

Linux Cluster Architecture

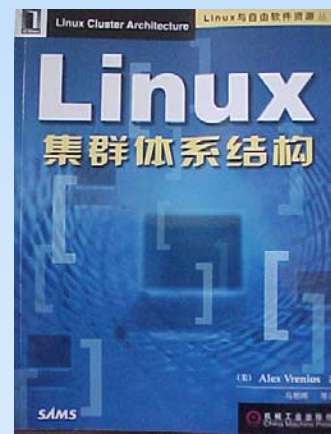
by

Alex Vrenios

(Shameless Plug)



English



Chinese

Slide # 1

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Overview:

- Why would anyone want to build a cluster system?
- Computer Architecture Review: UPs through Clusters
- Gathering the PC computer hardware (on the cheap!)
- Connecting the node computers into a local area network
- Local and remote software development file structure
- Configuring relevant Linux OS files for internetworking
- Subtasks and sockets for local or network programming
- The design of our master-slave cluster server software
- Internal and external performance monitoring and tuning

Slide # 2

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

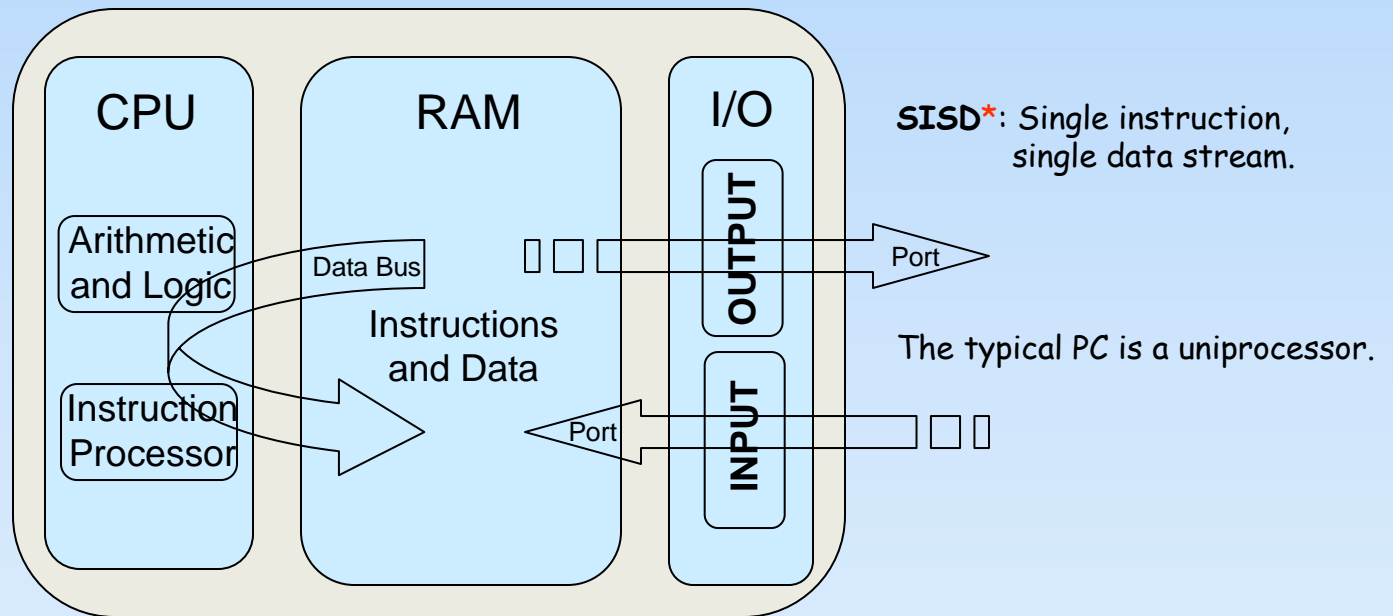
Why would anyone want to build a cluster system?

- Hobbyists:
It's a new and interesting pathway to experience! (And how many of your friends have a cluster server anyway?)
- Professionals:
Sophisticated systems are often developed in parallel, meaning the hardware won't be ready when you want to test your software. Having a test bed will get you past the hardware independent bugs, and put you in a position to polish your product when the platform is finally ready.
- Managers:
This is all bleeding edge stuff; you'll want to prepare for the issues your people might face and the questions they might ask. Experience gives you the insight you will need.
- Academics:
Analyze data from a live system, when you can, simulation can be over-simplified and its results can be misleading.

Linux Cluster Architecture

Computer Architecture Review:

- Uniprocessor or UP



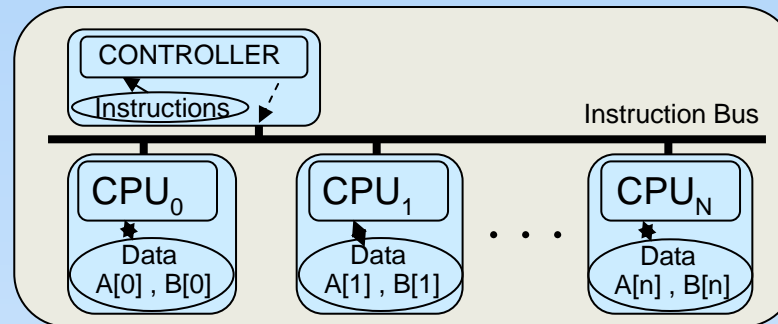
* Flynn proposed this taxonomy - some other configurations follow...

Slide # 4

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

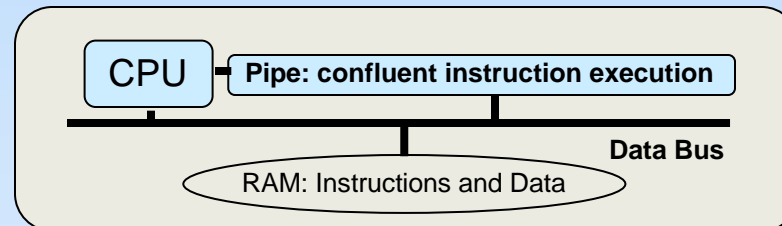
- **Array or Vector Processor**



SIMD: Single Instruction, multiple data stream.

(ILLIAC IV, IBM 390, DSPs, etc.)

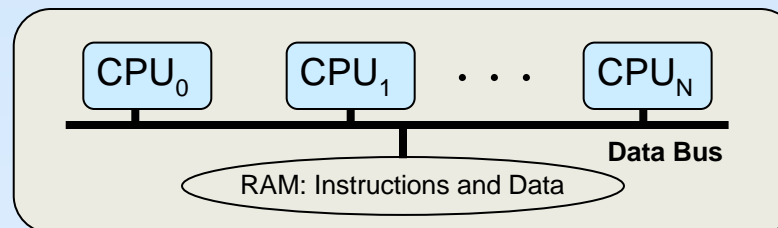
- **Pipeline Processor**



MISD: Multiple instruction, single data stream?

(Some say there is no MISD.)

- **Multiprocessor or MP**



MIMD: Multiple instruction, multiple data streams.

