Bochum Linux Blues

or

How to keep more than 150 Computers and 20 Scientists up-and-running

Axel Kohlmeyer

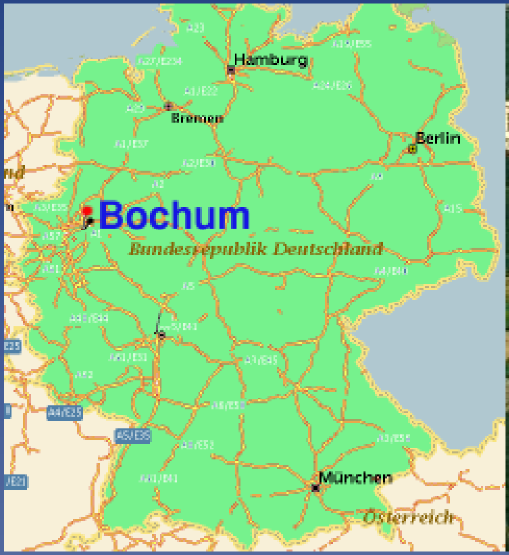
Lehrstuhl für Theoretische Chemie
Ruhr-Universität Bochum
<axel.kohlmeyer@rub.de>

May 2003

Anatomy of the Talk

- Introduction
- Plans and Results
- The Big Machine
- Some Challenges and Solutions
- Lessons Learned

Ruhr-Universität Bochum



- founded 1965, 8th largest university in germany
- 55.000 students, 20 faculties, 460 professors
- 2000 scientific staff, 2100 administrative staff

Timeline

- fall 1999: Prof. Marx replaces Prof. Kutzelnigg
  => complete change of research focus

- 2000: remodelling
  - new power wiring, cooled computer rooms
  - new network (100BaseTX switched, VLAN)

- 2000-2002: buying the new hardware
  - all desktops and compute servers new
  - large parallel machine(s) in fall 2002

Boundary Conditions

- CPMD == black hole for CPU cycles
  => use money most efficiently
- only small annual budget
  => low maintenance costs (replacement parts)
  => avoid annual licenses & support contracts
- no technical staff
  => simple maintenance, robust setup

Concept

- use linux wherever possible,
  selected Windows/VMware machines
- uniform filesystem namespace (NFS, automounter)
- home filesystems local on desktops
- use desktop machines also as (compute) server
- licensed software: FORTRAN90 compiler, VMware

Parallel Machines

- 1 Top500 parallel machine:
  64 nodes, dual athlon, SCI-network

- 1 ’small’ parallel machine:
  12 nodes, dual athlon, SCI-network

- 3 Ethernet connected clusters:
  - 6 nodes, dual athlon, 1000BaseTX
  - 8 nodes, single athlon, 1000BaseTX
  - 8 nodes, single athlon, 100BaseTX

Serial Machines

- 6 alpha workstations (large memory/disk):
  2 dual ev68, 2 quad ev67, 2 single ev56/ev6
- 25 desktop machines:
  768-1024 MB RAM, 800-1533 MHz athlon
- 12 server machines:
  1536 MB RAM, 900-1533 MHz athlon
- 5 old and ’slow’ machines:
  NIS, SMTP, DNS, FTP/HTTP, batch, firewall, testbed

Stability and Utilization

- SCI-cluster
  - average uptime: 90 of 91 days
  - cputime usage: 98 percent
- server machines
  - average uptime: 83 of 91 days
  - cputime usage: 80 percent
- desktop machines
  - average uptime: 40 of 91 days
  - cputime usage: 85 percent

