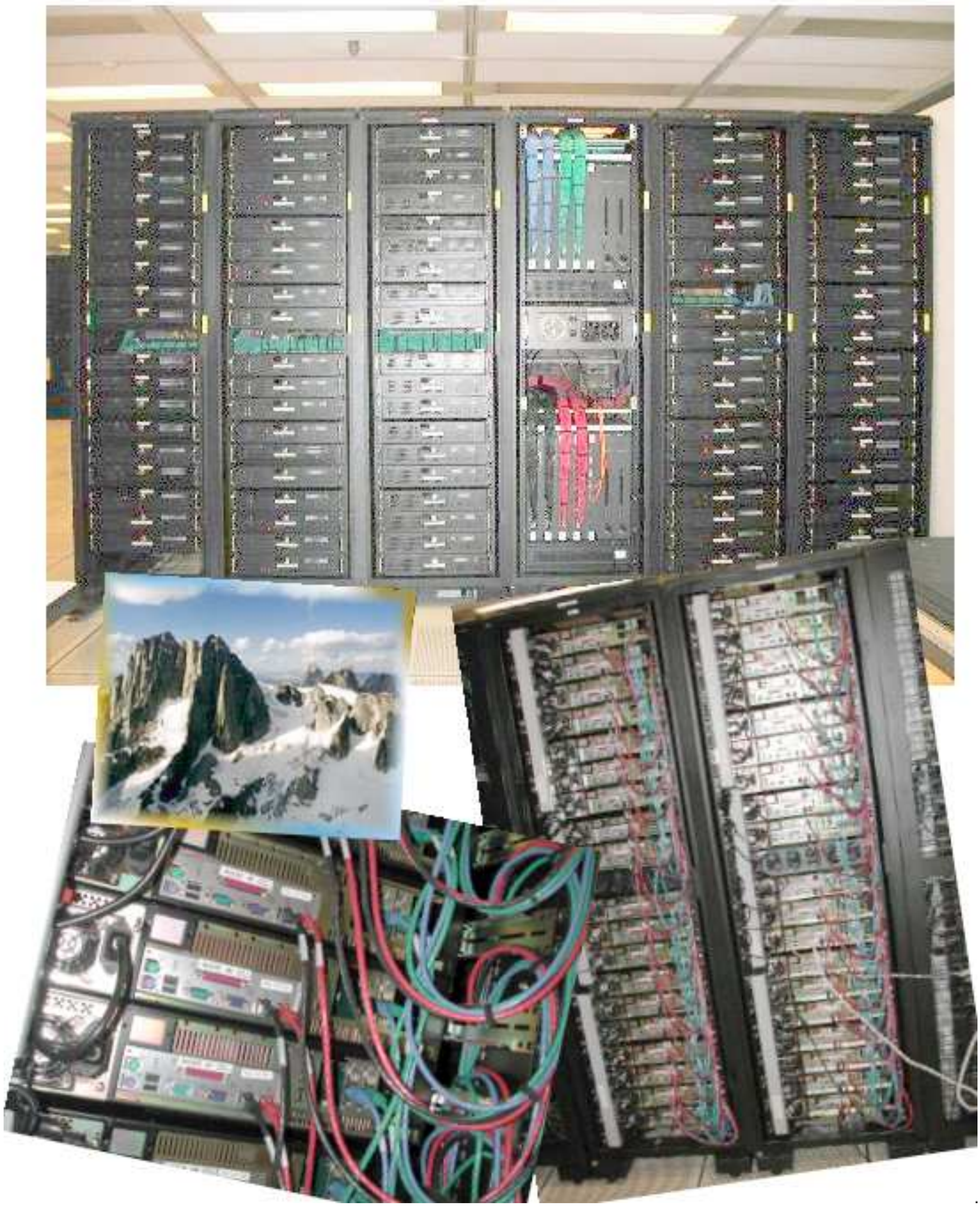
Building "The Bugaboos"
SFU's High-Performance Beowulf Cluster



Technical Data

- 96 nodes, 192 cpus
- CPU: AMD Athlon MP 1.2GHz
- Network: 3-way channel bonded fast ethernet
  + additional NFS network
- Linpack HPL performance: 144.6 Gflops

Linux Beowulf Clusters:

High-Performance Computing at SFU

SFU, Jun. 2002

- Introduction: Beowulf Clusters at SFU

- Hardware: Designing and Building a Cluster
  - Processors
  - Network
  - everything else ...

- Installation

- System Administration

- Using the Cluster: Parallel Programming (MPI)

What is a Beowulf?