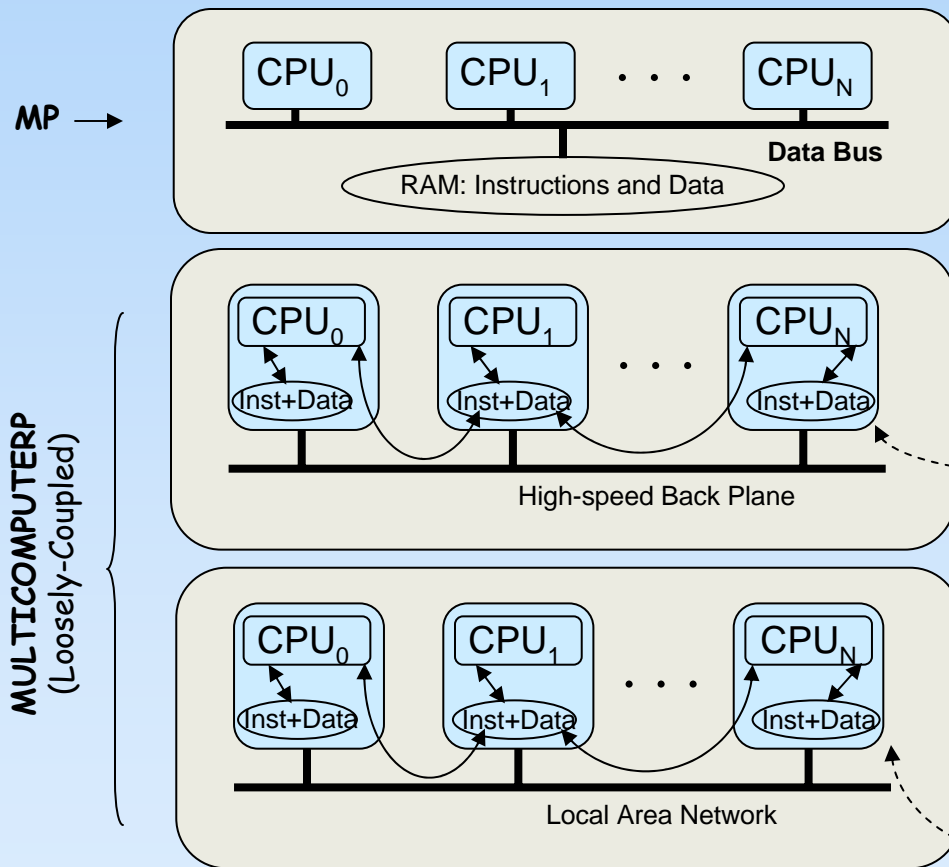
Mainframe, Workstation, etc.
(Mostly for the very wealthy!)

Slide # 5

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

- The MIMD is so interesting that gets its own taxonomy:



UMA: Uniform Memory Access

Tightly-coupled multiprocessor. All CPUs access instructions and data at the same transfer rate.

NUMA: Non-uniform memory access

VME Chassis, some Beowulf Clusters, and many embedded processor systems.

Plug-in boards, for example, each with a CPU and some local memory.

NORMA: No (hardware) Remote Memory Access (rare distinction)

Older Beowulf Clusters, Distributed Shared Memory Systems (IVY), and some Modern day cluster computers.

PCs, whose "personality" may be molded by its software!

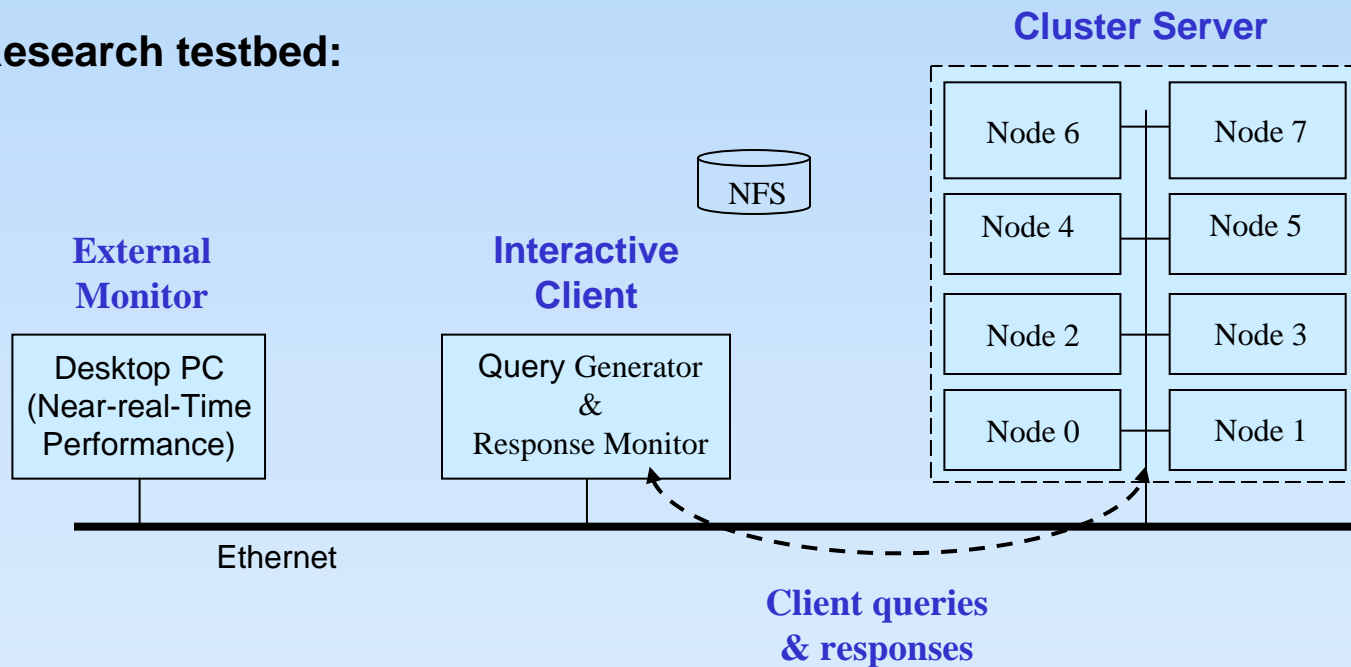
Slide # 6

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Network Block Diagram:

Research testbed:



Slide # 7

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

CHAOS: Cheap Array of Outmoded Systems:

- Finding and networking N personal computers:

4-way KV Switches
(may be optional)

Network Activity
Monitoring

Surge Protectors



Development System



Books!

and NFS File Server

Slide # 8

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Gathering PC Computer Hardware:

- Small computer stores (Renaissance Computer, e.g.)
- Newspaper and club and organization newsletter ads
- Family, friends and neighbors (closets, garage sales)
- Large corporations? (hospitals, Am Exp, Mot, etc.)
- Computer salvage outlets:

