



Monitoring and administration of openMosix clusters

by Matt Rechenburg

Legal openMosix is Copyright (c) 2002, 2003 by Moshe Bar.

Mosix is Copyright (c) 2002 by Amnon Barak. The Mosix is a trademark of Amnon Barak.

Linux is a Registered Trademark of Linus Torvalds.

openMosix is licensed under version 2 of the GNU General Public License as published by the Free Software Foundation.



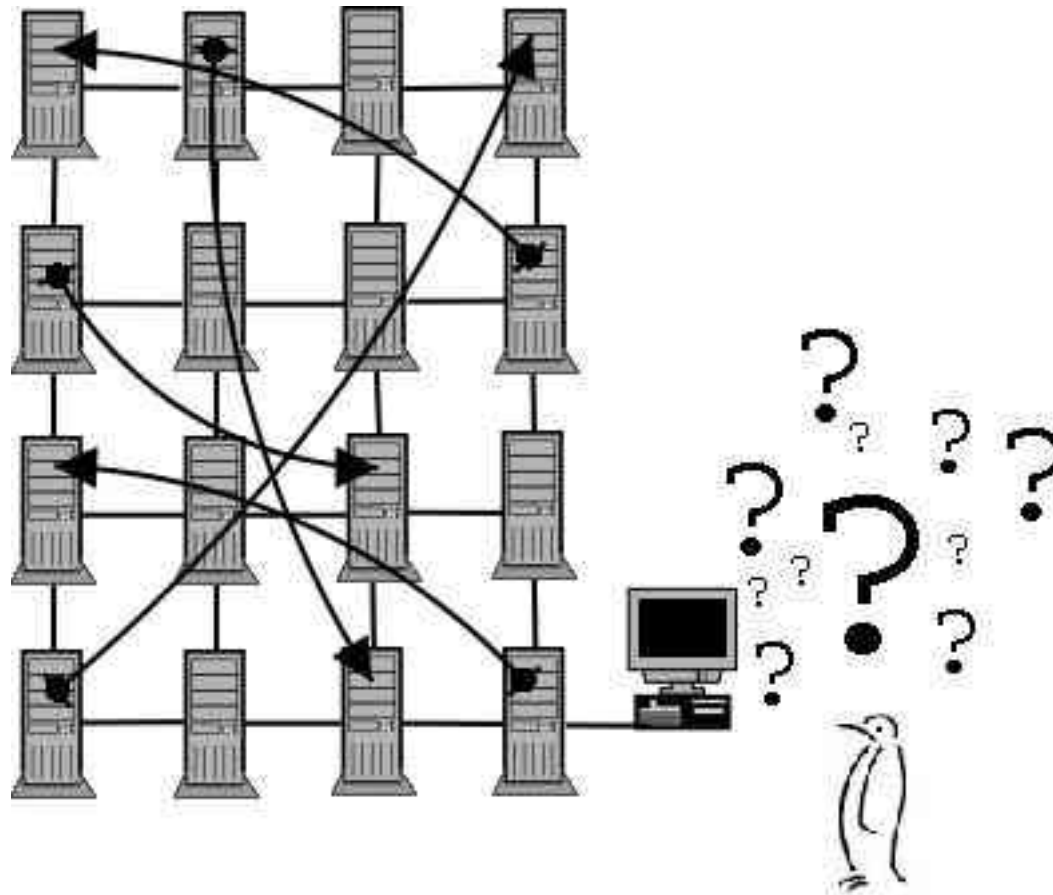
Matt Rechenburg

- Member of the openMosix community
- openMosix developer -> openMosixview
- Freelancer, working for Qlusters



- Monitoring openMosix
- administrating openMosix
- running applications on openMosix
- the openMosixview project
- the gomd project (featuring Gian Paolo)
- usefull utilities and examples
- Time for discussion

Monitoring in general.. exactly, what is going on ?





Monitoring in detail

the openMosix commandline monitoring utilities

- mosmon
- the /proc/interface
- contributed tools -> mps + mtop



```
root@demo: ~ - Befehlszeile 3 - Konsole
1395 1 0 S 0:00 /sbin/mingetty tty1
1398 2 0 S 0:00 /sbin/mingetty tty2
1399 3 0 S 0:00 /sbin/mingetty tty3
1400 4 0 S 0:00 /sbin/mingetty tty4
1401 5 0 S 0:00 /sbin/mingetty tty5
1402 6 0 S 0:00 /sbin/mingetty tty6
1408 ? 0 SW 0:00 (mfs_server)
1404 ? 0 SW 0:00 (mfs_server)
10710 ? 0 S 0:01 /usr/sbin/sshd
10718 ? 0 S 0:01 bash
11078 ? 0 S 0:00 /usr/sbin/sshd
11082 ? 0 S 0:00 -bash
11228 ? 0 S 0:00 /usr/sbin/sshd
11232 ? 0 S 0:00 -bash
11428 ? 0 S 0:00 /bin/sh /bin/mosrun -b openmosixview
11481 ? 0 S 0:00 openmosixview
11544 ? 0 S 0:00 /bin/bash ./start_distkeygen.sh
11551 ? 1 S 0:00 (distkeygen)
11552 ? 0 R 1:21 ./distkeygen
11553 ? 1 S 0:31 (distkeygen)
11554 ? 2 S 0:43 (distkeygen)
11555 ? 1 S 0:41 (distkeygen)
11556 ? 2 S 0:55 (distkeygen)
11557 ? 1 S 0:31 (distkeygen)
11558 ? 0 R 0:00 nps ax
1251 ? 0 S 0:00 (atd)
328 ? 0 S 0:00 (rpc.statd)
298 ? 0 S 0:00 (portmap)
[root@demo root]#
```




```
rut@dem: ~ - Befehlszeile 3 - Konsole
```

9:08pm up 1:06, 1 users, load average: 1.26, 0.62, 0.17
58 processes: 56 sleeping, 2 running, 0 zombie, 0 stopped
CPU states: 277.4% user, 20.5% system, 0.0% nice, 0.0% idle
Mem: 255444K av, 153348K used, 102096K free, 0K shrd, 272K buff
Swap: 522104K av, 0K used, 522104K free, 125610K cached