T. L. Sterling, J. Salmon, D. J. Becker, D. F. Savarese,
How to Build a Beowulf:

"A Beowulf-class system is a cluster of mass-market commodity off the shelf (M^2COTS) PCs interconnected by low cost local area network (LAN) technology running an open source code Unix-like operating system and executing parallel applications programmed with an industry standard message passing model and library."

Hardware:

- off the shelf PCs
- network

Software:

- open source Unix-like operating system
- message passing library

$\Rightarrow$ All of these determine the performance of the Beowulf cluster!

History of Research Computing at SFU

- until 1999: one big machine
  beaufort.sfu.ca: 8 processor SMP SGI
- Spring 2000: first Beowulf cluster
  - 8 node / 16 processors
  - PIII 500MHz
  - fast ethernet network
  - RedHat Linux 6.1

  → extremely reliable
- May 2001: pre-release dual Athlon boards
- May 2002: first users on new Beowulf cluster

Hardware

- Which processor?

  PIII, P4, Athlon, Itanium?
  → price/performance depends on the applications!

PIII:	low price, average performance
P4:	tiny cache, no duals (May 2001)
Athlon:	superior floating point performance, duals since summer 2001
Itanium:	expensive, 64 bit

⇒ Best overall performance: dual Athlons

- What type of Network?

  fast ethernet, GigE, Myrinet?
  $\rightarrow$ price/performance: again - it depends ...

100baseT:	cheap, low latency, low bandwidth (90 Mbit/s)
GigE:	expensive (switches!), high latency, high bandwidth
Myrinet:	expensive, low latency, high bandwidth

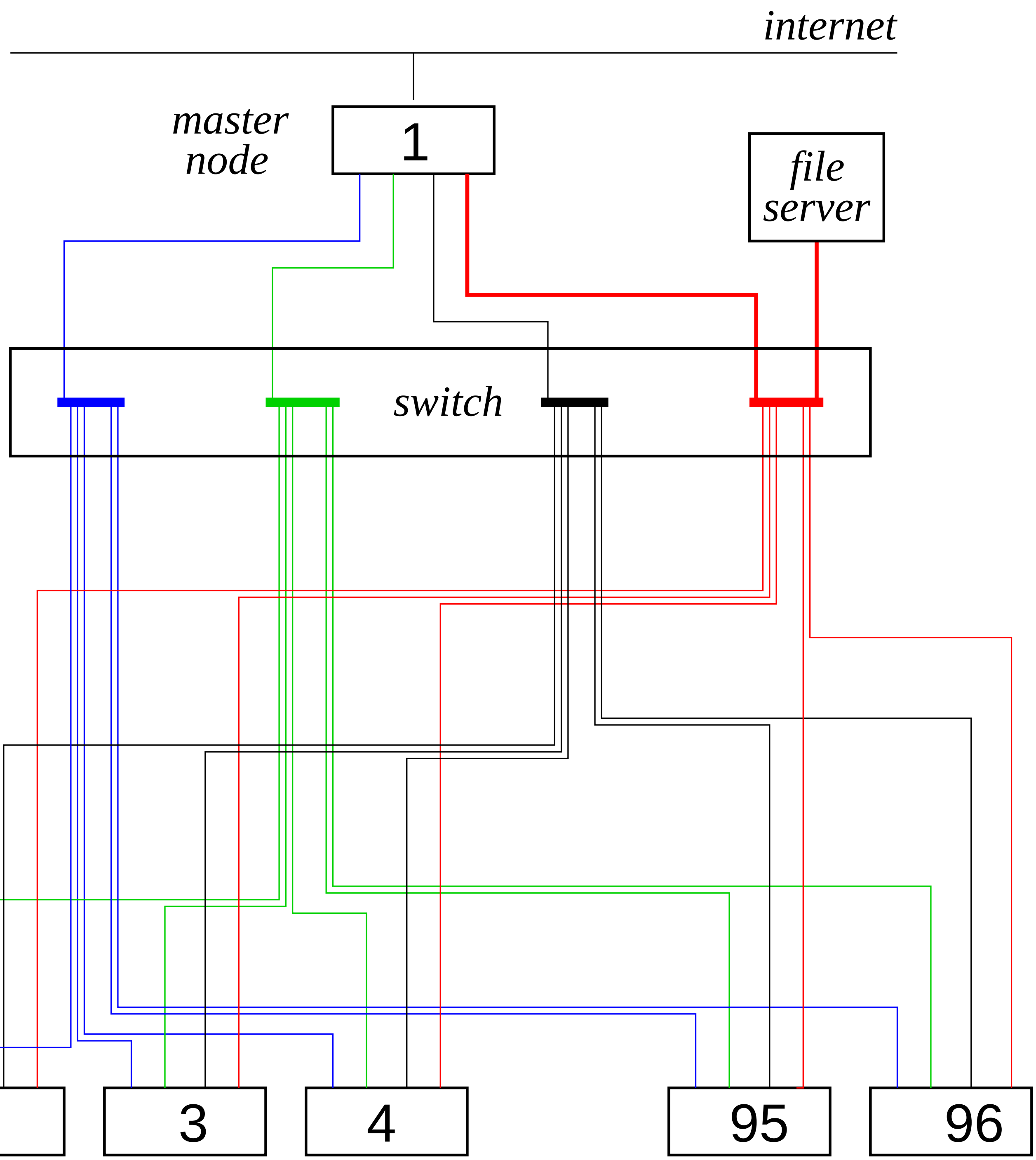
$\Rightarrow$ Linux channel bonding:

