General openMosix Daemon

Gian Paolo Ghilardi
Matthias Rechenburg
GOMD TEAM

What's GOMD?

- GOMD stands for "**general openMosix daemon**".
- GOMD is a daemon which executes commands and gets information from the nodes of an openMosix cluster. It has to run on every node in order to collect data, and it waits for commands to execute.
- The purpose of the project is to provide:
  - an integrated way to collect cluster statistics
  - a simple and well-defined format for statistics and data
  - easy-to-use APIs to receive raw or already-formatted statistics
  - new tools and/or mechanisms (i.e. SCX)
  - support for new interesting openMosix tools
  - and much more … ☺
- However our primary goal is to serve the openMosix community. ☺

Features (1/3)

- **Access Control List (ACL)**
  This feature allows you to specify:
  - allowed or blocked IPs/subnets
  - what they can do (permissions)

- **clusterSnapshot (CS)**
  This feature collects statistics/information of all available gomd-enabled openMosix nodes. When enabled, this facility, generates an HTML file with the summary of the cluster, node-by-node.

Features (2/3)

- **Remote Gomd Searcher (RGS)**
  Known also as Autodiscovery.
  The daemon is able to search for other daemons.
  Found daemons are cached in a special cache file.

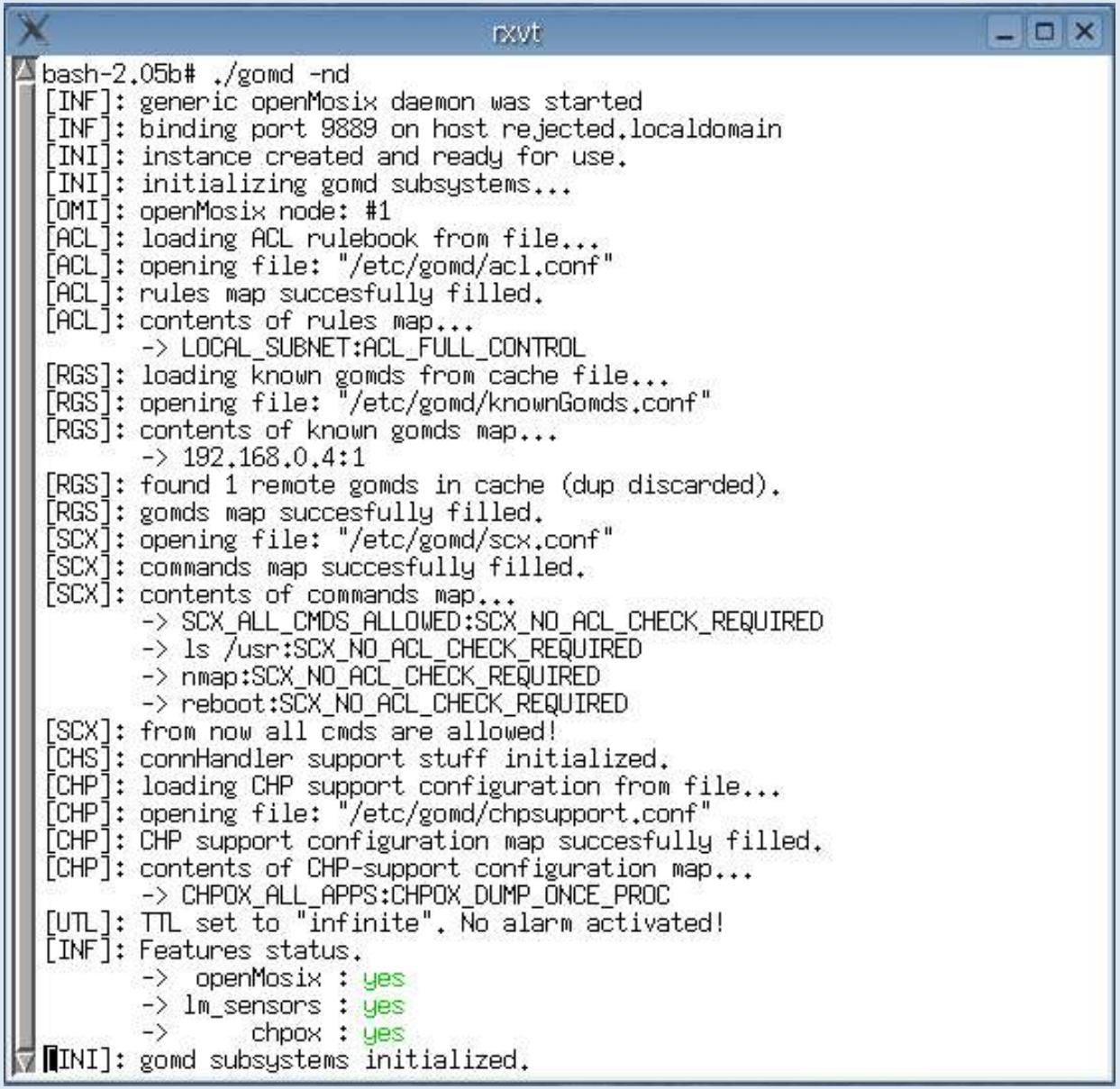
- **Secure Cluster-wide (command) eXecution (SCX)**
  This feature allows you to execute a command on the whole cluster. Obviously, you need a daemon running on each openMosix node. ☺
  You can specify exactly the allowed commands but you can also set a MACRO to allow or block all of the commands (this option is also known as "setting the default behavior"). If you prefer, you can also execute commands on the local node only.