PID	USER	PR	NI	SIZE	RSS	SHARE	STAT	N#	%CPU	%MEM	TIME	COMMAND
11552	root	18	0	972	972	784	R	0	81.6	0.3	1:46	distkeygen
11556	root	18	0	456	456	268	S	2	51.3	0.1	1:10	distkeygen
11554	root	18	0	456	456	268	S	2	49.7	0.1	0:58	distkeygen
11553	root	18	0	792	792	716	S	1	35.6	0.3	0:41	distkeygen
11553	root	18	0	456	456	268	S	1	34.0	0.1	0:42	distkeygen
11557	root	19	0	792	792	716	S	1	33.3	0.3	0:42	distkeygen
11564	root	13	0	880	880	724	T	0	4.1	0.3	0:00	ntop
1103	root	9	0	1300	1300	1112	S	0	2.8	0.5	1:12	qcnssd
11453	root	9	0	7652	7652	5756	S	0	1.2	2.9	0:03	openmosixview
11228	root	9	0	1920	1920	1624	S	0	0.9	0.7	0:00	sshd
1	root	8	0	480	480	120	S	0	0.0	0.1	0:07	init
2	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	kernel
3	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	kernel
4	root	18	19	0	0	0	SWN	0	0.0	0.0	0:00	kssoftirqd CPU0
5	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	kswapd
6	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	bdflush
7	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	kupdate
11	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	khubd
269	root	9	0	0	0	0	SW	0	0.0	0.0	0:02	rpciod
270	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	lockd
276	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	nvfs main server
278	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	nvfs_gc



The openMosix API

- user + programming interface to openMosix
- provides functions to get/set values
- Available in several programming languages



Administering openMosix

the openMosix commandline administration utilities

- mosctl (basic administration)
- mosrun (running applications)
- setpe (configure cluster-map)
- migrate + miggroup (manual process migration)



User + account management

- using NIS
- home directory on oMFS/NFS
- user permissions in the cluster



Time synchronisation

- using NTP
- Using rdate



ssh/rsh access to the cluster-nodes

- not needed for openMosix at all
- needed for openMosixview
- usefull for administration
- security aspect in untrusted networks



Checkpointing on openMosix

- install chpox and add the chpox module
 - register processe and its libraries
 - sequential dump process state to file(s)
 - restore process from dump-file
- > full-migration of processes possible



openMosix calibration & tuning

- io and/or cpu intensive applications
- decay value / load-statistics
- speed value
- locking processes



openMosix hardware tuning

- increasing the network bandwidth + latency
- increasing the processor power
- increasing the number of nodes



Running applications on openMosix

- decentralized concept -> no master-node
- applications installed once on one node
- applications can be started on any nodes



Application data / storage

- using oMFS
- using NFS
- using ogfs/GFS
- using lustre

--> openMosix and oMFSDFSA



Parallel applications on openMosix

- consists of more than one process
- starts processes on every cluster-nodes
- uses own communication mechanism
- mostly uses special parallel library



running PVM applications on openMosix

- master/slave concept of PVM applications
- static load/process assignment by PVM
- dynamic loadbalancing by openMosix
- minimizes the configuration needs for PVM



MPI applications on openMosix

- MPI standart and implementations
- similar master/slave concept as PVM
- multiple program single data (MPSD)
- limitations and common pitfalls
- openMosix patch to MPICH



openMosix and sequential applications

- parallelization of the problem to solve
- splitt into subtasks by splitting data
- run as MPMD (multiple program multiple data)
- forking using the openMosix API

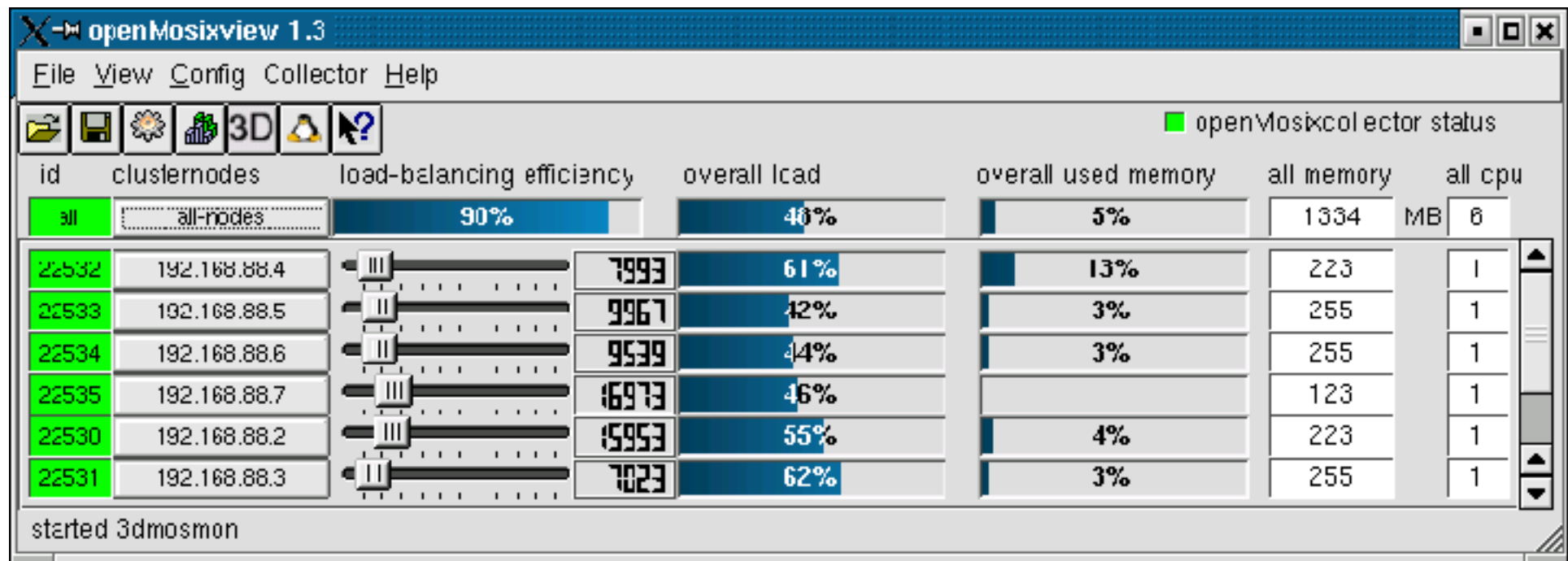


the openMosixview project

- intuitive GUI
- supports all common used actions
- QT based user interface
- 8 usefull monitoring applications
- easy to install

