A Parallel Supercomputer Made Out Of PCs?

Tuesday, June 3, 1997

Hank Dietz
Assoc. Prof. of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-1285
hankd@ecn.purdue.edu
http://dynamo.ecn.purdue.edu/~hankd

What Is And Isn't There?

- Parallel PC architectures:
  - SIMD within a register (using MMX)
  - SMP (using an MPS machine)
  - PC host for attached processor(s)
  - PC cluster
- The Linux OS:
  (`http://sunsite.unc.edu/mdw/linux.html`)
  - A nice, free, highly popular UNIX
  - Support for parallel processing?

What Constitutes A Typical PC?

- Intel 386 family or compatible processor(s)
- 256 - 512 KBytes Cache & 8 - 512 MBytes DRAM
- A 3.5" 1.44 MByte floppy drive
- 1 - 9 GByte hard disk(s)
- 6x - 16x CDROM drive, backup?
- VGA-compatible video card
- 15" - 21" color monitor
- RS232 serial, parallel
- ISA/EISA & PCI or PCMCIA busses for I/O cards
- 200 - 300 Watt power supply
- A bunch of fans...

New Processors... The Good News

Processor	Approx. SPECint95	Approx. SPECfp95
Pentium 75	2.4	2.1
Pentium 90	2.9	2.5
Pentium 100	3.2	2.8
Pentium 120	3.6	3.0
Pentium 133	4.0	3.3
Pentium 150	4.1	3.3
Pentium 166	4.6	3.8
Pentium 200	5.1	4.2
Pentium 166 MMX	5.6	4.3
Pentium 200 MMX	6.4	4.7
Pentium Pro 150	6.1	5.4
Pentium Pro 180	7.3	6.1
Pentium Pro 200	8.2	6.2
Pentium II 233	9.5	6.4
Pentium II 266	10.8	6.9
PowerPC 604 133	4.7	3.8
HP PA-7200 120	4.6	8.3
UltraSPARC 200	7.7	11.4
Alpha 21164 300	7.3	12.2

- And these benchmarks don't even use MMX...
- New Intel MMX competition: AMD K6 & Cyrix M2...

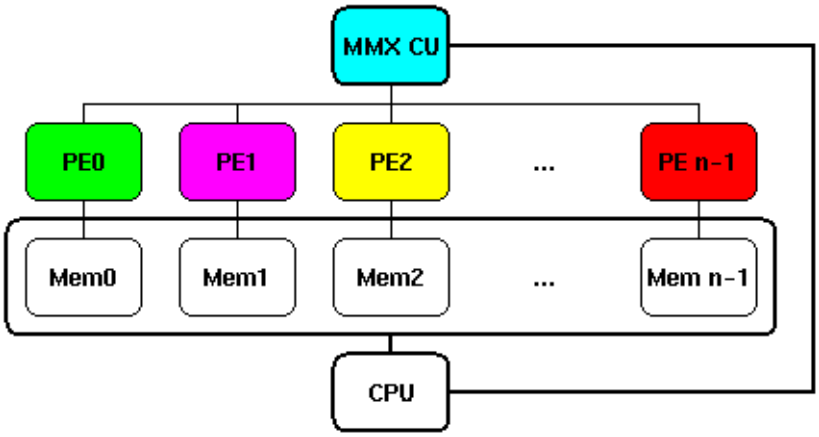
Linux

- Written by Linus Torvaldis...
  used and improved by thousands
- Copyleft source code
- Full UNIX, mostly POSIX
- XFree86, C, C++, Fortran, Pascal, etc.
- Supports "IBM compatible" PCs,
  DEC Alphas, Sun SPARCs, PowerPCs, etc.
- **Linux Documentation Project**
  `http://sunsite.unc.edu/mdw/linux.html`

Linux Vs. Other OS

- Free BSD variants
  - Optimized for heavy load
  - Bigger and slower
- Solaris, SCO, etc.
  - Not free, no source code
  - Generally bigger and slower
- MS Windows NT
  - The not-quite-UNIX future?
  - Supports more hardware, has better security
  - Not free, no source code, bigger and slower