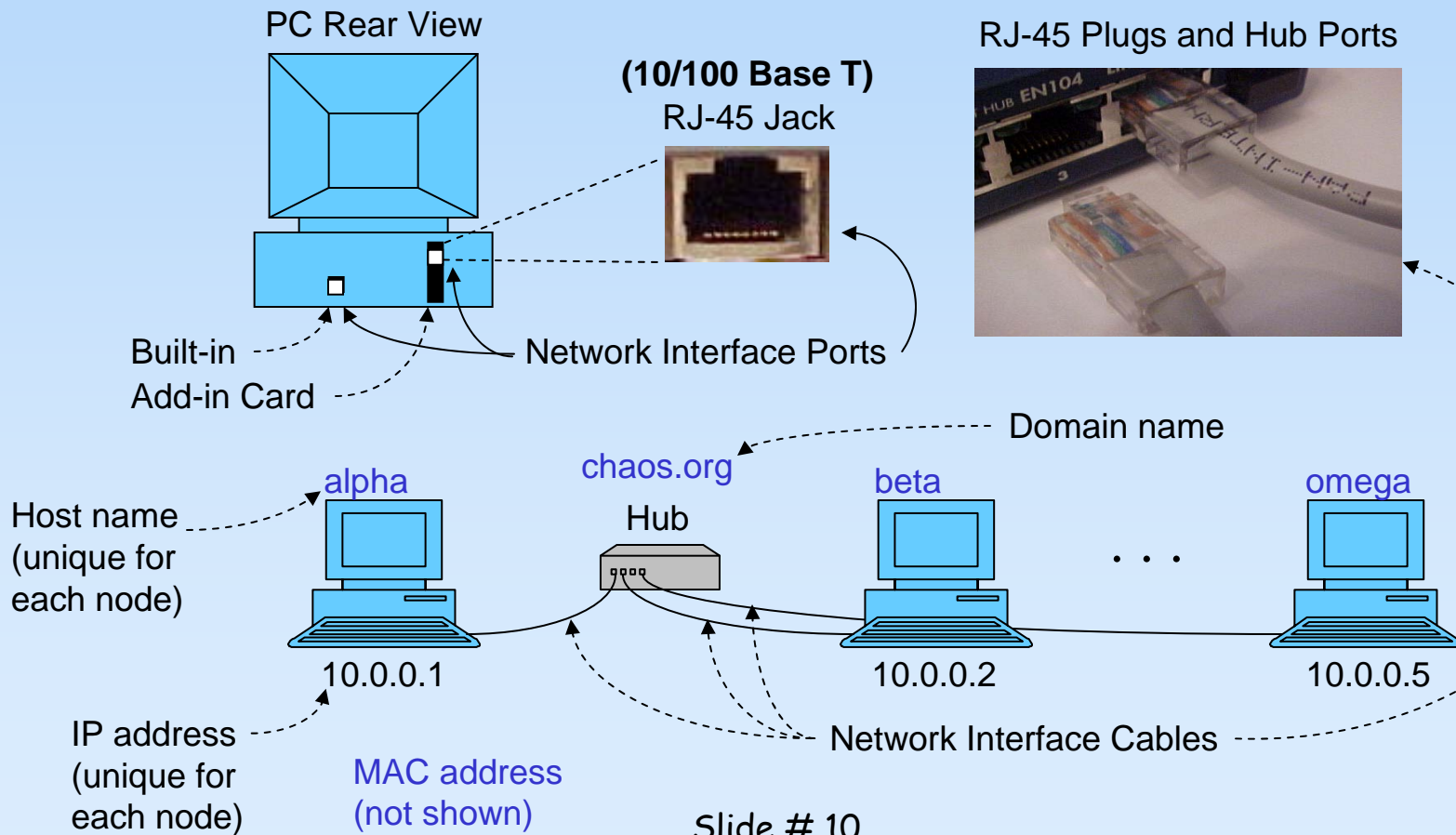


Slide # 9

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

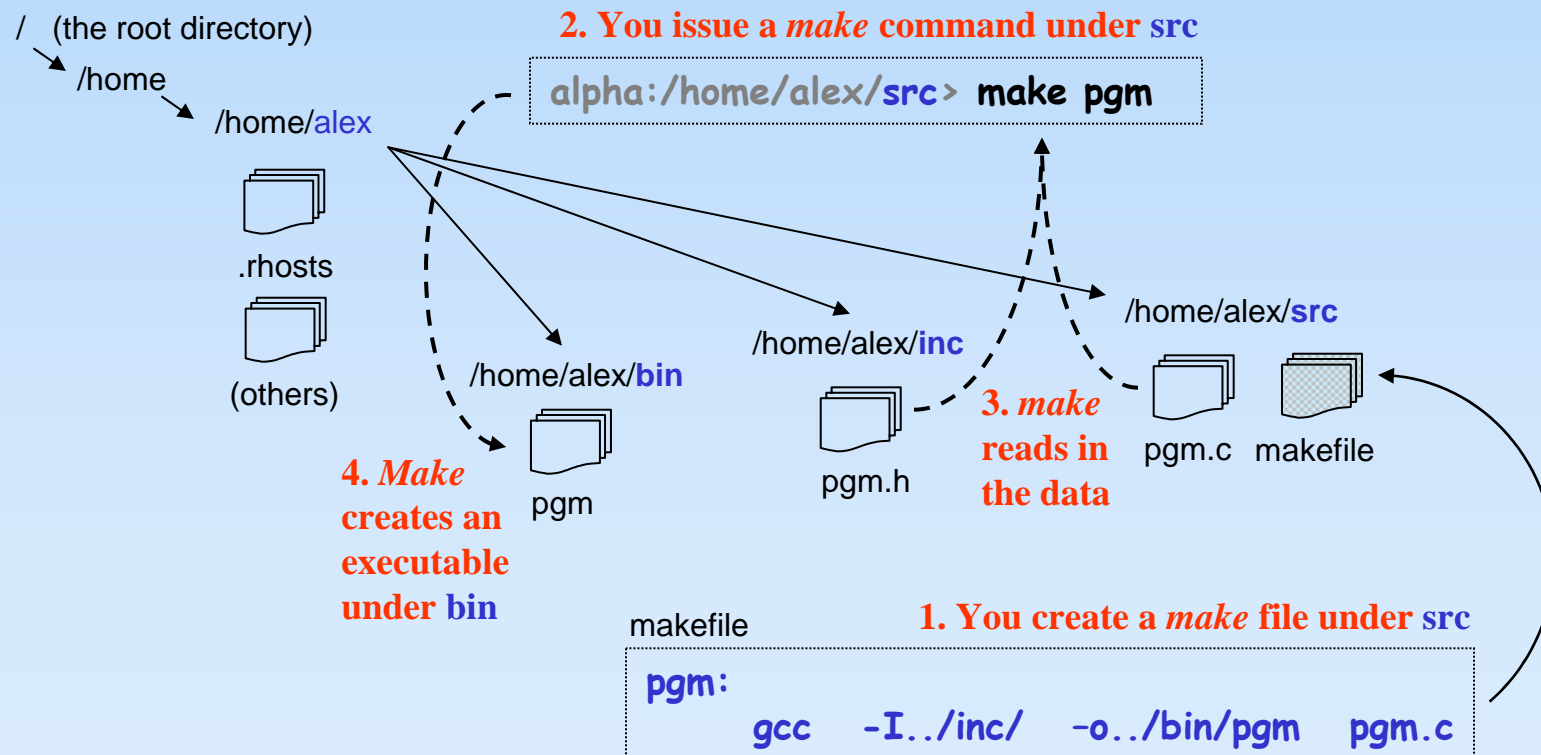
Connecting the Node PCs into a LAN:



Linux Cluster Architecture

Local User File Structure:

One way to organize your files for software development:



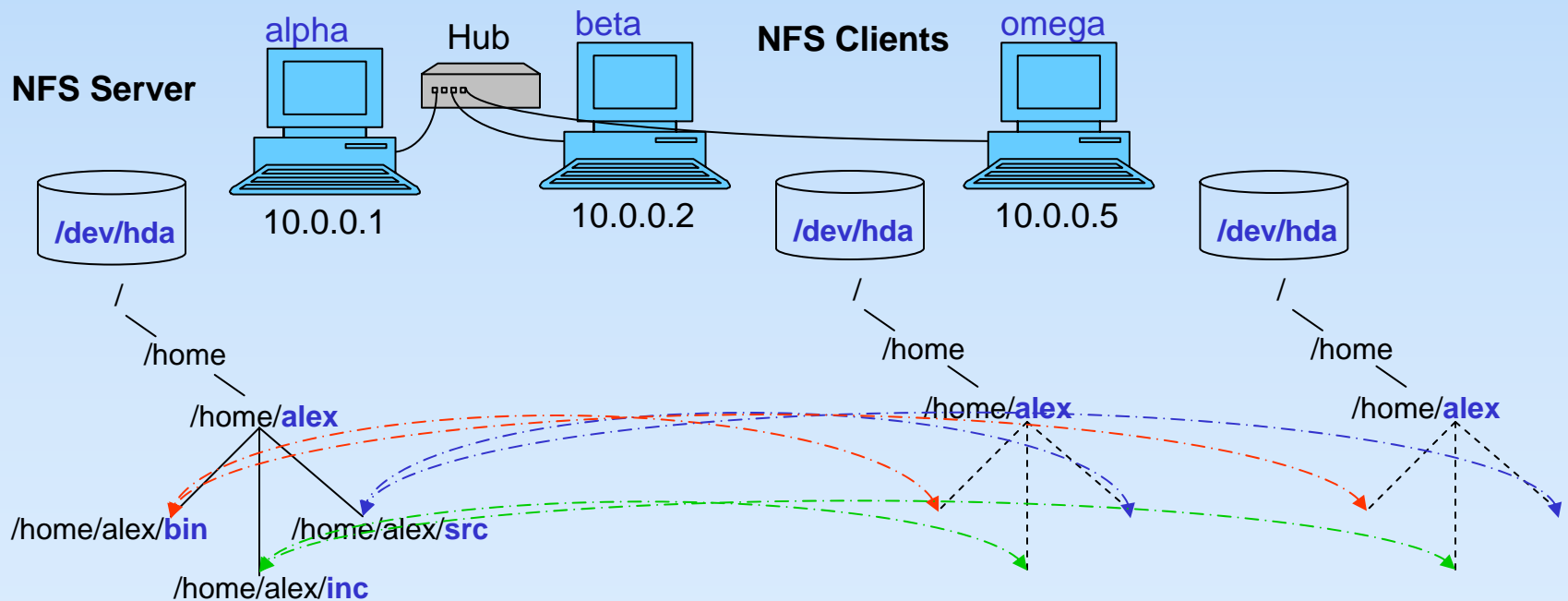
Slide # 11

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Remote User File Structure:

Network File System (NFS): the *illusion of locality* via remote-mount points



Slide # 12

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Linux OS Networking Files:

- First, file `/etc/hosts` belongs on all the network nodes:

```
127.0.0.1      localhost      localhost.chaos.org
10.0.0.1      alpha         alpha.chaos.org
...
10.0.0.5      omega        omega.chaos.org
```

- Next, file `/etc/exports` on 10.0.0.1, the NFS server named alpha:

Server → `/home` (rw)

- Finally, file `/etc/fstab` on every node except the server named alpha:

```
Clients → /dev/hda1      swap          swap          defaults      0 0
          /dev/hda2      /             ext2          defaults      1 1
          10.0.0.1:/home  /home        nfs           rw            0 0
          /dev/fd0        /mnt/floppy  ext2          noauto        0 0
          none           /proc        proc          defaults      0 0
```

Slide # 13

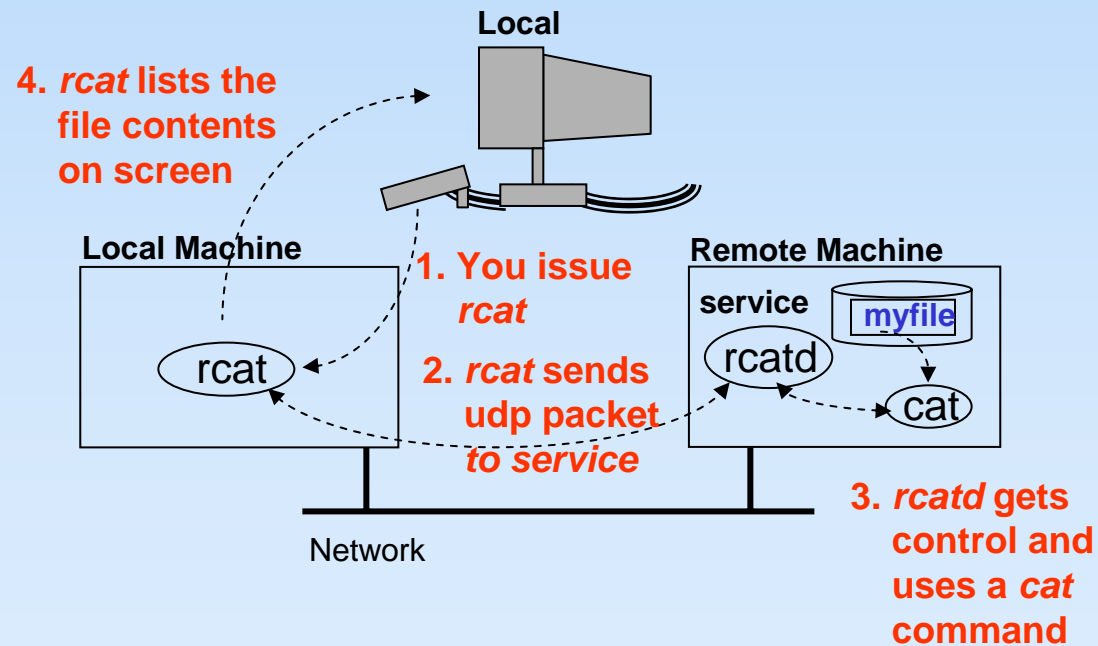
Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

UDP or TCP
socket

Internetworking Services: Operation

The *illusion of locality* via internetworking services



Slide # 14

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Internetworking Services: Configuration (using inetd)

- Add a line to file /etc/services on each remote-server node:

`rcatd 5000/udp # remote-cat UDP service on port 5000`

Refers to
entry in
services



- Add a line to file /etc/inetd.conf on each remote-server node:

`rcatd dgram udp wait alex /home/alex/bin/rcatd`

- Reconfiguration: `omega:/root> killall -HUP inetd`

Sequence of events:

1. Client process sends a UDP packet to server's port 5000
2. Daemon (inetd) starts process at /home/alex/bin/rcatd
3. Service reads incoming UDP packet data from "keyboard"

Linux Cluster Architecture

Internetworking Services: Configuration (using xinetd)

- File `/etc/xinetd.d/rcatd` on each (xinetd) remote-server node:

```
service rcatd
{
    port                = 5000
    socket_type         = dgram
    protocol            = udp
    wait                = yes
    user                = alex
    server              = /home/alex/bin/rcatd
    only_from           = 10.0.0.0
    disable             = no
}
```

Refers to name of service

Means 10.0.0.*

- Reconfiguration: `omega:/root> /etc/rc.d/init.d/xinetd restart`

Linux Cluster Architecture

Distributed Systems C-Language Skills:

Subtasking on a single processor
(details are in the book)