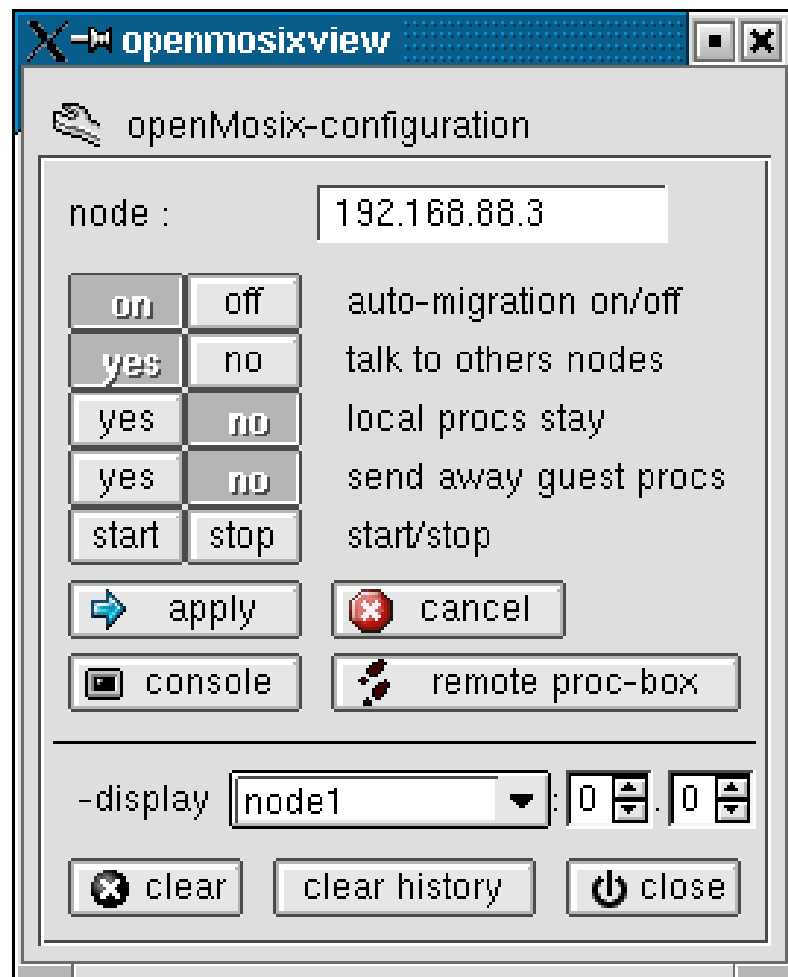


the main openMosixview application window



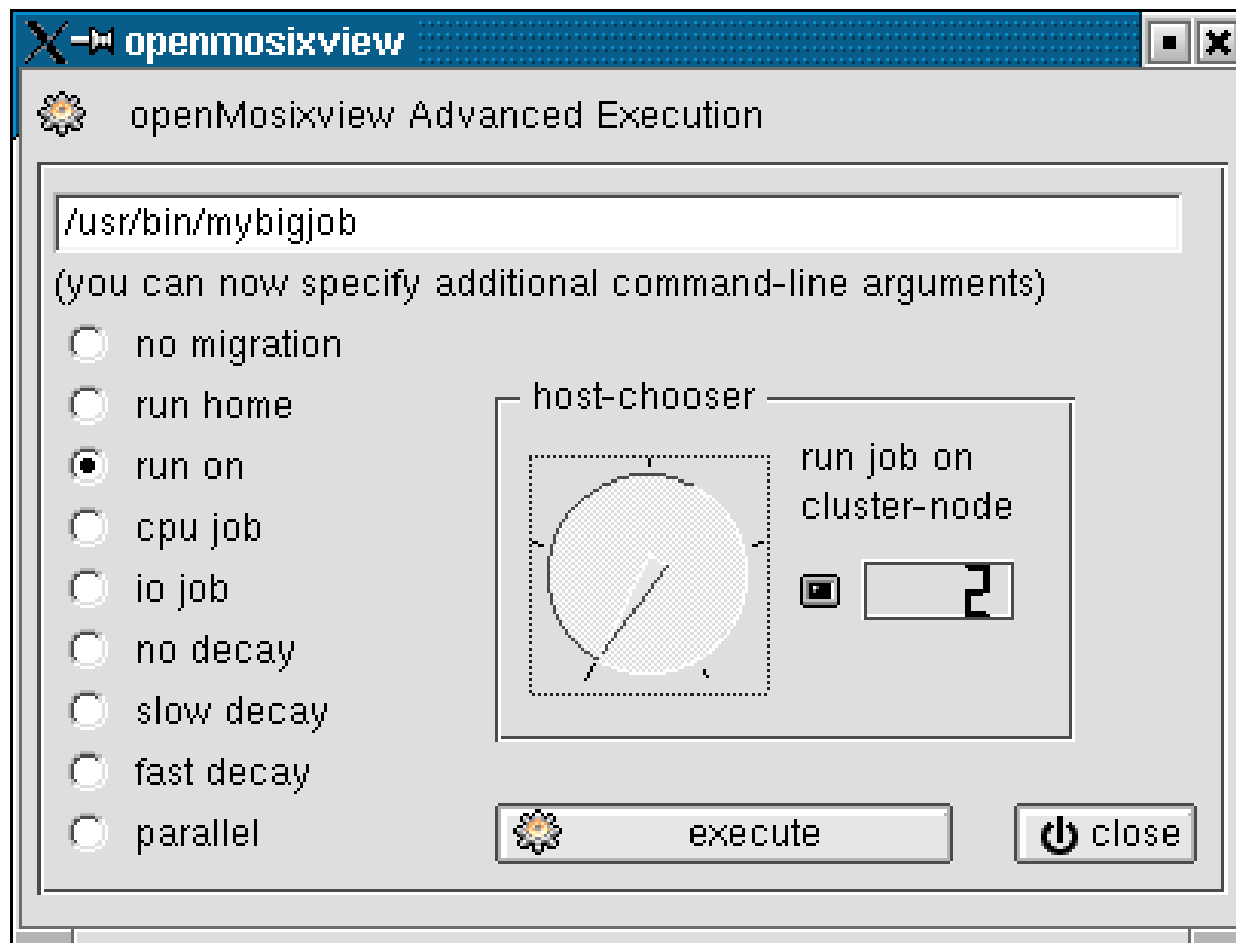


the configuration dialog





advanced execution





openMosixprocs, a graphical "top"

