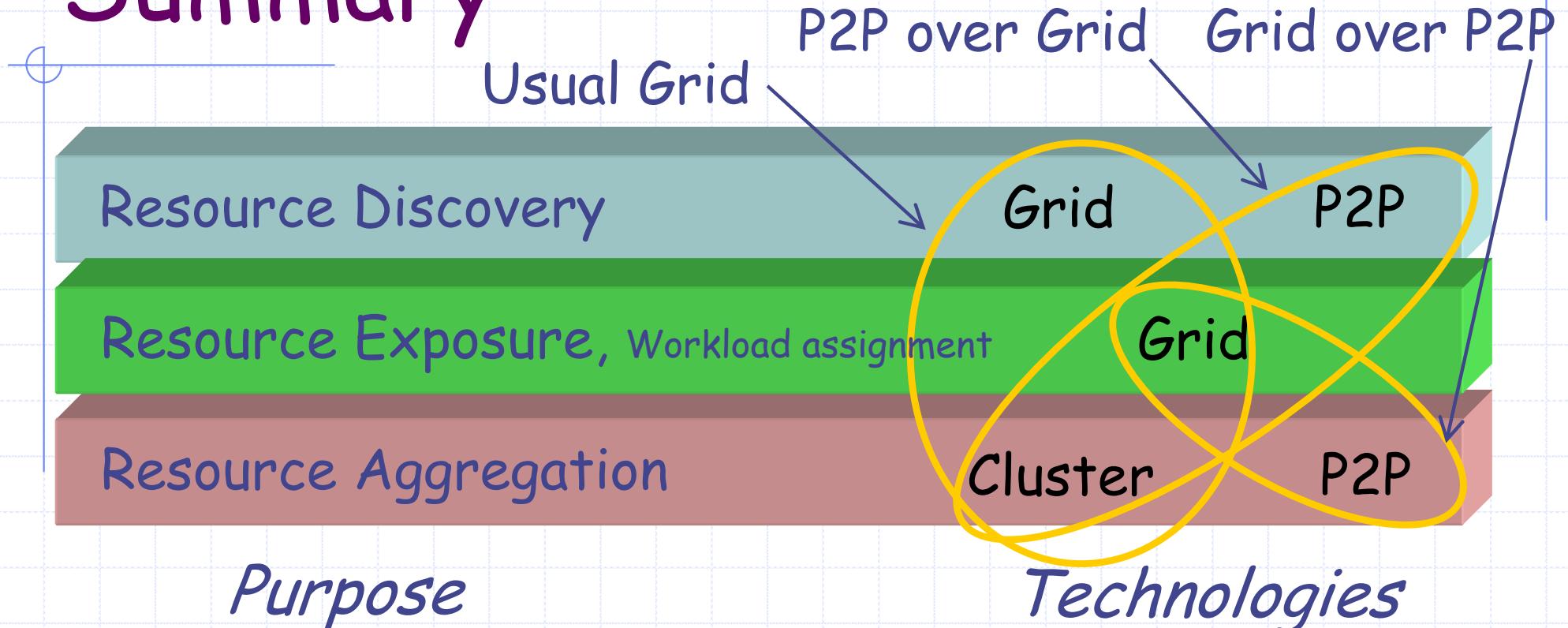


# Grid over P2P

- ◆ Aggregate resources with **P2P technologies** and assign workloads/data onto the resources with **Grid technologies**
  - Secure co-allocation
- ◆ ex.
  - United Devices' Grid MP + Globus Toolkit



# Summary



- ◆ Other ways of classification?
- ◆ Other integration pattern? tech application?
- ◆ New value with unexplored patterns?