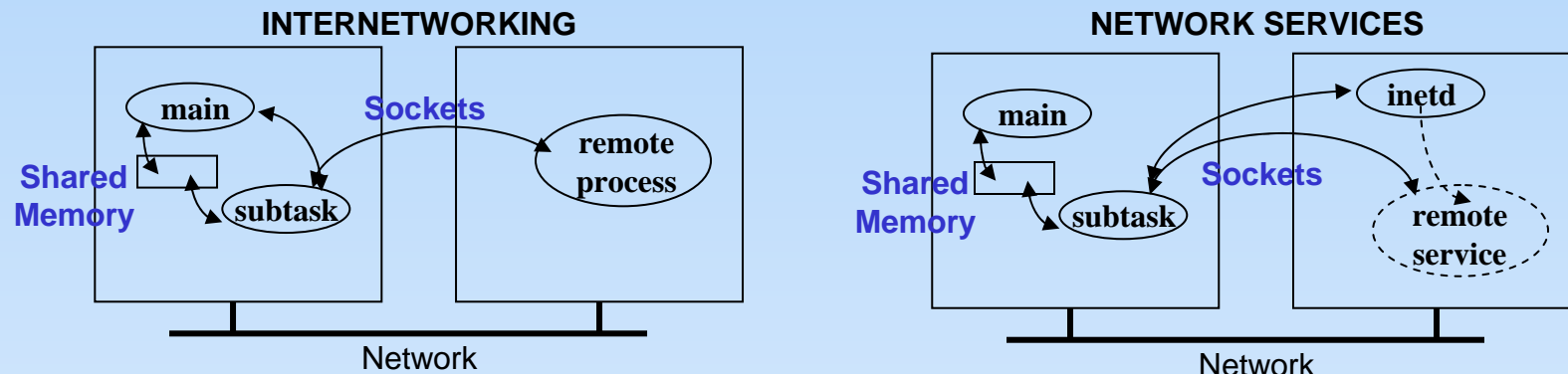
- Shared memory and semaphores
- Signal header/structure
- Handler initialization
- Signal processing and handler re-initialization

Slide # 17

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Distributed Systems C-Language Skills:



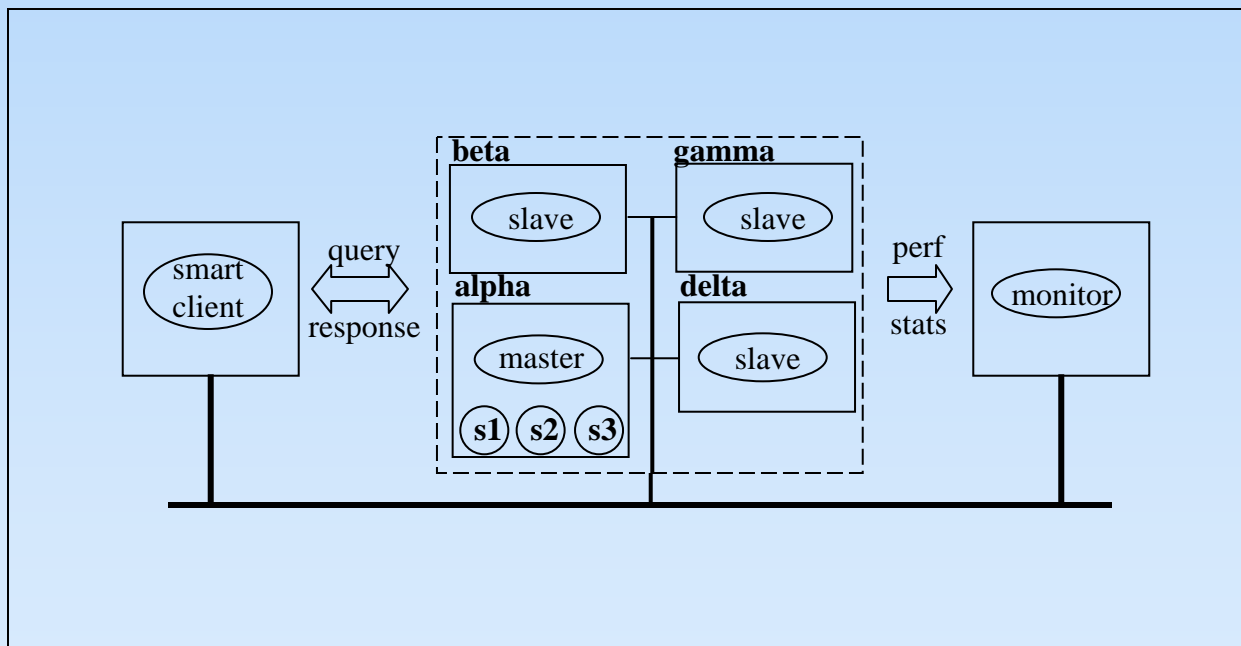
- Internetworking headers/structures
- Socket initialization
- Remote service "discovery"
- Reliable communications (topic discussions, anyway)

Slide # 18

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Master-Slave Cluster Server - Overview:



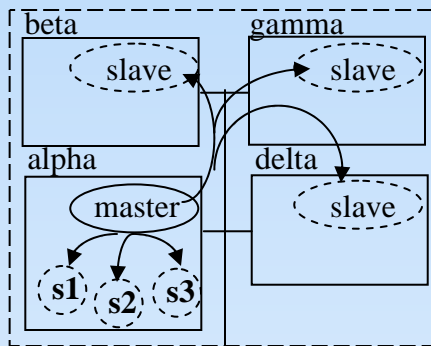
Slide # 19

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

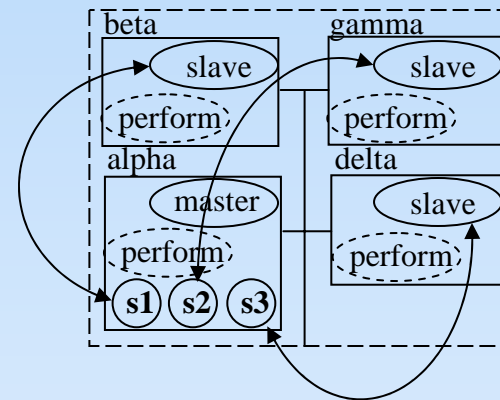
Master-Slave Cluster Server - Initialization:

Broadcast starts *slave* tasks...



master starts local subtask, one for each registering remote slave.

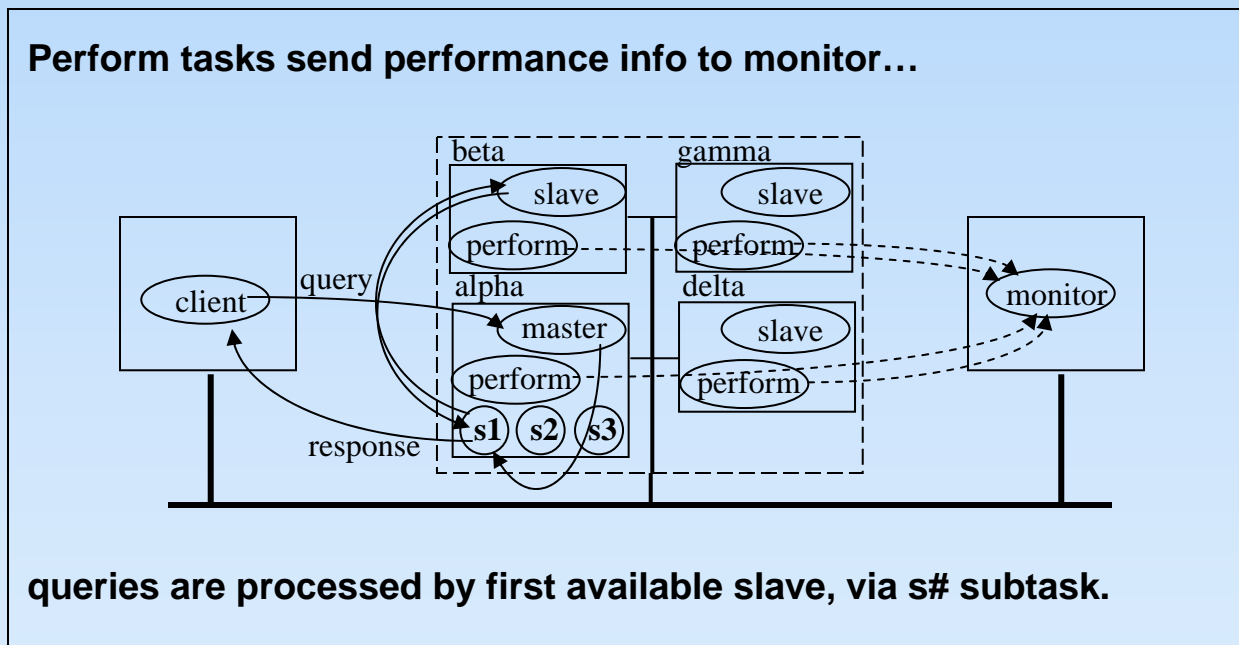
Local subtasks contact slaves...



all tasks start a *perform* subtask.

