4VP+: A Novel Meta OS Approach for Streaming Programs in Ubiquitous Computing

Yaoxue Zhang, Yuezhi Zhou
Tsinghua University
Contents

1. Introduction
2. Overview of Transparent Computing
3. Concept and architecture of Meta OS
4. Program streaming through 4VP+
5. System implementation & evaluation
6. Conclusions
1. Introduction

1.1 Change of Computing Paradigms

- **Mainframe Computing**
  - T/M

- **Desktop-based Network Computing**
  - C/S \( \rightarrow \) C/N \( \rightarrow \) C/VE

- **Pervasive Computing (Ubiquitous Computing)**
  - Everything is a Computer
  - Technologies disappear

Timeline:
- \(~1970’s\)
- \(~2000’s\)
- 21st Century
1.2 The Characteristics of Ubiquitous Computing

Remote communication
Fault tolerance
High Availability
Remote information access
Distributed security

Mobile networking
Mobile information access
Adaptive applications
Energy-aware systems
Location sensitivity

Distributed Systems
Mobile Computing
Pervasive Computing

Smart spaces
Invisibility
Localized scalability
Uneven conditioning

(Personalized Service (Active Service))

(From M. Satyanarayanan, 2001)
1.3 The concept of traditional stored program

“store its instructions in its internal memory and process them in its arithmetic unit, so that in the course of a computation they may be not just executed but also modified at electronic speeds.”

By Von Neumann 1945

(From “The stored program concept”, Aspray. W. IEEE 1990)
1.4 Some problems to realize Ubiquitous Computing in traditional computer systems

- Users can not choose different OSes and applications in the same hardware platform
- Regular OS can not be installed on light-weight devices, Embedded OS is not so sensitive to needs of users
2. Overview of Transparent Computing

2.1 Transparent Computing Paradigm

- Users do not care the technical details of the computing; they only focus on the service provided!
- Users can select services based on heterogeneous OSes from the same hardware platform
- **Execution and Storage of Programs are separated in different computers**

**Execution:** light-weight devices or clients
(as assembling factory)

**Storage:** Servers (as warehouse)
→ JITC (Just In Time Computing)
2.2 The D&T extended Von Neumann architecture

- The basic considerations:
  - Users only need services
  - Why users can not get smart services?
    - Storage and execution have not been separated in traditional computers, so that the computers become more complex with applications increasing

- Separated Storage and execution of programs into different computers:
  - Server computer: Storage and management
  - Light-weight client: I/O and execution
  - Network: connecting servers and clients (Similar to bus in traditional computers)
  - Program streaming: block-streaming between server and clients according to interruption or I/O requirement
2.3 Topology of Transparent Computing Paradigm

**TC**
(Transparent Clients)

**TDN**
(Transparent Delivery Network)

**TS**
(Transparent Servers)

Light-weight devices:
- PC, PDA, Intelligent Mobile Phone, Digital Appliance, Dedicated Clients, et al.
- Ethernet, CATV, 802.11, IEEE 1394, et al.
  (Ubiquitous Communication)

Regular PC, PC Server, et al.
3. Concept and architecture of Transparent Computing

3.1 Execution environment of Meta OS

- **Instance OS:** Traditional commercial OS running on traditional computers, such as Linux, Windows, etc.
- **Meta OS:** A program to control Instance OSes.

- Users choose instance OSes & Applications Using light-weight devices.
- Meta OS schedules services to Clients to execute.
3.2 Layered Architecture of Meta OS environment

- Application Layer
- Heterogeneous OS Layer
- Meta OS Layer (VDMS, NSAP, MRBP, etc)
- TDN Layer

- JITC Layer (Non-Resident)
- Meta OS
- MRBP
- TDN Layer

TS

Resident Module

TC

Non-Resident Module
4. Programs Streaming through 4VP+

4.1 The Modules of Meta OS

4VP+ System

- VUM (Virtual User Management)
- VFM (Virtual File Management)
- VDM (Virtual Disk Management)
- VIOM (Virtual I/O Management)
- MRBP (Multi-OS Remote Booting Protocol)
- NSAP (Network Service Access Protocol)

Server Main Board

4VP
(Non-Resident)

MRBP

Client Main Board

TS

Network

TC
4.2 MRBP: Start Delivery and Booting of OSes

TS

Waiting

If Received the signal from a client

Search the Boot blocks

Send the Boot block numbers to Client

Wait for user Choose for Default Start

Start NSAP

Search and delivery Meta OS

......

