An Introduction to Cluster Computing

With ordinary things, do extraordinary things

Soma

Disclaimer:
The contents are not validated and the reliability is not checked However, efforts were put ensure that they are true to best of my knowledge

What is a cluster?

- A group of connected workstations/ computers that can handle the job/ task in a shared manner.
    - 1 + 1 = 3

What is cluster Computing?

- It represents an attempt to solve larger problems, or to solve problems in a more cost effective manner, than the more conventional systems of the time.

Speed – Due to Compute-bound problem and/ or I/O-bound problem, consider parallelism

What is parallelism?

- It all depends on the application

- CONCURRENT parts of the algorithm are those that can be computed independently

- PARALLEL parts of the program are those CONCURRENT parts that are executed on separate processing at the same time

- Computation & Implementation are NOT interchangeable

- Cluster Computing is a way to exploit (forced) CONCURRENCY!

Algorithm is not parallel. Hardware makes the concurrent parts of the algorithm RUN parallel

Why a Cluster/ Why not a Cluster ?

- Performance of Commodity & network hardware increase and their prices decrease, it is more practical to build parallel computers

- Buying single large CPU time: History
 Group off-the shelf commodities to build parallel computers: Present
 Home Super Computers: Future

- CPU speeds have been DOUBLING every 18 months
Processors speeds will NOT continue to do so after 2005

- It is possible to do several things at once with MTOS. This is natural parallelism that can be exploited by clusters

Parallel computing speeds up things anywhere from 2 to 500

Platforms for Computing

- Single-tasking OS: MS-DOS
- Multi-tasking OS: UNIX, NT using single CPU
- Multi-tasking OS with Multiple CPUs: UNIX, NT with more than one CPU
- Threads on Multi-tasking OS with Multiple CPUs:
- UNIX, NT with more than one CPU on the same motherboard

Many ways to implement parallel computing. Consider portability, scalability, performance, and COST before making the decision

Hardware Architecture:

Local Memory Machines that communicate through messages (Beowulf)
Shared Memory machines that communicate through memory (COW)

Software Architecture:

Using Messages sent between processors
Using Operating system threads

Locomotives (olden days) were connected in

- Shunt when you need speed
- Series when you want starting torque

A major classification of cluster computing
Cluster of Workstations (COWs) Vs Beowulf

Beowulf

- One (typically) server node, one or more client nodes connected via Ethernet (or some other N/W)
- Built of commodity hardware
  - PC running Linux
  - Standard Ethernet adapters & switches
- No custom hardware components
- Trivially reproducible
- Uses Message Passing Interface (MPC) and Parallel Virtual Machine (PVM)
- Clients are in the private N/W
- It is NOT a new software package or N/W topology or the latest kernel

Beowulf behaves like a single machine rather than many workstations

Eg.

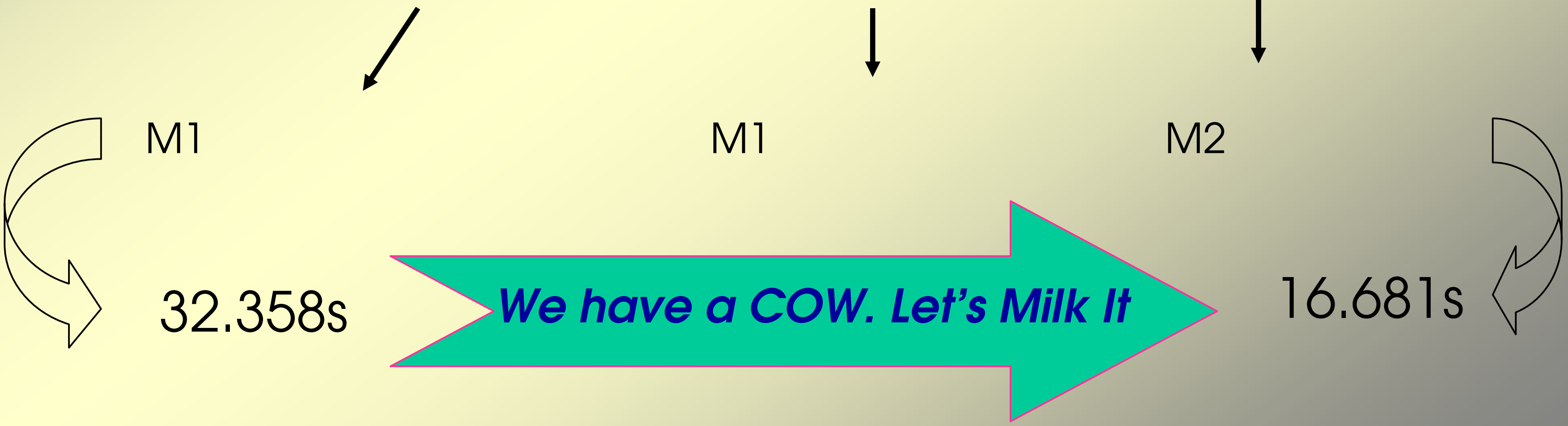
- We have Network File System (NFS) and NIS server