Linux Cluster Architecture

Master-Slave Cluster Server - Operation:



Linux Cluster Architecture

Performance Statistics - Pseudo-file Text*:

- CPU Utilization in /proc/stat - Running Jiffy Counts in each State

```
cpu 1256 0 1566 565277
```

Idle (since boot)
system
nice
user

- Disk Reads and Writes in /proc/stat - Running I/O Counts

```
disk_rio 1270 0 0 0  
disk_wio 1337 0 0 0
```

I/O count (since boot)

* Note that the exact meaning of proc file content can be OS release dependent: see `man proc`

Linux Cluster Architecture

Performance Statistics - More Pseudo-file Text:

- Memory Utilization in /proc/meminfo - Current Values


```
Mem: 14942208 13713408 1228800 ...
```



Free (right now)
Used
Total

- Packets Sent and Received in /proc/net/dev - Running I/O Counts

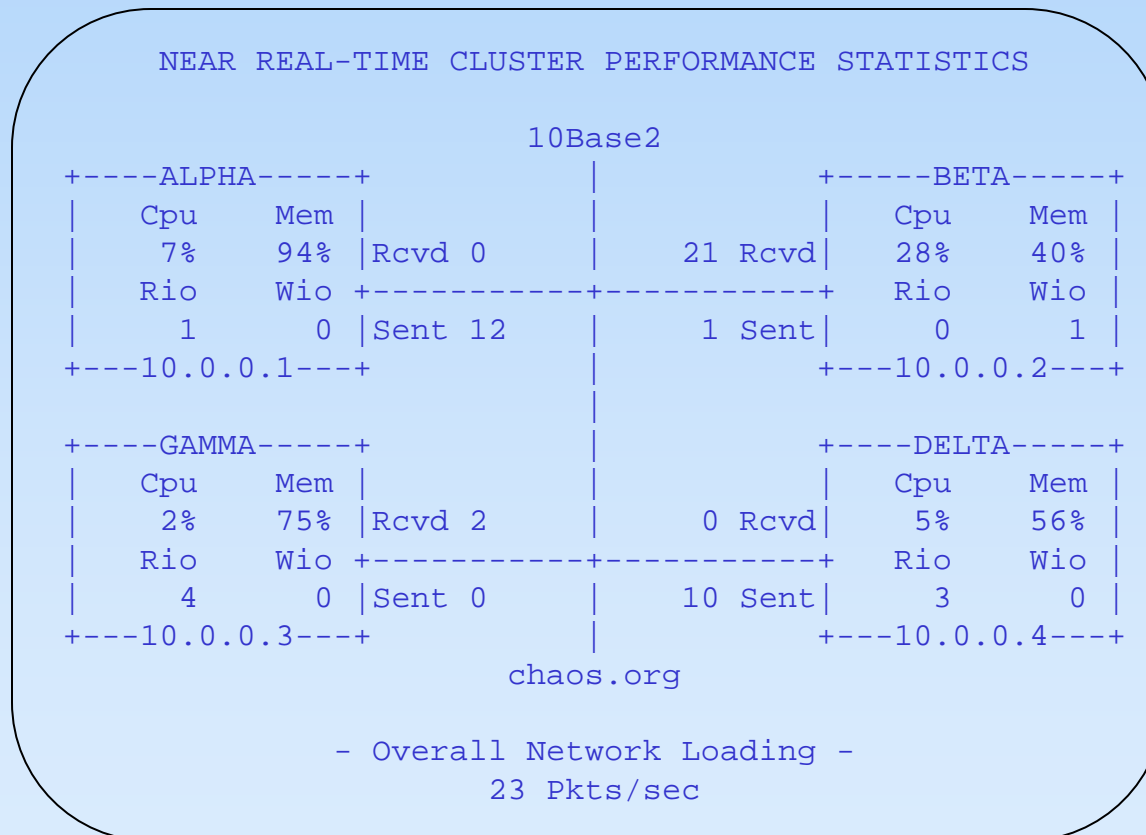
```
lo: 80 0 0 0 0 80 0 0 0 0  
eth0: 115 0 0 0 0 68 0 0 0 0
```



Transmitted (since boot)
Received

Linux Cluster Architecture

Internal Performance Statistics - Display:



Slide # 24

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Performance Monitoring - External (displayed):

- Resource utilization reporting via a smart client process:

RESPONSE		OBSERVATIONS				
TIME (msec)		10	20	30	40	50
1	10					
11	20					
21	30	*****				
31	40	*****				
41	50	**				
51	60					
61	70					
71	80					
81	90					
91	100					

50 Total Observations

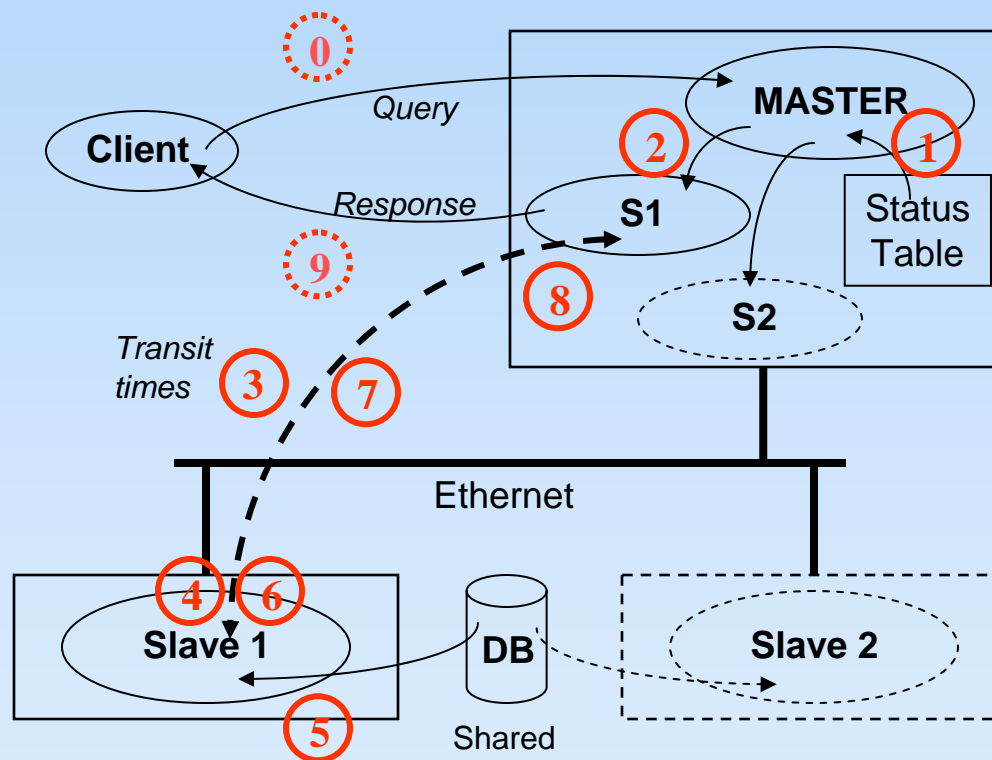
Average = 30 milliseconds ...what if you're not happy with this level of performance?