3-way channel bonded fast ethernet:

- need 3 times as many ethernet cards, switches
- low latency, good bandwith (270 Mbit/s)
- cheap

Problem: How do you connect a file server?
$\rightarrow$ additional network for NFS (fast ethernet)

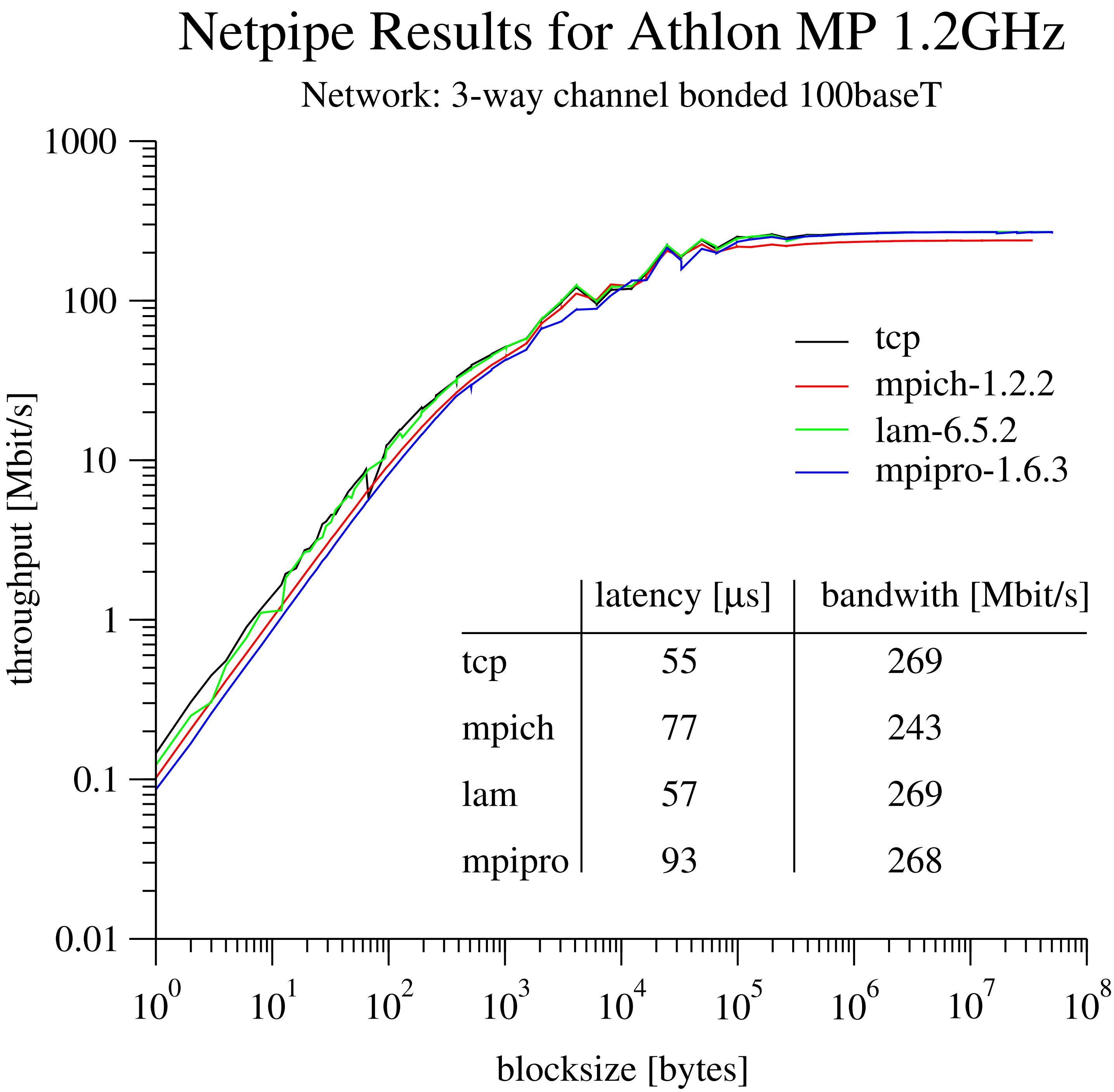
The Bugaboos: Network Topology



Network Performance

3-way Linux Channel Bonding

kernel 2.4.18



Installation

- master node
  - Linux
  - `/usr/local`

    → all cluster specific software
    * message passing library
    * scientific libraries
    * compilers

      ...
    * distribution for the other nodes
  - NFS export `/usr/local`

- all other nodes
  - Linux (bare OS)
  - kickstart installation:
    * about 4 minutes per node
    * can be done in parallel
    * completely unattended
    * node reboots and is fully configured

Running the Cluster

A word of caution:

> it may be not too difficult to build a cluster,
> however, it is difficult to run a cluster.

- power requirements, AC
- compile your own Linux kernel
- software optimization (libraries!)
- batch queueing/scheduling system
- monitoring

Tools:

`clsh`: run a command on cluster nodes
`clsh reboot`

Monitoring:

`ruptime`: load on cluster

`rsensors`: CPU temperatures, fan speeds

`bigbrother`: monitoring software
`http://www.bb4.com`