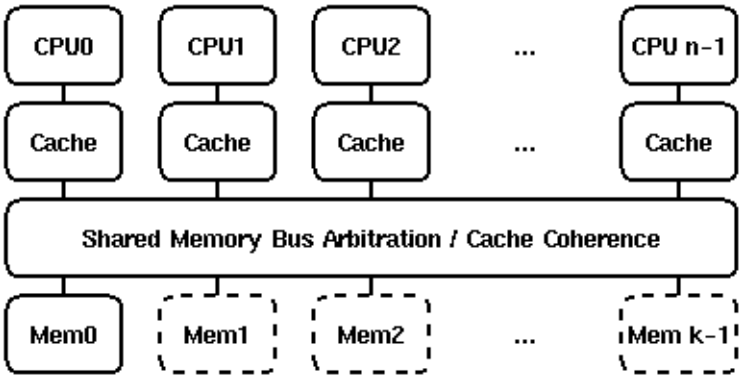
SWAR (Using MMX)



- SIMD over integer fields in 64-bit registers
- Processors with (MMX) MultiMedia eXtensions
- Developing SWAR module languages/compilers

  `http://dynamo.ecn.purdue.edu/~hankd/SWAR/`

SMP (On MPS Systems)

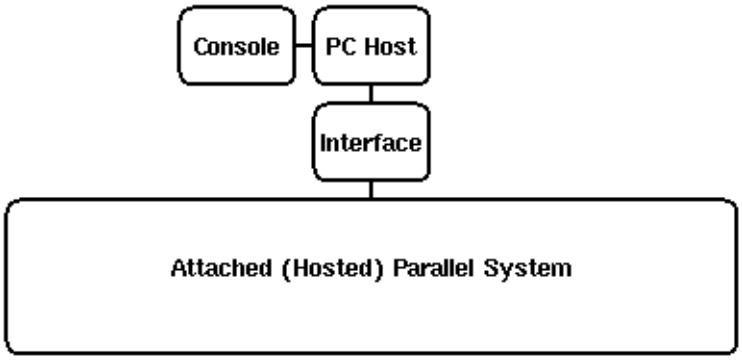


- **Intel MultiProcessor Standard (MPS) 1.1 and 1.4**, `http://www.intel.com/IAL/processr/mpovr.htm`
- Shared cache, shared bus, multiple busses, etc.
- Many vendors... most with 2 or 4 PEs `http://www.uruk.org/~erich/mps-hw.html`
- SMP Linux supports up to 16 processors
- Everything works, but only 1 CPU in kernel

Programming Models

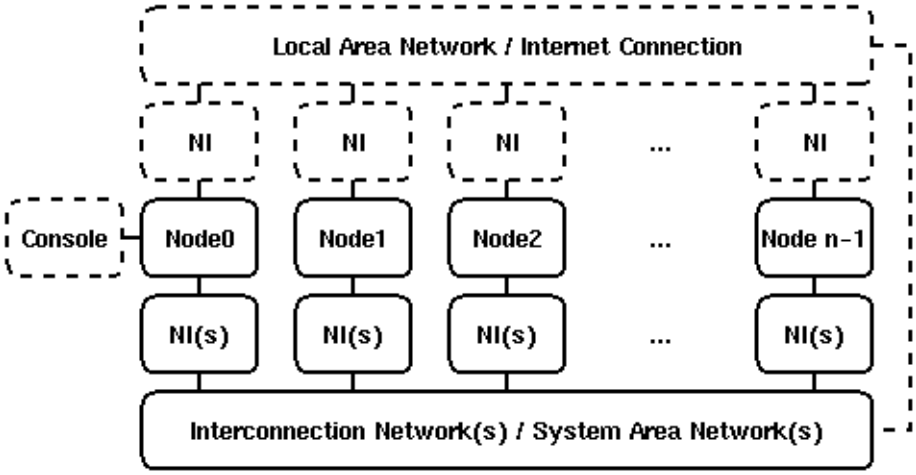
- "Shared everything"
  - Supported by Linux `clone()` call
  - `bb_threads`
  - LinuxThreads (POSIX Pthreads port)
    `http://pauillac.inria.fr/~xleroy/linuxthreads`
- "Shared something"
  - System V IPC shared memory segments
  - `mmap()` segments
- `volatile`, "false sharing," scheduler issues, etc.
  `http:/yara.ecn.purdue.edu/~pplinux/ppsmp.html`
  - +: MIMD, μs latency, shared-memory, message-passing
  - -: Surprises, scaling trouble, I/O contention

Attached Processors



- Linux PC is front-end for a parallel machine
- Lots of choices, nothing standard
- DSP cards, FPGA cards, array processors

Linux PC Cluster



- Cluster of PCs connected by one or more networks
- Scalable, but generally high latency

Linux PC Cluster

- PAPERS 960801 Linux Cluster With VGA Video Wall

Why Clusters?