Slide # 25

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

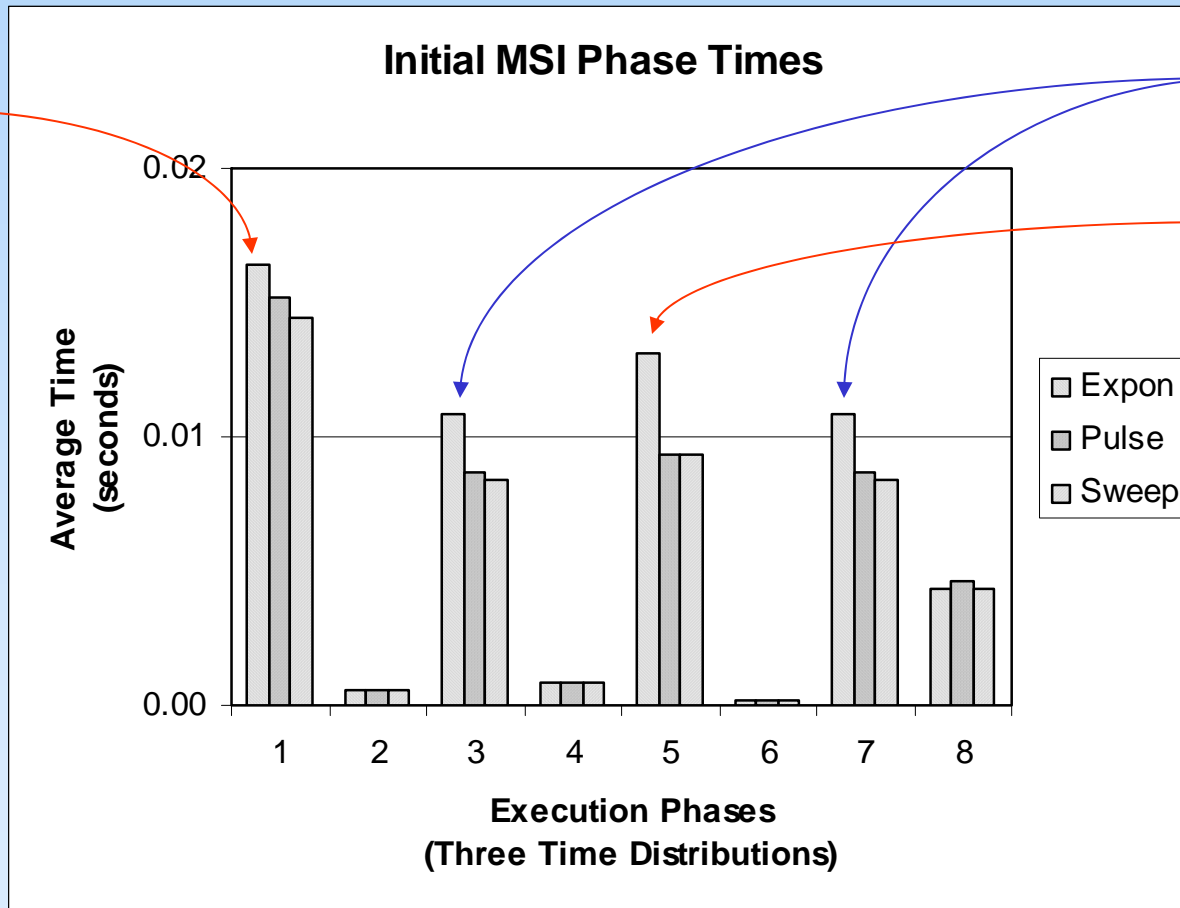
Performance Tuning - Defining Execution Phases:



Linux Cluster Architecture

Performance Tuning - Execution Phase Times:

Found a bug!



Not a SW issue.

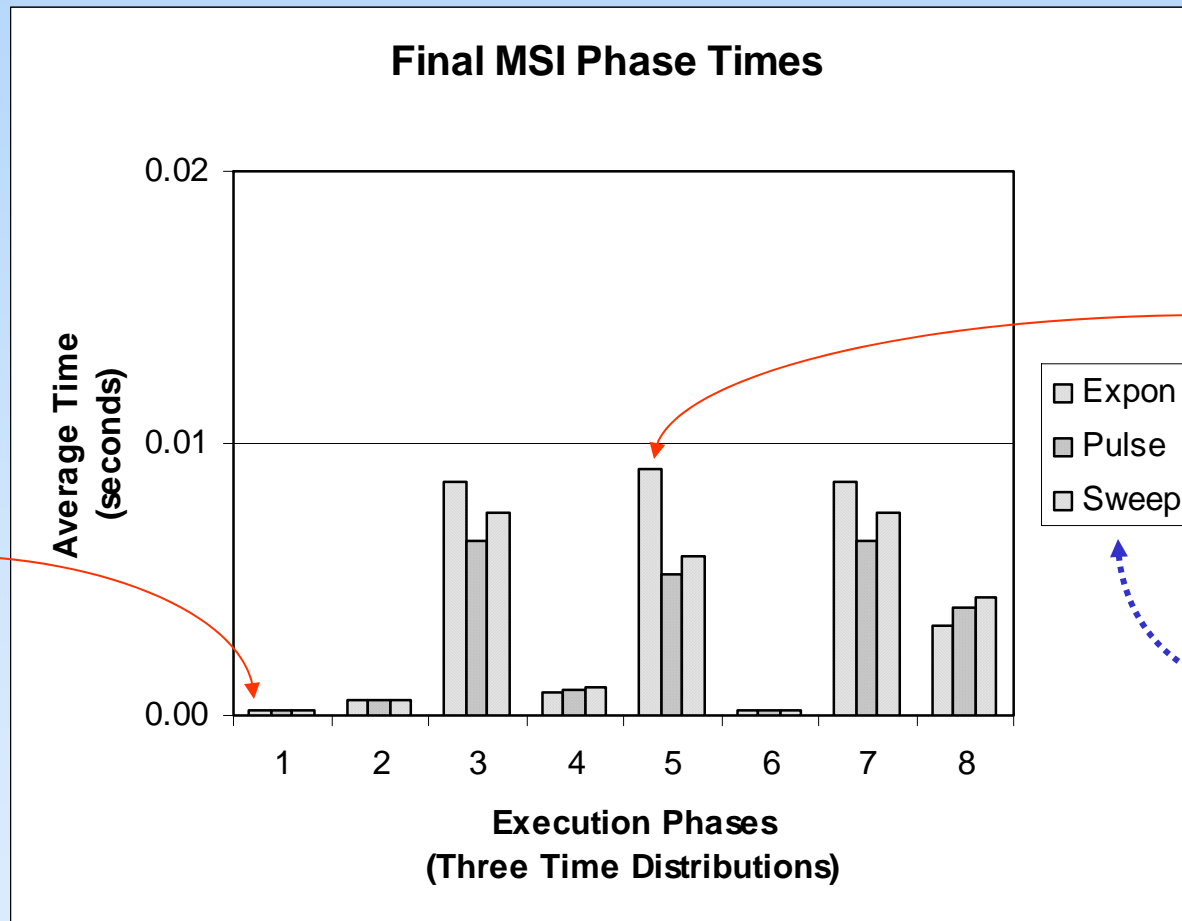
Leave the file open?