  Data is shared by all machines in the N/W
- We have machines of same configuration (not a significant point but eases the COW implementation)

  we can run the same code on these machines
- Use RSH (remote shell) to do the job. Don’t forget to put them in background

Example:

$$\sum_{i=0}^{200000000}\sqrt{i} = \sum_{i=0}^{100000000}\sqrt{i} + \sum_{i=100000001}^{200000000}\sqrt{i}$$



M1

M1

M2

32.358s

We have a COW. Let’s Milk It

16.681s

w.r.t SimSET

- Allocation of activity is concurrent
  - can be run on different machines
  - server doing the pre-processing to distribute the load
  - use bottom-up approach for load balancing
- No. of decays per cell/ grouped cells
  - use PRNGs with sufficiently longer period
  - search for PRNGs to be run on threads
- Cost analysis for installing/ maintain a Beowulf cluster
  - what hardware acceleration is possible/ feasible
  - TBD

Beowulf gives better price/ performance ratio & is dedicated for parallel processing. Suitable for specific applications → SimSET !!!

DO’s before clustering

- Determine the concurrent parts of the body
    - Number Crunching
    - I/O
- Estimate parallel efficiency
    - Good Old Trail & Error
    - Is it CPU limited/ Hard-Disk limited
    - Gut Feeling
    - A of DMA
- Describing the concurrent parts of the algorithm
    - Implicit (User)
    - Explicit (Compiler)

A Compute bound problem may work better with few faster CPUs!
An I/O bound problem may work better with more slow CPUs!!!

Resources/ Links

Beowulf Homepage http://www.beowulf.org
Extreme Linux http://www.extremelinux.org
The latest version of the Beowulf HOWTO http://www.sci.usq.edu.au/staff/jacek/beowulf
Building a Beowulf System http://www.cacr.caltech.edu/beowulf/tutorial/building.html
Linux Parallel Processing HOWTO http://yara.ecn.purdue.edu/\~{}pplinux/PPHOWTO/pphowto.Html
Cluster computing review http://www.npac.syr.edu/techreports/hypertext/sccs-748/cluster-review.html
Computing Resources for Macromolecular Crystallography at DND-CAT
http://tomato.dnd.aps.anl.gov/macrocryst/resources.html
Frequently Asked Questions. Tips, and Hints (IIT) http://www.iit.edu/~gigawulf/faq.html
Home Supercomputing with Linux http://www.cris.com/~rjbono/html/pondermatic.html
Useful Links http://shspc1.chem.psu.edu/pa/int/linuxcluster.html
Papers http://www.beowulf.org/papers/HPDC96/hpdc96.html
http://www.beowulf.org/papers/AA97/aa97.ps
http://www.beowulf.org/papers/IPPS96/ipps96.html
Price watch for hardware components http://www.pricewatch.com/ , http://computers.cnet.com/
Why IBM is doing Linux Clusters http://www-1.ibm.com/servers/eserver/clusters/whitepapers/linux_wp.html

GE CRD Whitney Information Services eBook Collection http://www.netlibrary.com/index.asp
How to build a Beowulf system
Software Updates, Archives, Documentation on Beowulf: http://www.beowulf-underground.org/

Now Beowulf is hosted by http://www.scyld.com/

We don't offer full support for your Linux clustering requirements!!

IBM Linux Cluster

http://www.cineca.it/HPSystems/Resources/LinuxCluster/

Peak performace: 145 Gflops/s
Processor: PIII 1.1333GHz
No. : 128
Compilers: GCC C, C++, F90
OS: LInux

Acknowledgements

Jack Radajewski & Douglas Eadline
helpdesk
and
WWW

80% of this presentation was prepared using CAP Tools
and is 100% recyclable

No Surprise if you see
“building super computers at home for dummy’s”