- Can use what you already have
  - Underutilized PCs
  - Existing or easily added networks
- Can coexist with other PC uses;
  e.g., "Windoze" by day, Linux cluster by night
- Easy to upgrade incrementally
  (but may yield a heterogeneous system)
- Video walls, etc.
- Scalable to *lots* of processors, memory, disk I/O

Linux... A PC Cluster OS?

- Small & fast UNIX/POSIX on a PC
- Supports UDP and TCP sockets
- Supports IP load sharing
- Some support for ATM AAL*
- Supports user-space I/O mapping
- Supports NFS (Network File System)
- In summary, Linux is a good **node OS**

What Linux Does Not Provide

- A cluster of PCs acts like multiple PCs
  - No parallel execution control
  - No parallel file system (NFS doesn’t count)
  - No parallel machine administration
- These can be built on top of Linux

Software Libraries

- PVM

  `http://www.epm.ornl.gov/pvm/pvm_home.html`

- MPI (standard, LAM, and MPICH)

  `http://www.mcs.anl.gov:80/mpi/`

  `http://www.osc.edu/lam.html`

  `http://www.mcs.anl.gov/home/lusk/mpich/`

- AFAPI

  `http://garage.ecn.purdue.edu/~papers/AFAPI/`

- Various page-fault DSM systems...

- Parallel file I/O...

  `http://www.cs.dartmouth.edu/cs_archive/pario/bib.html`

Parallel Languages

- Jade/SAM (concurrent C dialect) on PVM
  `http://suif.stanford.edu/~scales/sam.html`
- MPL (Maspar Programming Language) on AFAPI
  `cohen@epic.eb.uah.edu`
- Parallaxis (data parallel Modula-2) on PVM
  `http://www.informatik.uni-stuttgart.de/ipvr/bv/p3/p3.html`
- pC++ (data parallel C++) on PVM
  `http://www.extreme.indiana.edu/sage/`
- ZPL (array-based language) on PVM
  `http://www.cs.washington.edu/research/projects/orca3/zpl/www/`

PC Network Hardware

- Many very different types
- What's available changes daily
- Bandwidth from 115 Kb/s to 1.6 Gb/s
- Latency always high, but a wide range
- Cost from a few $ to thousands per PC
- Max number of PCs from 2 to thousands
- Max cable length from 10' to miles
- Could even use the internet...

Lots Of Networks!

Network Type	Prog. Model	Available?	Linux?	PC Cost Mult.	Port	Max. PEs	Bandwidth (Mbits/s)	Latency (µs)	*n*-PE O( )
HiPPI		Product		2.8	EISA, PCI	16/hub	1,600.0		log *n*
CAPERS	AFAPI	Public	Yes	1.0	SPP	2	1.2	3	
Serial HiPPI		Product		3.3	PCI	16/hub	1,200.0		log *n*
SCI							1,000.0		
FC							1,062.0		
Myrinet	Library	Product	Yes	1.9	PCI	8/hub	1,280.0	> 9	log *n*
FireWire		Product			Chipset	63	200.0	125	*n*
SCSI	File I/O	Product	Yes	1.2	Chipset	2			*n*
ParaStation	HAL	Product	Yes	2.0	PCI	> 100	125.0	2	sqrt *n*
SHRIMP		Research	Yes		EISA		180.0	5	sqrt *n*
ParaPC		Research			EISA	2?	40.0	5	
TTL_PAPERS	AFAPI	Public	Yes	1.1	SPP	4/hub	1.6	3	log *n*
ParaStation	Socket Library	Product	Yes	2.0	PCI	> 100	93.0	13	sqrt *n*
PLIP	Socket	Copyleft	Yes	1.0	SPP	2	1.2	1,000	
ATM	AAL5	Product	Yes	2.5	PCI	16/hub	155.0	100	log *n*
Fast Ethernet (unswitched)	Socket	Product	Yes	1.2	PCI	16/hub	100.0	500	> *n*
USB		Product			Chipset	127	12.0	1,000	*n*
Ethernet	Socket	Product	Yes	1.1	ISA	200	10.0	500	> *n*
Fast Ethernet (switched)	Socket	Product	Yes	1.4	PCI	16/hub	100.0	500	log *n*
ATM	Socket	Product	Yes	2.5	PCI	16/hub	155.0	1,000	log *n*
Ethernet (switched)	Socket	Product	Yes	1.2	ISA	16/hub	10.0	500	log *n*
IrDA	IrLAP	Product	Yes		Chipset	2	4.0	1,000	
ARCNET	Socket	Product	Yes	1.2	ISA	255	2.5	1,000	*n*
SLIP	Socket	Copyleft	Yes	1.0	RS232	2	0.1	10,000	

