17th APAN Meetings,
Application Tech. Workshop:
P2P and Grid: Convergence and Challenges

P3: Personal Power Plant
Makes over your PCs into power generator on the Grid

Kazuyuki Shudo <shudo@ni.aist.go.jp>,
Yoshio Tanaka,
Satoshi Sekiguchi
Grid Technology Research Center, AIST, Japan
P3: Personal Power Plant

Middleware for distributed computation

Traditional goals
- Cycle scavenging
  - Harvest compute power of existing PCs.
- 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?
- Circulation of computational resources
  - Transfer individual resources (C2C, C2B) and also aggregated resources (B2B).
  - Commercial dealings need a market and a system supporting it.
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 to promote the development of economy.
  - A Protein-Folding application is working on P3 and we test practical use of P3.

- **Scalable**
  - Of course 😊
  - 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.
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).
Design Goals (cont'd)

- **NA(P)T and firewall traversable**

  - **P3 uses JXTA for all communications.**
    - JXTA is a widely accepted P2P protocol, project and library that provides common functions P2P software requires.
    - JXTA enables bidirectional communication over NA(P)T and many kinds of firewall (incl. unidirectional HTTP only FW).
      - **P3** provides message-passing API for parallel programming besides master-worker API.

  - **Other aims in adopting 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 feature.
    - **Multi-protocol:** JXTA relay peers mediate messages between TCP, HTTP, IP multicast and possibly other protocols like Bluetooth.
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.
Design Goals (cont’d)

易部署和自动更新

- 安装和更新的劳动量与PC的数量成正比，可能非常巨大。
- 软件无法自动更新时，易受攻击的客户端软件会大部分保持原状。
  - 2003年4月，SETI@home客户端软件中发现了一个漏洞。
- P3可以通过点击网页上的鼠标点击来安装，并且可以自动更新。
  - cf. Java Web Start (JWS)
Structure of P3

**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.
  - **Job monitor**: shows a state of a job and attending Hosts.

**Parallel programming libraries**
- Application programs that use these libraries can run on P3.
  - Master-worker
  - Message Passing (like MPI)
Job Management Subsystem: Controller

A resource user submits and control jobs with Controller.
Job Management Subsystem: Host

A daemon program runs on a provided PC.

- A Host can be invoked in a head(GUI)-less mode. In that case, it 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

Total view

Number of processed workunits

Calculation speed

Host view
### Host Information

<table>
<thead>
<tr>
<th>Rank</th>
<th>name</th>
<th>Result Received [WU]</th>
<th>CPU Time</th>
<th>Average Calc Speed [sec/WU]</th>
<th>Fastest Time per WU [sec]</th>
<th>AverageCalc Time [sec/WU]</th>
<th>OS</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sumpemo11 - ekudo</td>
<td>2711</td>
<td>0:34:29</td>
<td>1.24</td>
<td>1.09</td>
<td>1.21</td>
<td>i386</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>sumpemo12 - ekudo</td>
<td>2653</td>
<td>0:50:35</td>
<td>1.27</td>
<td>0.82</td>
<td>1.14</td>
<td>i386</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>sumpemo14 - ekudo</td>
<td>2523</td>
<td>0:36:18</td>
<td>1.33</td>
<td>0.76</td>
<td>0.86</td>
<td>i386</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>sumpemo13 - ekudo</td>
<td>2172</td>
<td>0:40:33</td>
<td>1.55</td>
<td>0.77</td>
<td>1.12</td>
<td>i386</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>sumpemo15 - ekudo</td>
<td>1916</td>
<td>0:29:21</td>
<td>1.76</td>
<td>0.77</td>
<td>0.92</td>
<td>i386</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>sumpemo07 - ekudo</td>
<td>1262</td>
<td>0:23:42</td>
<td>2.67</td>
<td>0.83</td>
<td>1.13</td>
<td>i386</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>sumpemo06 - ekudo</td>
<td>837</td>
<td>0:12:57</td>
<td>4.02</td>
<td>0.82</td>
<td>0.93</td>
<td>i386</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>sumpemo05 - ekudo</td>
<td>419</td>
<td>0:06:25</td>
<td>8.03</td>
<td>0.82</td>
<td>0.92</td>
<td>i386</td>
<td></td>
</tr>
</tbody>
</table>
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

1. Create a Job Peer Group
2. Join the Job Peer Group
3. Share application code in the group with JXTA CMS service
Participation in a Job

(1) Discover Job Peer Groups

(2) Decide to join a discovered job

(3) Join the Job Peer Group

(4) Discover Application code

(5) Obtain the code from a Controller
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.
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.
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.
Communication Throughput

- Message passing library is used.
- About **100 Mbps** \((100 \times 10^6 \text{ bps})\).
  - 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.
A load test with small workunits.
- Brute-force attack on RC5 cryptsystem.
  same as distributed.net working on RSA RC5 challenge.
- P3 is tolerant of such granularity of workunits (taking several seconds)
  with dozens of PCs.

Workunit processing time:
- 0x8000: 1.4 sec
- 0x4000: 0.69 sec
- 0x2000: 0.36 sec

Very small. Unusual for Internet-wide computation.
Related Work

- **JNGI**
  - being developed by Sun Microsystems.
  - uses JXTA.
  - utilizes peer groups to manage many PCs efficiently.
    - 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.

- **XtremWeb, GreenTea, Javelin, Bayanihan, ...**
  - 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.
Future Plan

- Public release
  - 2Q 2004 planned

- Test with more PCs
  - Several hundreds or more PCs
  - with AIST super cluster?
    - Having over 1000 PCs

- Write a paper
  - A Japanese paper will be accepted, but