Slide # 27

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Performance Tuning - Final Times:



Dramatic reduction!

About a 10% improvement (not too bad)

See book for further details on statistical distributions.

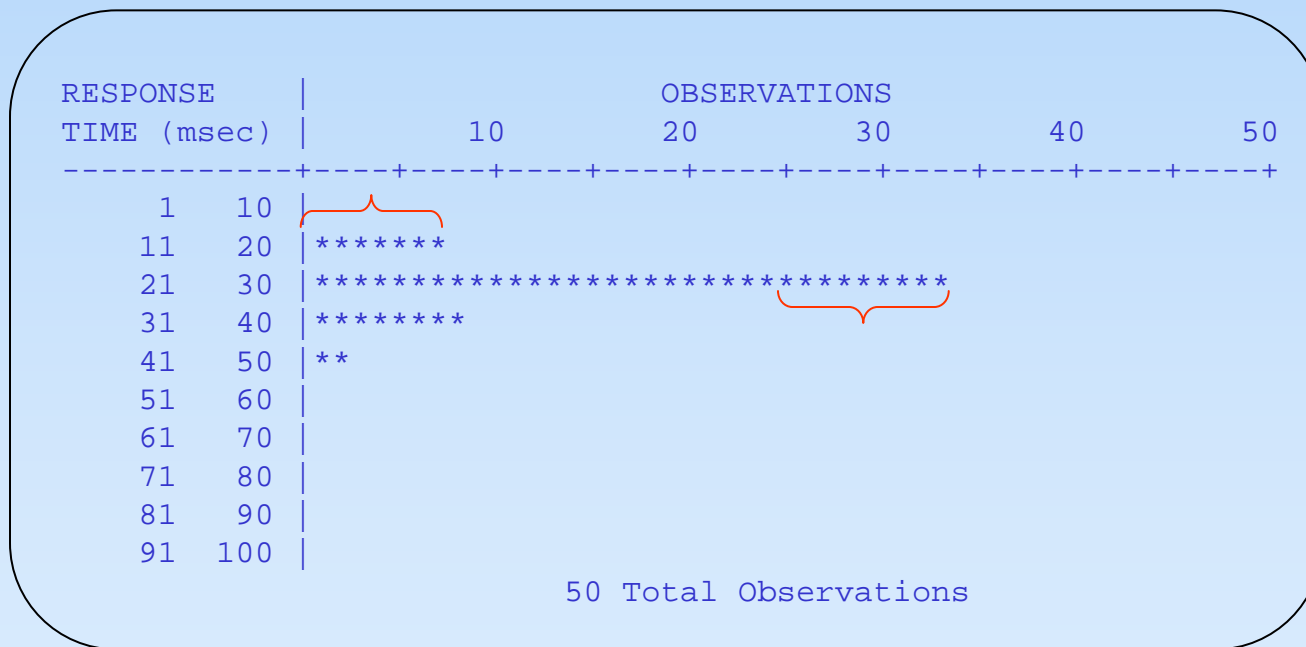
Slide # 28

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Performance Tuning:

- The proof is in the pudding!



Average = 25 milliseconds = 17% improvement!

Slide # 29

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Further Details are in the Book:

- Download all the C source code for free:

<http://www.sampublishing.com>

- Search on "Linux Cluster Architecture" or "Vrenios"
- Click on the "Downloads" link in the book description
 1. Individual chapter examples are in zip files
 2. A complete user-files environment is in a tar.gz file

- Book Signings:

Sep 8th

Borders Chandler, Sunday @ 2pm

Sep 15th

Borders Arrowhead, Sunday @ 2pm

Oct 25th

Barnes & Noble Arrowhead, Friday @ 7pm

Slide # 30

Copyright © 2003 Alexander Vrenios

Linux Cluster Architecture

Further Reading:

Distributed Operating Systems, Andrew S. Tanenbaum
(of **MINIX** and **AMOEBA** fame!), Prentice Hall, 1995

Unix Distributed Programming, Chris Brown, Prentice Hall, 1994

Advanced Programming in the UNIX Environment,
W. Richard Stevens, Addison-Wesley, 1992

Linux Programming White Papers, Rushling, et al, CoriolisOpen, 1999

-> Further details about the early development of my cluster are in:

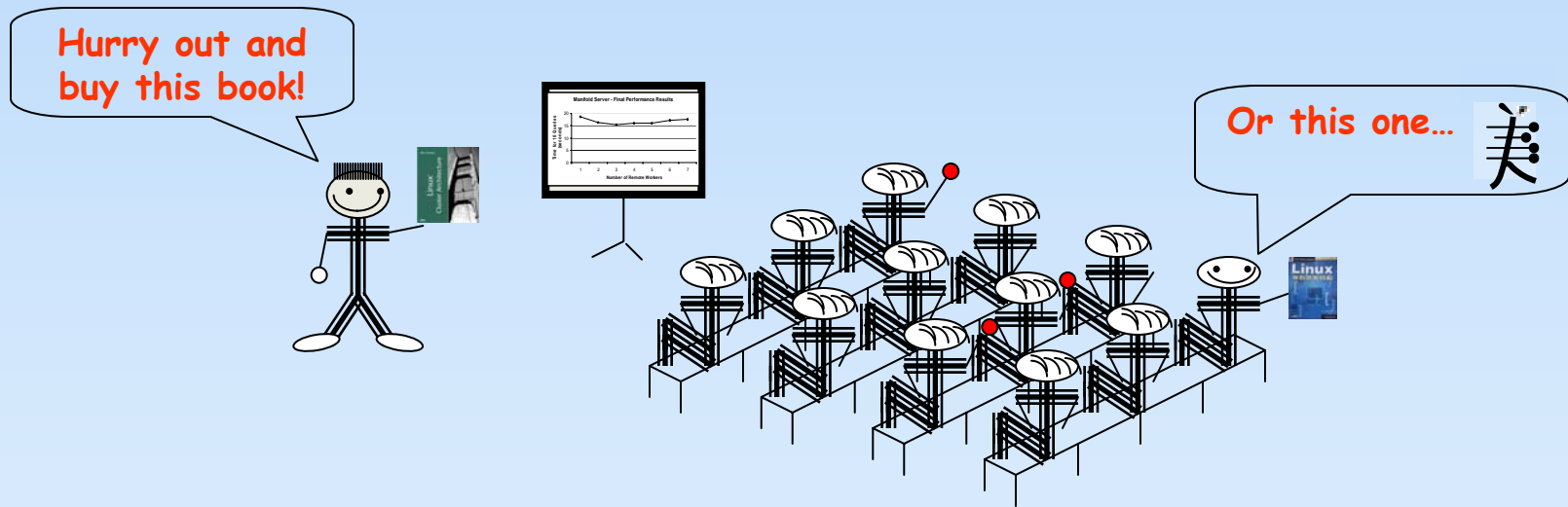
"CHAOS: A Cheap Array of Outmoded Systems," Alex Vrenios,
LinuxGazette.com, October 1998

"CHAOS Part 2," LinuxGazette.com, Alex Vrenios, December 1998

Linux Cluster Architecture

You've been a terrific audience!

Any questions?



Slide # 32

Copyright © 2003 Alexander Vrenios