Networks (Decreasing Bandwidth/Latency/Cost per Node)

Some Networks Don’t Scale

- **CAPERS** (Cable Adapter for Parallel Execution and Rapid Synchronization)
  `http://garage.ecn.purdue.edu/~papers/`

- **SCSI** (Small Computer Systems Interface)

- **ParaPC**
  `http://wwwipd.ira.uka.de/parastation`

- **PLIP** (Parallel Line Interface Protocol)

- **IrDA** (Infra-red Device Access)
  `http://www.irda.org/`

- **SLIP** (Serial Line Interface Protocol)
  Also **CSLIP**, **PPP**, etc.

Some Not Widely Available

- **SCI** (Scalable Coherent Interconnect)
  `http://www.cern.ch/HSI/sci/sci.html`

- **FC** (Fibre Channel)
  `http://www.amdahl.com/ext/CARP/FCA/FCA.html`

- **FireWire** (AKA IEEE 1394)
  `http://www.firewire.org/`

- **SHRIMP** (Scalable, High-Performance, Really Inexpensive Multi-Processor)
  `http://www.CS.Princeton.EDU/shrimp/`

- **ParaPC**
  `http://wwwipd.ira.uka.de/parastation`

- **USB** (Universal Synchronous Bus)
  `http://www.usb.org/`

Some Not Supported By Linux

- **HiPPI** (High Performance Parallel Interface)
  Serial HiPPI
  `http://www.cern.ch/HSI/hippi/`

- **SCI** (Scalable Coherent Interconnect)
  `http://www.cern.ch/HSI/sci/sci.html`

- **FC** (Fibre Channel)
  `http://www.amdahl.com/ext/CARP/FCA/FCA.html`

- **FireWire** (AKA IEEE 1394)
  `http://www.firewire.org/`

- **ParaPC**
  `http://wwwipd.ira.uka.de/parastation`

- **USB** (Universal Synchronous Bus)
  `http://www.usb.org/`

The Currently Viable Networks

- **Myrinet**

  `http://www.myri.com/`

- **ParaStation**

  `http://wwwipd.ira.uka.de/parastation`

- **PAPERS** (Purdue's Adapter for Parallel Execution and Rapid Synchronization)

  `http://garage.ecn.purdue.edu/~papers/`

- **ATM** (Asynchronous Transfer Mode)