processes on node1

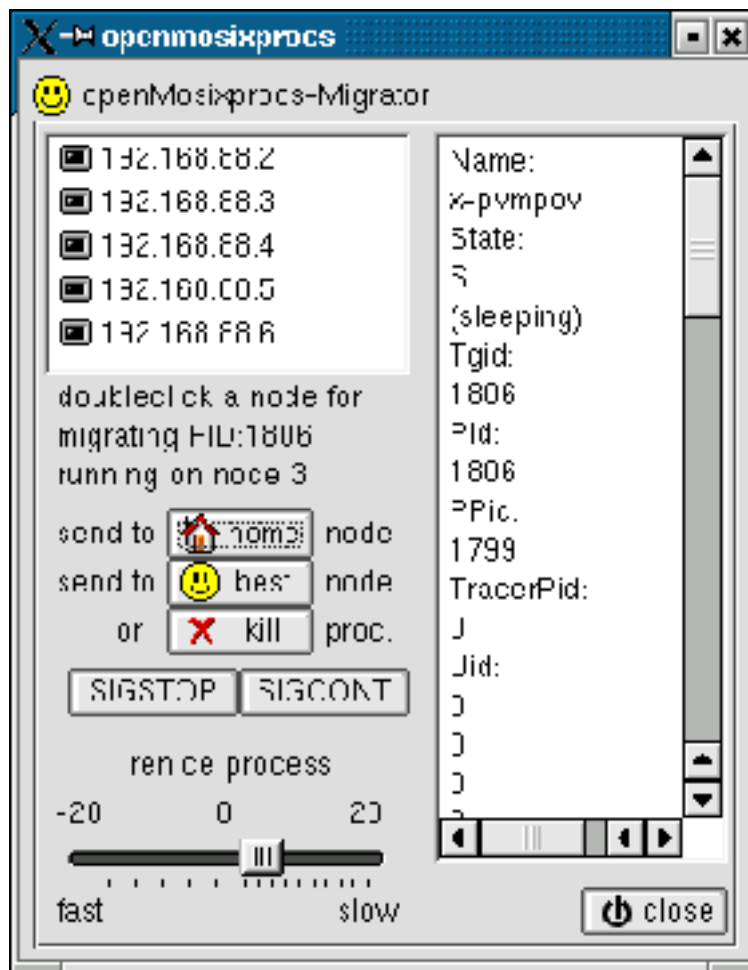
refresh: all processes last managed process

pic	r#	lock	rmigs	sta.	cmd line	nice	UD
12075	22535	0	2	S	/distkeygen	0	L
12074	22535	0	3	S	/distkeygen	0	C
12072	22535	0	0	S	/distkeygen	0	C
12063	22534	0	3	S	/distkeygen	0	C
12063	22533	0	3	S	/distkeygen	0	C
12007	22530	0	0	S	/distkeygen	0	C
12077	22531	0	2	S	/distkeygen	0	C
11986	22531	0	3	S	/bin/cash	0	L
12073	22530	0	1	S	/distkeygen	0	C
12071	22530	0	1	S	/distkeygen	0	C
12063	22530	0	2	S	/distkeygen	0	C
983	0	1	0	S	/usr/sbin/cfd	0	C
647	0	1	0	S	xfs	0	43
652	0	1	0	S	cmd	0	C
632	0	1	0	S	adm	0	L