Security:

security of cluster $\Leftrightarrow$ security of master node

`ssh`	for connecting to master node
`rsh`	without passwords between nodes
`ipchains`	firewall on master node
`check-rpms`	for keeping RPMs up-to-date
	`http://www.sfu.ca/acs/security/linux/check-rpms.html`
	$\rightarrow$ `clsh check-rpms --update`

`sphinx.sfu.ca:/vol/vol1/distrib/redhat/<RHversion>`

patched RedHat distributions

`<RHversion>` = 6.2, 7.1, 7.2, and 7.3

Configuration:

every change in the configuration is also made in the kickstart file!

$\rightarrow$ reinstalling a node is easier than redoing all changes.

Parallel Programming

Problem: How can results calculated on one node accessed from another node?

$\Rightarrow$ MPI Message Passing Interface $\Leftarrow$

- standardized message passing system
- set of library routines
- portable
- latest version of the standard: MPI-2; however, available distributions support only MPI-1 with some extensions.
- MPICH
  `http://www-unix.mcs.anl.gov/mpi/mpich`
  LAM
  `http://www.lam-mpi.org`

You must change your program.

Two day workshop on MPI this fall at SFU!

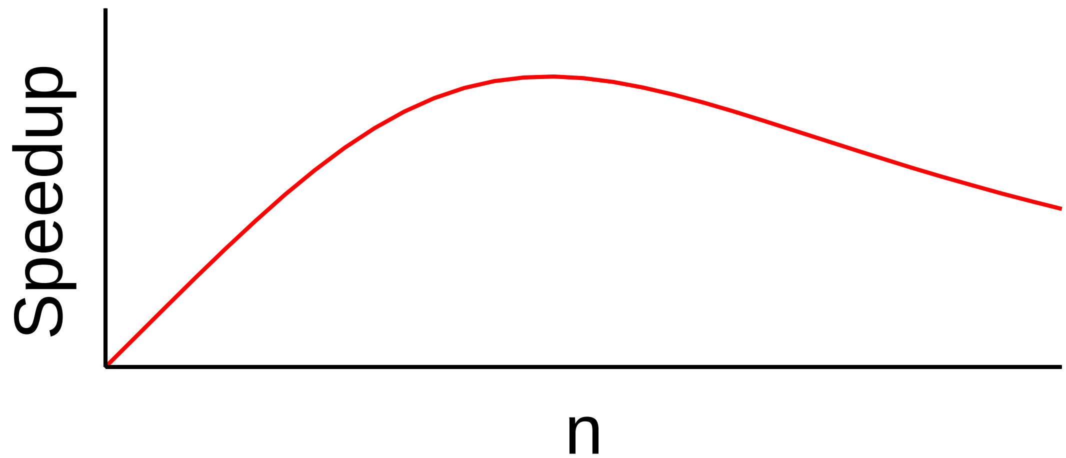
References

- Using MPI (2nd ed.), W. Gropp, E. Lusk, and A. Skjellum
- MPI – The Complete Reference, Vol. 1 (2nd ed.), M. Snir *et al.*