  `http://lrcwww.epfl.ch/linux-atm/`

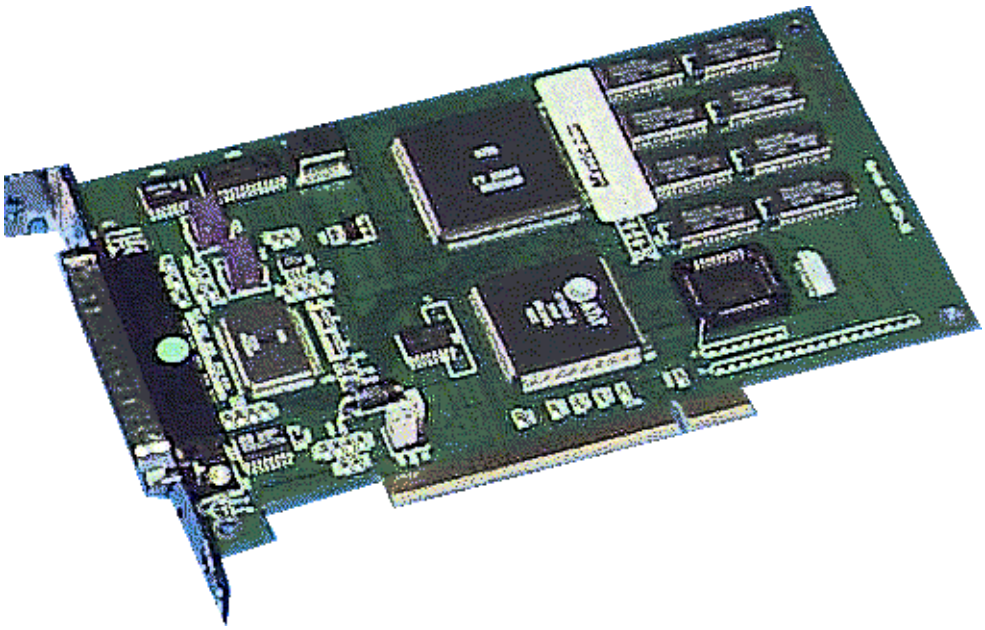
- **Ethernet** *in any of its many flavors*

  `http://cesdis.gsfc.nasa.gov/linux/beowulf/beowulf.html`

- **ARCNET**

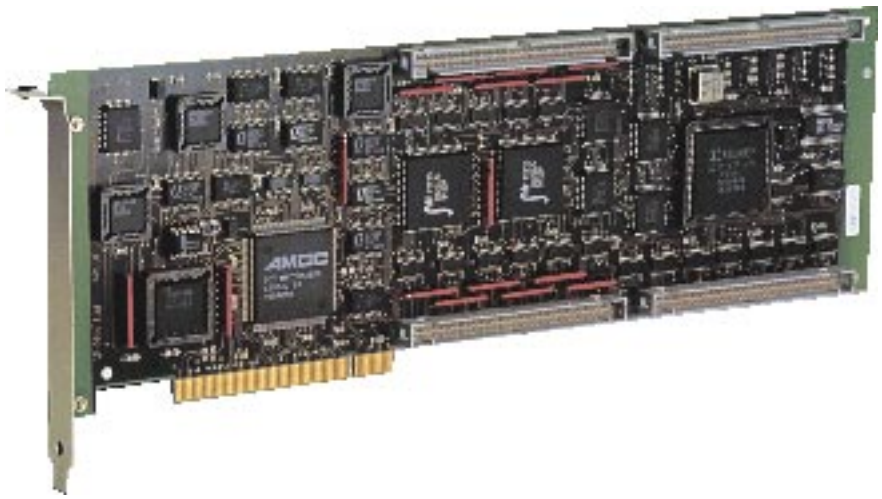
  `http://www.arcnet.com/`

Myrinet



- Available from a single vendor, Myricom
- Two incompatible versions: LAN & SAN
- Custom PCI bus cards & switched hubs
- Topology flexible, hypercube recommended
- Message passing with low latency & high bandwidth
- `http://www.myri.com/`

ParaStation



- University of Karlsruhe Department of Informatics (buy from `http://www.hitex.com:80/parastation/`)
- Custom PCI bus card (successor to the ParaPC EISA card)
- Cards connect in 2D toroidal mesh
- User-level socket compatible interface
- Message passing with low latency & good bandwidth, also implements barrier synchronization
- `http://wwwipd.ira.uka.de/parastation`

TTL_PAPERS

- Purdue University
  School of Electrical and Computer Engineering
  (hardware design and software are public domain;
  can buy units from `http://idt.net/~hgdietz/`)
- Custom 4 processor "hub"
- Hubs connect in 4-ary tree topology
- AFAPI (Aggregate Function API) functions directly access parallel port registers from user process
- Aggregate functions with low latency & low bandwidth
- `http://garage.ecn.purdue.edu/~papers`

ATM

- Standard, available from many vendors
  (still some interoperability problems)
- PCI bus cards & switched hubs
- Support for isochronous audio, video, etc.
- Topology flexible, long wires are ok
- Offers low latency, but not with sockets...
- Highest bandwidth standard network supported by Linux
- `http://lrcwww.epfl.ch/linux-atm/`

Fast Ethernet (100 Mb/s)