manage proc from remote 67 processes on this system quit

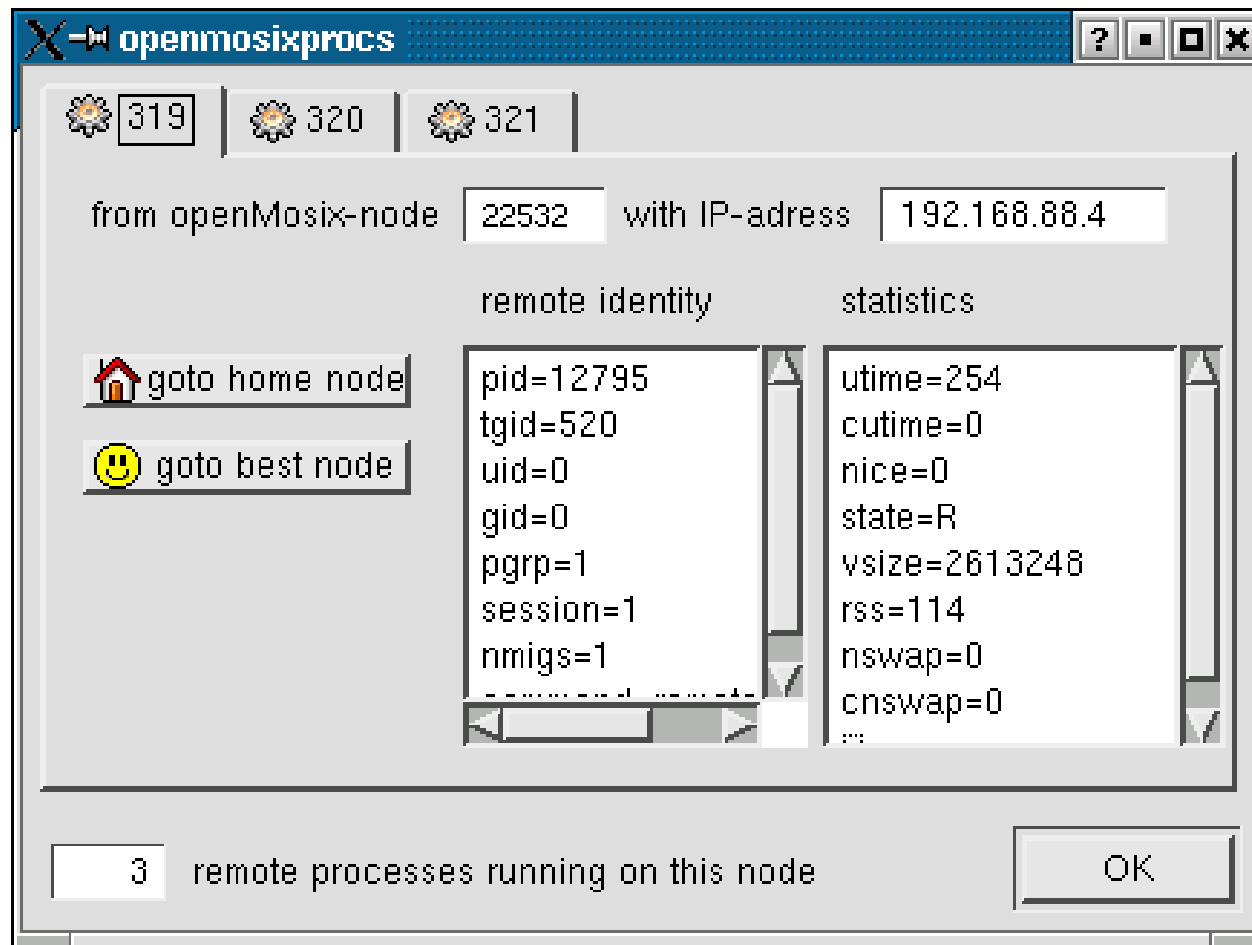


migration dialog



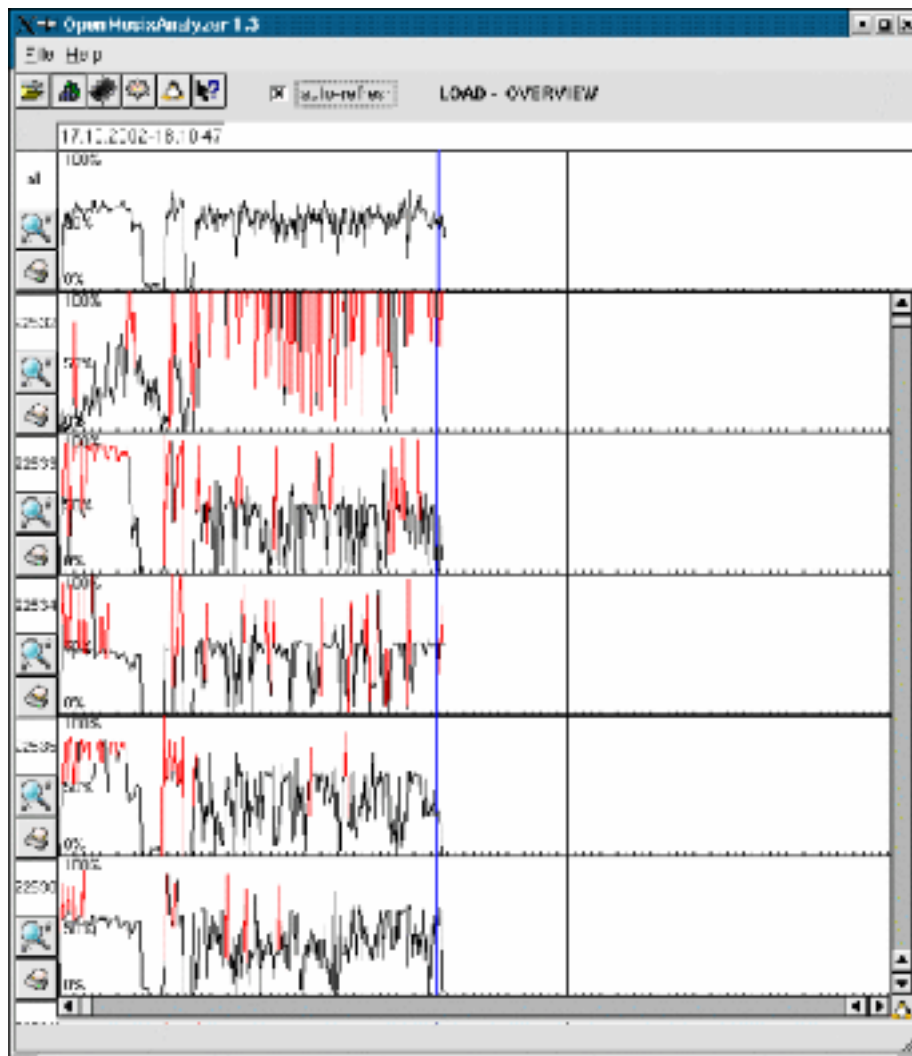