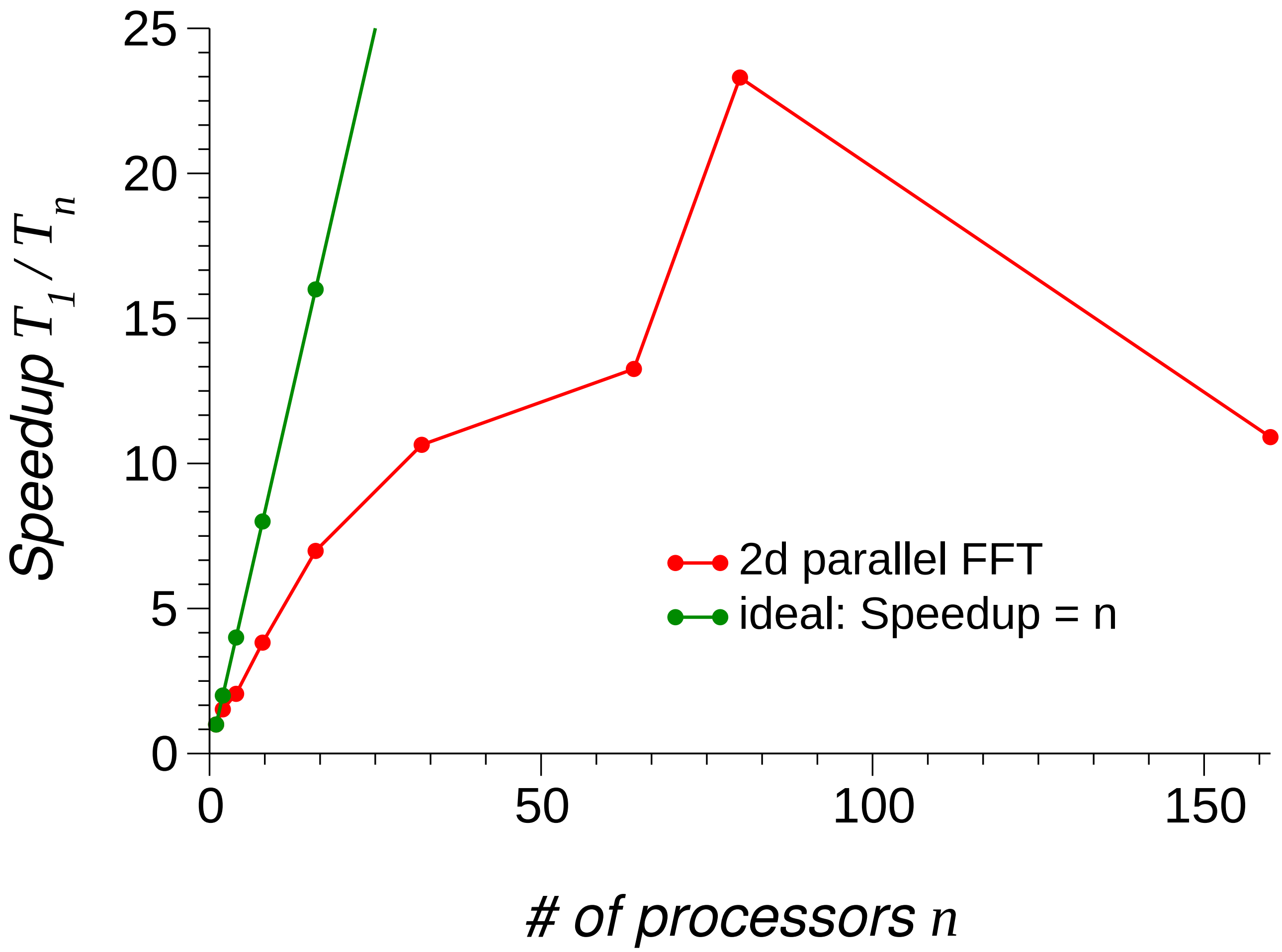
Performance

Linpack HPL benchmark: 144.6 Gflops

typical Speedup T_n/T_1:



more demanding: n^2 communication pattern



Summary

- Linux Beowulf clusters = low cost alternative to mainframe style computing
- Design depends on applications:
  - fast CPUs?
  - fast network?
- Installation $\rightarrow$ kickstart
- System administration nontrivial
- `beowulf@beowulf.org`
- huge selection of freely available programs
- buy some programs, e.g., Fortran compiler

Important: Learn MPI programming!

References

- http://www.sfu.ca/acs/cluster
- http://www.sfu.ca/acs/linux
- `beowulf@beowulf.org`