- Standard, available from many vendors
- Somewhat compatible with old 10 Mb/s Ethernet, and with upcoming 1 Gb/s Ethernet
- Cards are cheap, switched hubs are not; topology & multiple cards/PC can help
- Currently best performer for IP sockets
- Linux can load share across multiple Fast Ethernets
- `http://cesdis.gsfc.nasa.gov/linux/beowulf/beowulf.html`

Ethernet (10 Mb/s)

- The ubiquitous standard;
  you probably already have it
- Cards are nearly free, switched hubs are cheap
- Various topologies, even hubless (BNC "thin net")
- Wide variation in performance of cards
- Linux supports load sharing as a kernel option
- `http://cesdis.gsfc.nasa.gov/linux/beowulf/beowulf.html`

ARCNET

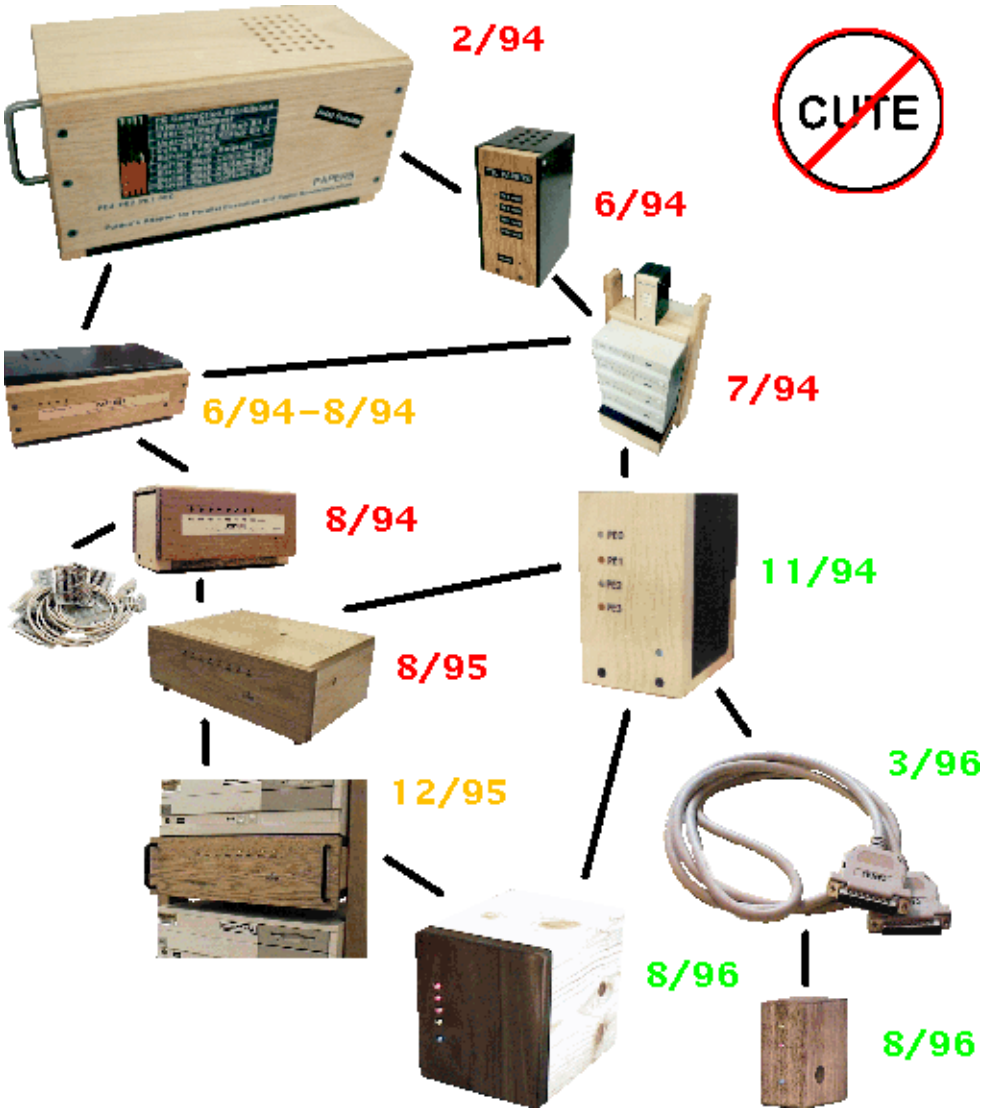
- An old standard for ATM-like stuff
  (mostly in embedded control systems)
- Physical bus topology, logical token ring
- Not cost effective for clustering...
- `http://www.arcnet.com/`