About remote processes



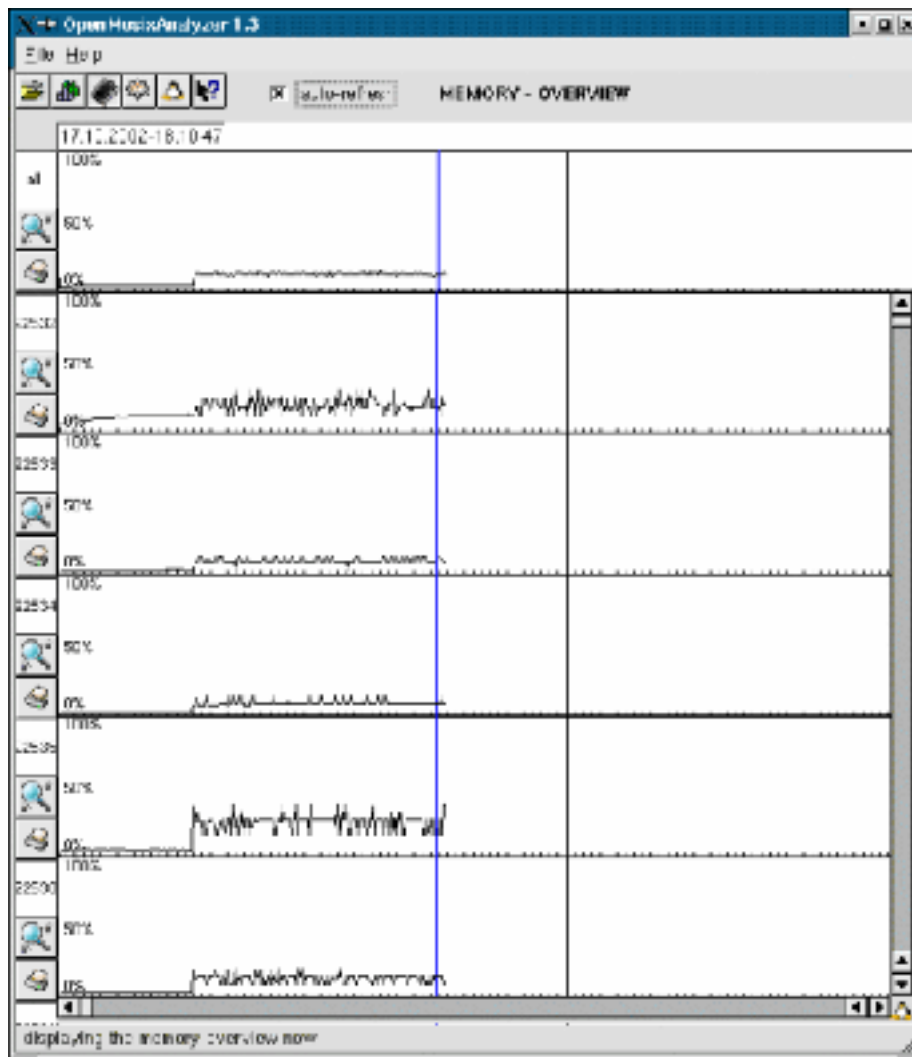


the openMosixanalyzer (load)



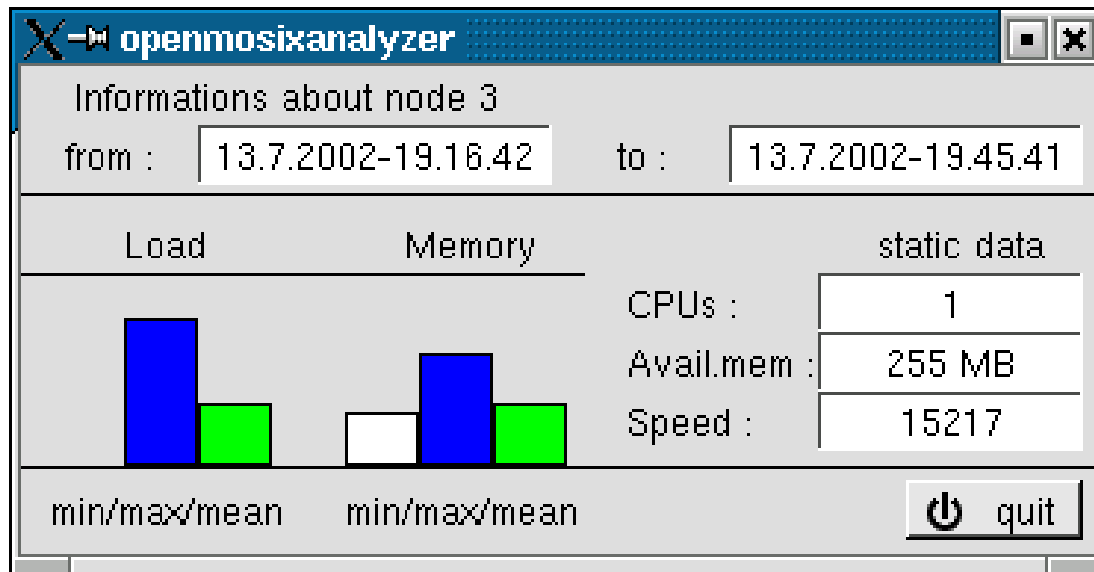


the openMosixanalyzer (memory)