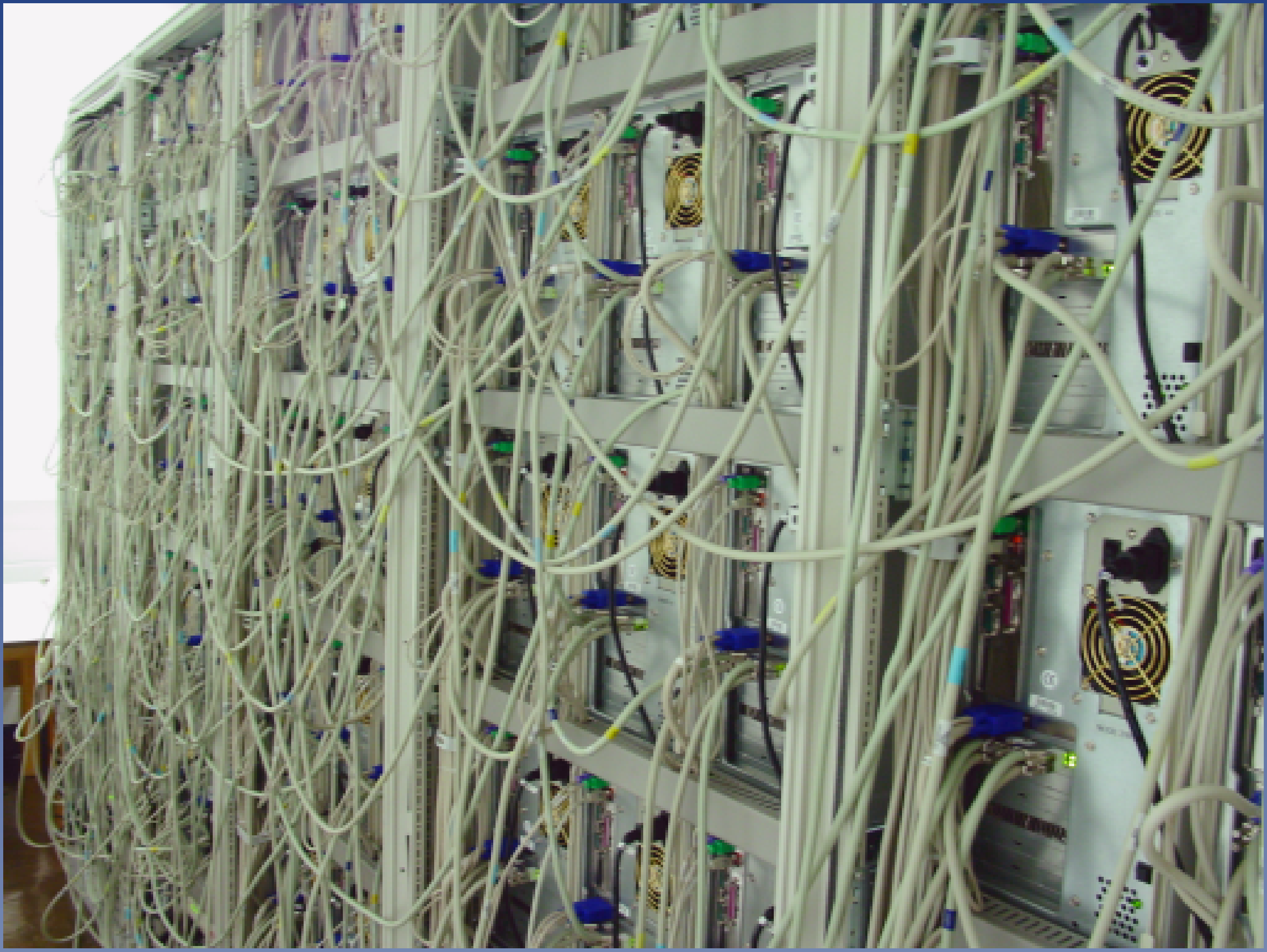
What do you need to get a Top500 Machine for 300 kEUR?

- luck: application runs best on the same machine as LINPACK
- hard work: find optimal combination of cpu and number of nodes
- hard work: optimize benchmark parameters and LAPACK/ATLAS
- paranoia: force (not ask) suppliers to build the machine you want
- paranoia: check quotes extremely careful and find a supplier that cares about you
- hard work: be willing to do most of the service yourself (note: if you select the 'right' machine this is no problem)

SCI Machine: the good, . . .

SCI Machine: . . . the bad, . . .

SCI Machine: . . . and the ugly



Why this machine?

- dual AMD Athlon MP 1600+ CPU with AMD 760MPX chipset:

  - AMD Athlon MP cheaper than Intel Xeon => more nodes

  - Intel Xeon needs expensive RAMBUS memory for high bandwidth

  - memory is performance bottleneck, not much gain with faster CPU

  - AMD 760MPX had best PCI-X throughput in PC class machines

- Dolphin SCI 2D-Torus:

  - cheaper than Myrinet => more nodes

  - no switch needed, better fault tolerance

  - no exclusive root access to device needed

  => better protection from application crashes

Uniform Filesystem Namespace

- same (elaborate) partitioning scheme everywhere
- only non-system filesystems NFS-exported (unique names)
- transparent filesystem access through automounter
- distributed **`/home`** filesystem through 2nd automounter
- no quotas needed, users have to take care
- flexible canonical paths to shared files through dummy accounts (e.g. **`backup, redhat, cpmd, ...`**)

Simple, Fast and (Somewhat) Fault Tolerant Backup Scheme

- no technical staff => low effort backup
- backup to large RAID-5 IDE-disk(s) => fast, high-capacity
- GNU tar backup => portable, easy to restore file format
- only ’essential’ files in backup sets:
  - ⋆ per-machine ‘system backup’: **`/etc /var/spool /root`**
  - ⋆ per-user ’home backup’ with per-file size limit
- listed incremental backups => daily, weekly, monthly
- backup of scratch directories delegated to users

Backup Client/Server Protocol

Server	Client
read and parse config	read and parse config
	register client in backup spool
loop over list of clients	
start backup	perform all backups for client
next client	
send email report	

- read local, write to NFS => track inodes, fast search
- small program(s): 1000 lines of perl code
- one configuration file for clients and server

Lessons Learned

- install only software that is really used
- find simple solutions, handle exceptions manually
- find convenient configurations for common cases
- identify separable tasks, automate extensively
- prepare solutions for (likely) failures ahead of time
- favor solutions that reduce effort in the long run
- educate your users, make them responsible

Thanks

- Prof. Dr. Dominik Marx
- Ruhr-Universität Bochum
- Priv. Doz. Dr. Eckhard Spohr
- past and present Colleagues
- Linux/GNU Community