Which Network(s) Do What

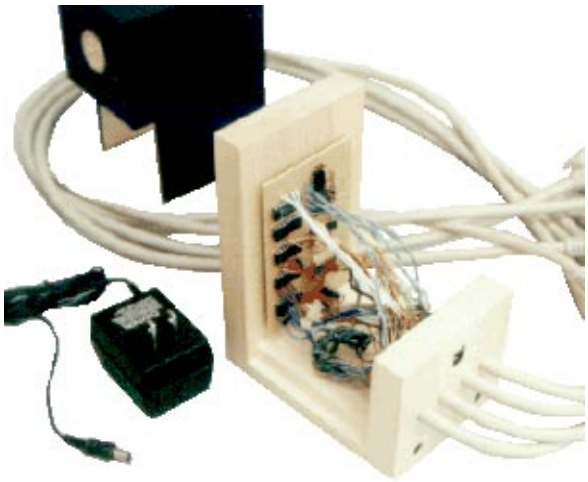
Use	Myrinet	ParaStation	PAPERS	ATM	Ethernet	ARCNET
MIMD	Good	Good	Fair	Good	Good	Good
SIMD	Poor	Fair	Good	Poor	Poor	Poor
Message Passing	Good	Good	Poor	Good	Good	Good
Shared Memory	Fair	Fair	Good	Poor	Poor	Poor
Aggregate Functions	Poor	Poor	Good	Poor	Poor	Poor
Bandwidth	Good	Good	Poor	Good	Good	Poor
Latency	Good	Good	Good	Fair	Poor	Poor
Standard?	No	No	No	Yes	Yes	Yes
PC Cost Multiplier	1.9x	2.0x	1.1x	2.5x	1.1x - 1.4x	1.2x

- Want a "Good" in every row
- Want a standard net (for IP, etc.)
- Want to be cost effective

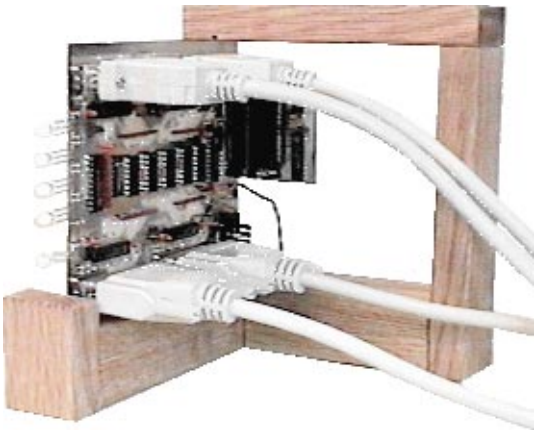
Evolution Of PAPERS



What's in TTL_PAPERS



- 4 Processor TTL_PAPERS (November 1994)



- Scalable PAPERS 960801 (August 1996)

What's in PAPERS

- Each PC connects via standard parallel port...
- Will improve bandwidth, add aggregates
- PAPERS unit contains 4 subsystems:
  - Fast barrier synchronization hardware
  - LED status display
  - Aggregate function data communication
  - Parallel signal support

Barrier Synchronization

- An *arbitrary group* of PCs can participate
- No PC executes past a barrier until *all* have arrived
- Two-cycle barriers using two barrier units:
  1. Output: present at barrier, reset barrier'
  2. Input: poll for *all* PCs present
- Synchronization time:
  - < 0.1 μs logic, < 0.2 μs cable, 1 μs port register
  - Total < 3 μs for 4 PCs; < 7 μs for 2,000+ PCs

LED Status Display

- Blue power on
- Bi-color LED/PE
  - Green: running
  - Red: at barrier
  - Orange: in OS
  - Black: no program

A Simple Demonstration

- Cluster of 4 sub-notebooks
- Multi-voice music
- Each note is randomly assigned to a PC; PCs without notes have red LEDs
- Without precise coordination, it isn’t music