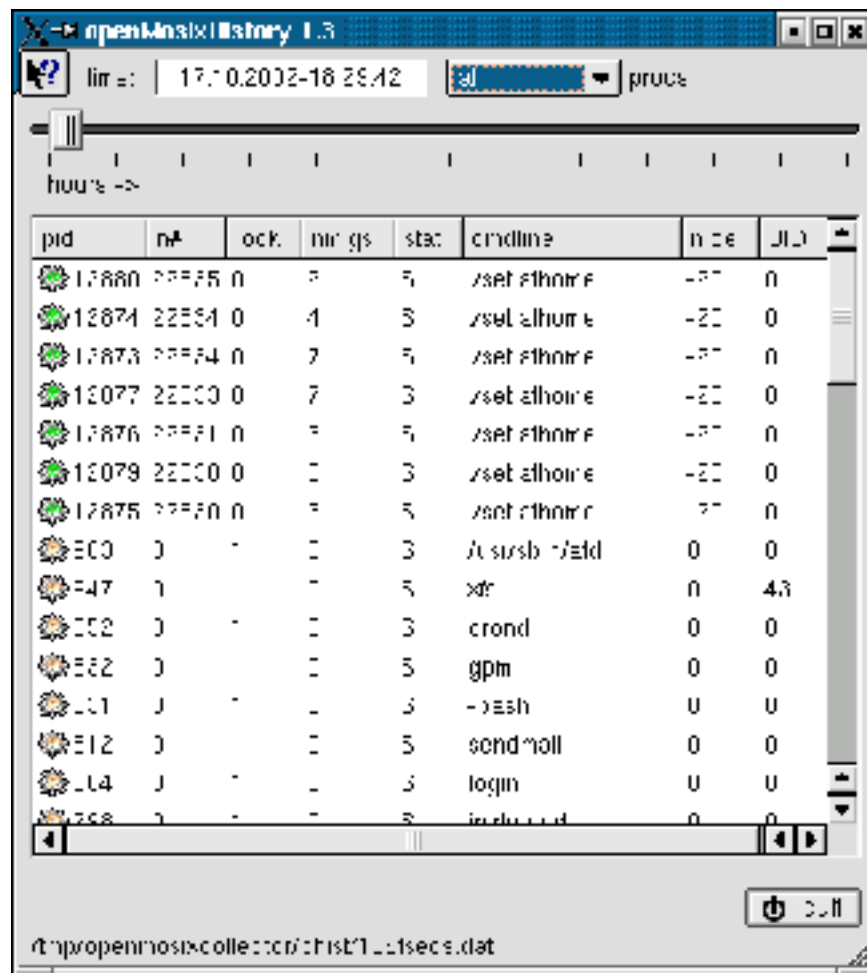


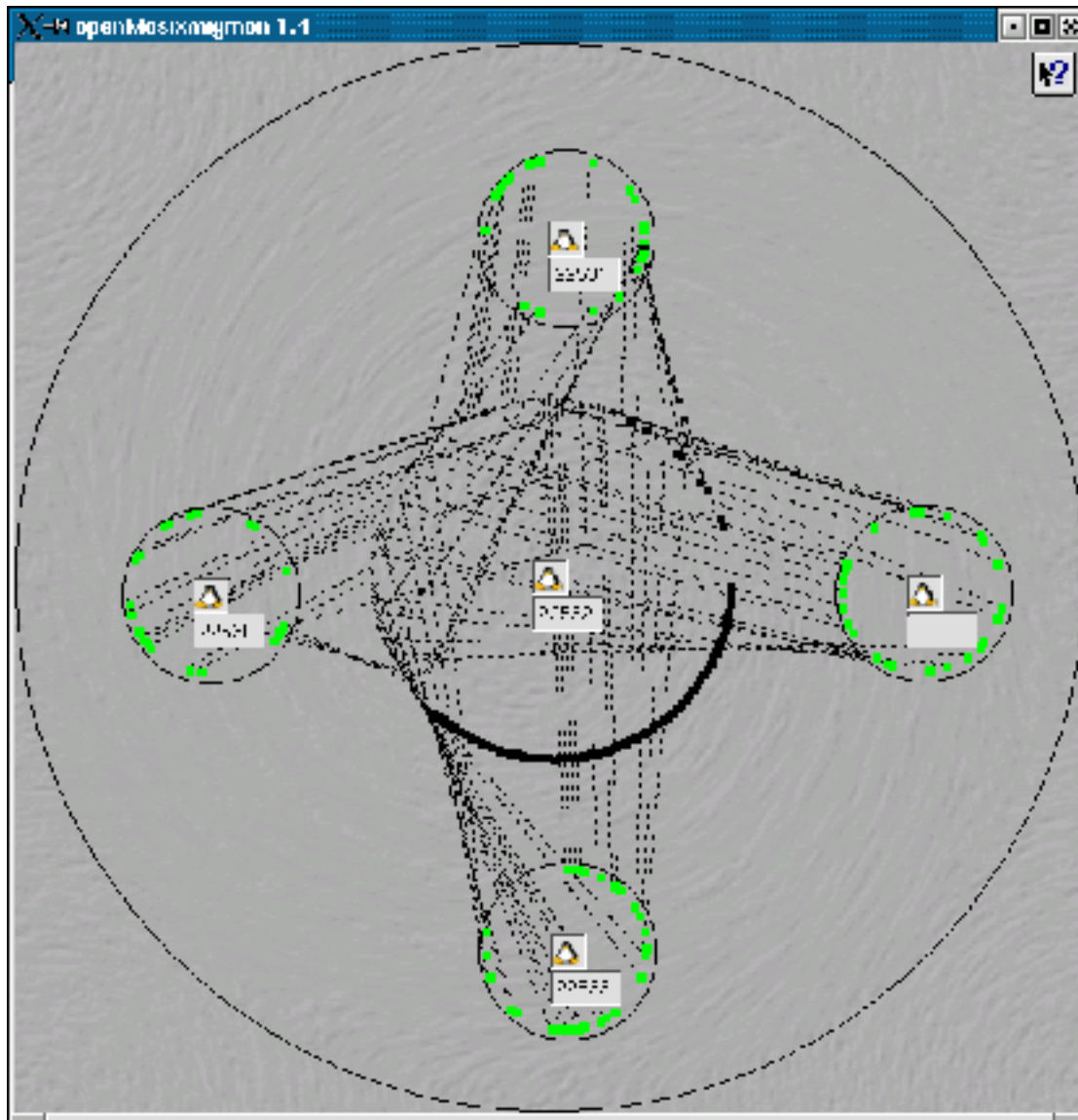
node overview





openMosixhistory

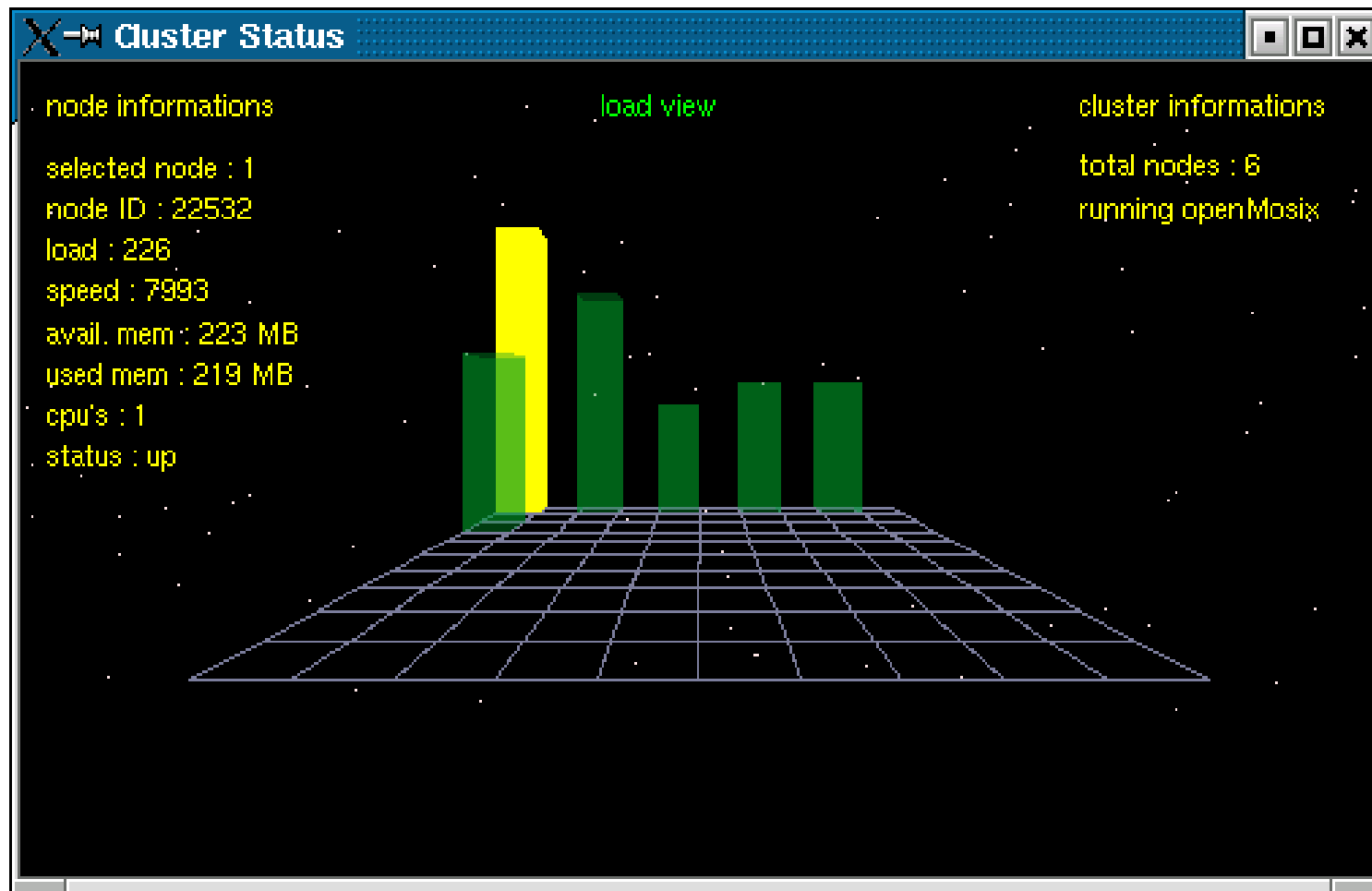




the openMosixmigmon

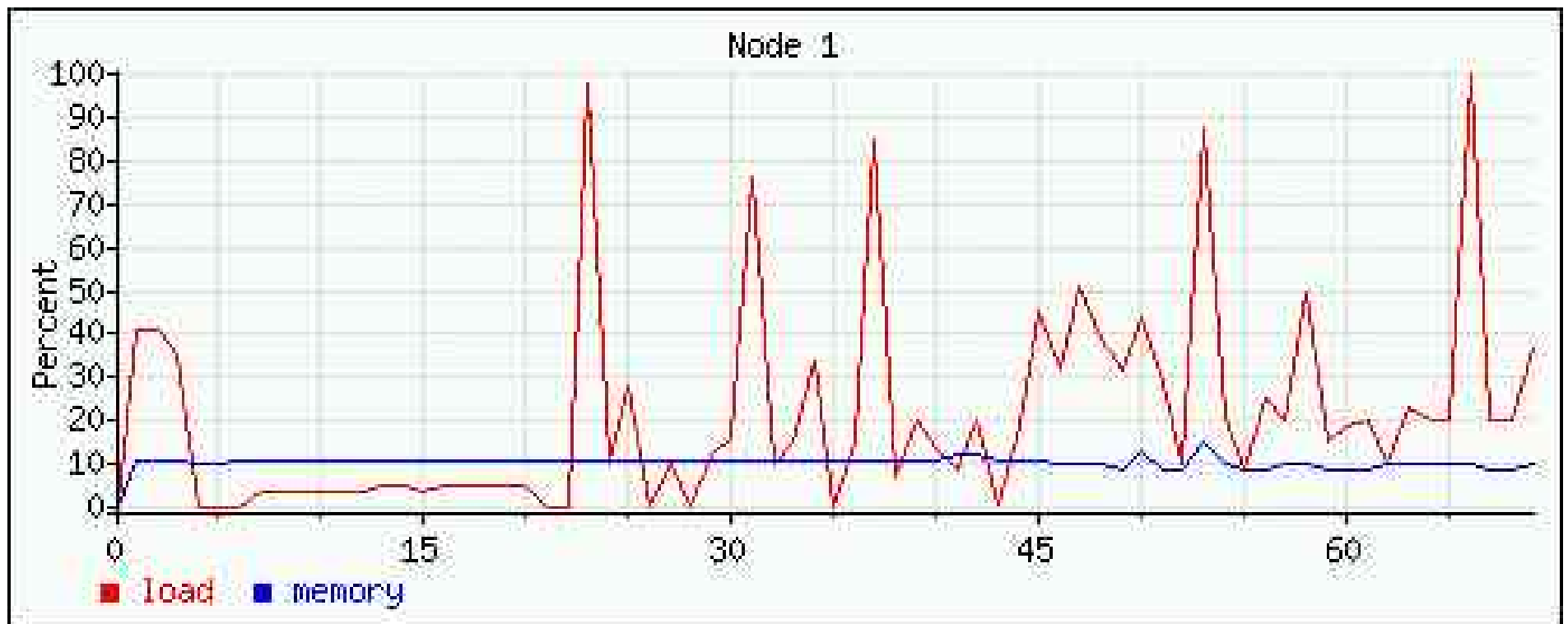


3dmosmon by Johnny Cache





openMosixWebview by Ramón Pons Vivanco





additional usefull utilities

- oMApplet
- KludgeKollection
- wmomload
- various bootable openMosix CD's
- openMosixLoaf
- Gentoo and Debian packages
- openMosixblaster
- and many more



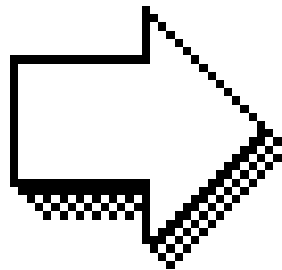
summary and conclusion

- transparent clustering for users and administrators
- easy to control and monitor, lots of tuning options
- provides speedup for almost any application
- lots of contributed projects and tools from the community
- one of the most active linux-clustering project



Question time now

?



!





Thank you !! :)

and thanks to the community,
to the Fosdem-organisators and
the sponsors

-> Qlusters, Myrinet, Unesco, NASA, AMD,
Johns Hopkins Medicine, Dolphin,
University of Arkansas, Democritos,
the university of Pisa
not to mention Debian + Gentoo

Happy clustering !