TC

Start

Get Start Interruption Signal

Send Message to Servers to Search Oses Boot blocks

If get the numbers of blocks, then show to user

If User Choose a OS

Default

Start NSAP to get Meta OS

Page Scheduling of instance OS

instance OS Running
4.3 Network Service Access Protocol & Virtualization

TS

TC

4VP System

Client Management Database

4VP Management Database

Client Request Handler

VDisk Images

NSAP

Running in Memory

File Redirector

VIOM Management

VFile System

Virtual Disk

VDisk Driver
5. System implementation & evaluation

5.1 Implementation Using C/S Model

TS

NAT

TC

TC

TC

NAT: Network Address Translation

Internet
TC:

- **Low Power:** <=15w
- **X86 Architecture:** Support Multimedia Instructions, and Windows, Linux
- **One-board Synthetic Design:** MPEG1, 3D/2D Graphical accelerator, IEEE1394, Ethernet, USB, TV-out, Fax/Modem, I/O, etc.
- **Low Cost:** about $100
5.2 Performance Analysis

Testbed

- **Hardware Configuration**
  - **TC** (30 sets)
    - Intel Celeron 1 GHz with 128 MB RAM
    - Onboard network card: 100 Mbps
  - **TS** (3 sets)
    - AMD Athlon64 3000+ with 2 GB RAM
    - Hard Disk: Seagate SATA 7200 RPM, 2*80 G (RAID 0)
    - Onboard network card: 1 Gbps
  - **TDN**
    - Huawei-3Com Ethernet switch with 48 100 Mbps interfaces (for client) and 2 1 Gbps interfaces (for server)

- **Software Configuration**
  - **TS**
    - Windows 98, Windows 2000, Linux and other application software
  - **TC**
    - No OS and applications
**Booting Performance**

**X-axis:** Number of TC  
**Y-axis:** Boot Time
NSAP Throughput

X-axis: Request Size
Y-axis: Throughput

Read, unbuffered in Windows
Write, unbuffered in Windows
Read, unbuffered in Linux
Write, unbuffered in Linux
## Function Evaluation

*(Start times in Windows 2000)*

<table>
<thead>
<tr>
<th>Applications/OS</th>
<th>1 PC</th>
<th>1 TC</th>
<th>10 TCs</th>
<th>28 TCs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Booting OS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows 2000 Server</td>
<td>53&quot;13</td>
<td>48&quot;73</td>
<td>70&quot;62</td>
<td>142&quot;57</td>
</tr>
<tr>
<td><strong>Office Applications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Word 2003</td>
<td>2&quot;23</td>
<td>1&quot;26</td>
<td>2&quot;28</td>
<td>11&quot;50</td>
</tr>
<tr>
<td><strong>Image processing applications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PhotoShop V7.0</td>
<td>13&quot;29</td>
<td>11&quot;08</td>
<td>16&quot;48</td>
<td>1'0&quot;51</td>
</tr>
<tr>
<td>Flash V6.0</td>
<td>18&quot;62</td>
<td>7&quot;16</td>
<td>31&quot;41</td>
<td>1'16&quot;56</td>
</tr>
<tr>
<td>3D MAX V8.0</td>
<td>29&quot;71</td>
<td>25&quot;68</td>
<td>34&quot;24</td>
<td>1'16&quot;56</td>
</tr>
<tr>
<td><strong>Copying files</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>28&quot;24</td>
<td>24&quot;33</td>
<td>49&quot;48</td>
<td>4'6&quot;99</td>
</tr>
<tr>
<td><strong>Playing multimedia</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Media Player</td>
<td>smoothly</td>
<td>smoothly</td>
<td>smoothly</td>
<td>smoothly</td>
</tr>
</tbody>
</table>
## Function Evaluation (Start times in Linux Redflag 4.1)

<table>
<thead>
<tr>
<th>Applications/OS</th>
<th>1 PC</th>
<th>1 TC</th>
<th>10 TCs</th>
<th>28 TCs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Booting OS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Linux 4.1</td>
<td>85&quot;67</td>
<td>58&quot;20</td>
<td>97&quot;72</td>
<td>153&quot;72</td>
</tr>
<tr>
<td><strong>Office Applications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIOffice</td>
<td>2&quot;03</td>
<td>1&quot;07</td>
<td>2&quot;35</td>
<td>10&quot;43</td>
</tr>
<tr>
<td><strong>Internet Explore</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Firefox</td>
<td>3&quot;67</td>
<td>2&quot;55</td>
<td>3&quot;89</td>
<td>10&quot;60</td>
</tr>
<tr>
<td><strong>Copying files</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20MB</td>
<td>12&quot;25</td>
<td>7&quot;80</td>
<td>18&quot;65</td>
<td>53&quot;15</td>
</tr>
<tr>
<td>50MB</td>
<td>33&quot;83</td>
<td>25&quot;44</td>
<td>53&quot;76</td>
<td>257&quot;79</td>
</tr>
</tbody>
</table>
Applications
6. Conclusions

- We proposed a new OS: Meta OS and Transparent Computing for Ubiquitous Computing

- Many issues need to be researched further more, e.g.:
  - How to support more OSes?
  - How to support more devices?
  - How to extend to more applications?
  - How to support different communication networks?
Thanks!
Q&A
Consider data and instruction as the same

- *Stored Program* has realized in the single computer system by now, stored data has realized in network.
- have not realized storage of instructions in network environment.