Aggregate Functions

- Not shared memory nor message passing
- As a barrier side-effect, each PC can:
  1. Output: a datum
  2. Input: selected *function of data from all PCs*
- Broadcast: `bcastPut`*`Type`*`(d)`, `bcastGet`*`Type`*`()`
- Multi-broadcast: `putget`*`Type`*`(d, s)`
- Associative reductions: `reduce`*`OpType`*`(d)`
- Scans (parallel prefix): `scan`*`OpType`*`(d)`
- SIMD conditionals: `any(f)`, `all(f)`
- VLIW multi-way branch: `waitbar(f)`
- Voting & scheduling: `vote(s)`, `match`*`Type`*`(d)`

Performance (μs Latency)

System	Barrier Sync.	32-bit Putget (permutation)	64-bit Broadcast
MasPar MP-1 (4 PEs)	0.1	44.0	31.0
2xP100 SMP Linux, SHMAPERS AFAPI	0.7	1.3	0.9
2xSPARCv9 Solaris, SHMAPERS AFAPI	0.7	1.0	1.3
4xSPARCv9 Solaris, SHMAPERS AFAPI	1.0	2.5	2.7
Cray T3D (4 PEs)	1.7		
4x486 Linux Cluster, TTL_PAPERS Library	2.5	216.0	137.0
4x486 Linux Cluster, PAPERS1 Library	3.1	27.0	81.0
4xP90 Linux Cluster, TTL_PAPERS AFAPI	3.8	232.0	112.0
Cray T3D (4 PEs), Cray PVM	21.0		82.0
Intel Paragon XP/S (4 PEs)	530.0	700.0	210.0
4x486 Linux Cluster, PVM3 10Mb/s Ethernet	49,000.0	100,000.0	40,000.0

Parallel Signals

- Any PC can asynchronously signal *all* PCs
- Separate barrier used for signal acknowledge
- Used for:
  - Parallel job control, gang scheduling
  - Asynchronous shared memory
  - Asynchronous message passing

Shared Memory

- Asynchronous atomic access to shared objects
- Each PC has a copy, always up-to-date
- C++ templates intercept operations on shared objects
  - Lvalue store causes `s_update(a, s)`;
    Signal and update, ~60 μs typical
  - Rvalue load causes `s_poll()`;
    Poll and local fetch, ~1 μs
- Load balancing shared-memory Mandelbrot
  achieves > 250 MFLOPS using 4xP90 TTL_PAPERS

Buying Hints

- Buy via mail-order, but get everything in writing
- Good for "Windoze" ain’t always good for Linux
- Get only what Linux supports
  - Check SCSI controller, audio, PCMCIA cards, etc. using latest kernel `make xconfig`
  - Check video using latest `xf86config`
- Linux CDROMs are older than online

Cluster Configuration Comments

- Get at least 1 spare PC (or 1 for every "bunch")
- Separate connection to external network
- Strongly recommend standard desktop/tower cases
  - Industrial rack mount PCs suffer time lag, low quantity (good mostly for high reliability)
  - Easy to build a "rack mount" for standard PCs
- VGA & keyboard switchers nice for cluster management; get cheap video cards if you do this

What’s next?

- Hardware
  - Pentium II MPS
  - Intel has competition (e.g., AMD K6 MMX)
  - "Production" versions of high-performance networks: e.g., 1 Gb/s Ethernet, FireWire...
  - Price doesn’t go down, configuration goes up
- Software
  - Improved OS parallelism in SMP Linux
  - Improved scheduling for SMP Linux
  - SWAR MMX support
  - Creation of a "standard" Linux parallel processing support disk set
  - More stuff ported and debugged

How Do I Keep Up?

- Subscribe to Computer Shopper
  (no, they didn't pay me to say that ;-)
- The Parallel Processing Using Linux site
  (home of the Parallel Processing HOWTO)
  `http://yara.ecn.purdue.edu/~pplinux`
- The Linux in High-Performance Computing page
  `http://www.cs.cornell.edu/home/mdw/hpc/hpc.html`
- The PAPERS Project Home Page
  `http://garage.ecn.purdue.edu/~papers/`
- The Linux Documentation Project
  `http://sunsite.unc.edu/mdw/linux.html`

Conclusion

- MMX, SMP, attached processors all help
- Fast Ethernet is cost/performance winner
- Myrinet is performance winner
- PAPERS is latency (grain size) winner