

CoMosix state

What's done and what would be done in CoMosix.

CoMosix now

CoMosix is built on a coLinux 0.6.1 and uses kernel 2.4.26 with openMosix patch. CoMosix depends on WinPCap while running but not for the installation process. Indeed the install is independent of any third-party software. CoMosix doesn't integrate any root file system, it uses NFS share to mount the root file system at boot time, so the system is easier to manage and it's completely similar on every CoMosix node. The CoMosix init script create necessary writeable files and directory in RAM.

CoMosix kernel

CoMosix uses a vanilla kernel 2.4.26 patched with coLinux and openMosix. Some tests have been done with migshm patch but is not used in actual CoMosix kernel. Root file system over NFS and RAM disk support are also compiled into the kernel.

CoMosix distribution

CoMosix uses a GNU/Linux Debian unstable as distribution. It allows to easy install packages, with Debian tools like apt-get, and to use standard configuration tools provided by Debian.

CoMosix installer

CoMosix installer is free of any third-party software. It's build with NSIS and include network card detection, NFS configuration, additional coLinux block disk and memory usage configuration. All this is done by an ini file provided to the installer. The coLinux configuration file is automatically generated by the installer and MAC address is randomly generated and always start with 00:FF, usually used for TAP device.

CoMosix tomorrow

In order to have a *useful* system built upon CoMosix there are several things that should be done. The actual state of CoMosix allows to use it as an openMosix cluster without known limitations, but we need more than a openMosix cluster. We want to use software like mpiBlast and a complete MPI environment is needed. So here is a *todo* list that would give a glimpse of the evolution of CoMosix.

Users creation

In CoMosix it's not possible to create new users as the home directory is not kept. But we need to create users for use LAM (it can't be run with root).

Complete LAM environment

LAM is an implementation of MPI. In order to use it we need to build and install it on the root file system. Then it would be nice to make LAM using mosrun to start any MPI software. Doing so allow us to follow the state of the CoMosix cluster as it lost or gain others nodes.

GenBank fully accessible

GenBank is a genetic database. It's a kind of reference in genetic science and often used to blast sequence against it. Each node must have an access to it.

tmpfs

Actually we use RAM disk device to have some point in read-write mode in the root file system. But RAM disk device take an amount of memory and release it when the system is going down. The temporary file system (tmpfs) is a bit different as it does what RAM disk device does, but doesn't lock any specified amount of memory. It just grows as more space is needed. We could improve the memory usage by this way and so more memory would be available for applications.

Networked kernel

The CoMosix kernel is tied into the installer. It means that we must reinstall completely CoMosix just to change the kernel. In a better world, the kernel could be changed just by overwriting a file on a network share. Thus we could change hundred of nodes in just five or ten seconds.

Single control point

The whole cluster is not actually easy to control. In order to change a configuration file we change it on the root file system (this is a single control point), but then we need to go on every node and restart the service or even CoMosix. That's too much pain. A simple mouse click or a single command should do that for us.

GenBank update

In order to keep GenBank up to date, we need to download it every month, every week or even every day. But that should be completely automatic, just set a the cron.

SSH with private/public keys

SSH authentication done with private/public keys would be useful for every task on every node.

Install mpiBlast on the cluster

mpiBlast is very useful and we need it.

Where are we now ?

Working on CoMosix lead us to success on some points, but some others are still not working or badly working.

Users creation

Now there's a possibility to create users and keep their home directory in a specified state. The home directory is copied to the RAM disk device at boot time. So we can set up SSH private/public keys authentication. The only limitation is that files created from a node are not kept after CoMosix shut down. All must be set up in a particular directory before CoMosix boot.

LAM environment

LAM is working. But not as we really want. It doesn't use mosrun to start MPI software and mpiBlast get segmentation fault every time we run it. It also need a read-write access to GenBank database but that's not set.

GenBank fully accessible

GenBank is accessible only read-only but for mpiBlast we need a read-write access.

tmpfs

Nothing done here.

Networked kernel

Nothing done here.

Single control point

There's a web single control point up and running. But it need more functionalities.

GenBank update

We've got a automatic update, but it's not doing well. We've spend lot of time on it but it seems to be not enough.

SSH with private/public keys

This works quit well, but the problem comes from RSA fingerprint. They're generated at boot time and so if we connected once, the next time SSH will complain that the RSA fingerprint is wrong. So we configure SSH to not check RSA fingerprint but that node the best solution. Generating SSH RSA fingerprint at boot time is time consuming and ignoring verification is a security problem.

Install mpiBlast on the cluster

mpiBlast is installed but it doesn't work. Segmentation fault occurs every time. It might be necessary to compile it on a node and then install it into to root file system.

Fault list

- mpiBlast seg fault.
- GenBank doesn't update well.
- SSH RSA fingerprint are not kept.
- mpiBlast complain about read-only access on GenBank.

Others important things

- When we moved the root file system from a NFS server to another, the root file system decided to not work any more (because of the /dev directory we suppose).
- Problem with CoMosix kernel compilation. It cannot be compiled for some obscures reasons. The whole process of patching, configuring and compiling the kernel must be done once more. We might did something wrong (as patching in a wrong order ?)

Conclusion

There's still a lot of work to do, but CoMosix becomes more and more useful and one day it would provide good services to all the IRO staff. Once mpiBlast would be up and running, I think that the IRO informatics group will go on creating more and more software for the cluster !