Features (3/3)

Experimental stuff: support for external programs.

- **Chpox Support (CHP)**
  This feature allows you to register and dump a process via the nice Chpox program. (status: WIP)
- **LM_sensors Support (LMS)**
  This feature allows you to get the status of some hardware sensors including CPU temperature, fan speed, … (status: WIP)
- **LoadLimit Support (LoL)**
  This feature allows you to easily set and control the load limit value for each openMosix node. (status: planned)

How it appears...



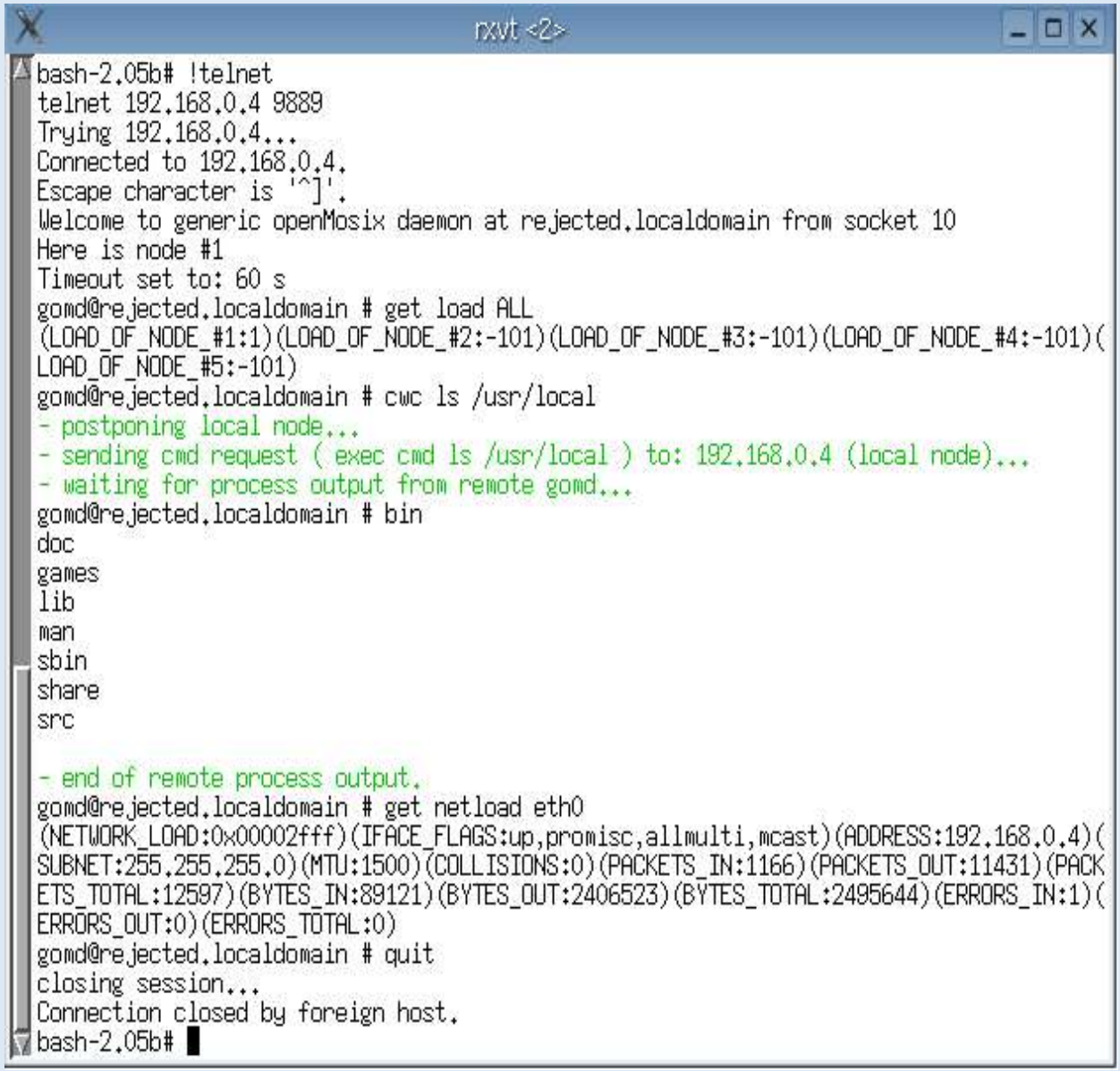
This is how the daemon appears at startup.

You can see the status of each loaded subsystem.

LEGEND:

INF	Debug Information
ACL	Access Control List
RGS	Remote Gomd Searcher
SCX	Secure Cluster-wide (command) eXecution
CHP	CHPOX Support

...and how to contact it.



You can interact with the daemon via a standard telnet session.

In the image on the left, you can see the execution of some standard commands, including a call to the SCX facility to propagate a command (a simple “ls /usr/local”) to the whole cluster.

The daemon can provide extra information via the excellent Libgtop library (optional).

What about daemon clients?

- Along with the daemon sources we provide some client examples including:
  - A “pure C” client
  - A C++ client
  - a QT-based client
  - a bash script client using expect
- In the future we'd like to integrate GOMD in the openMosixView project.
- Finally, the openMosixApplet now supports GOMD.

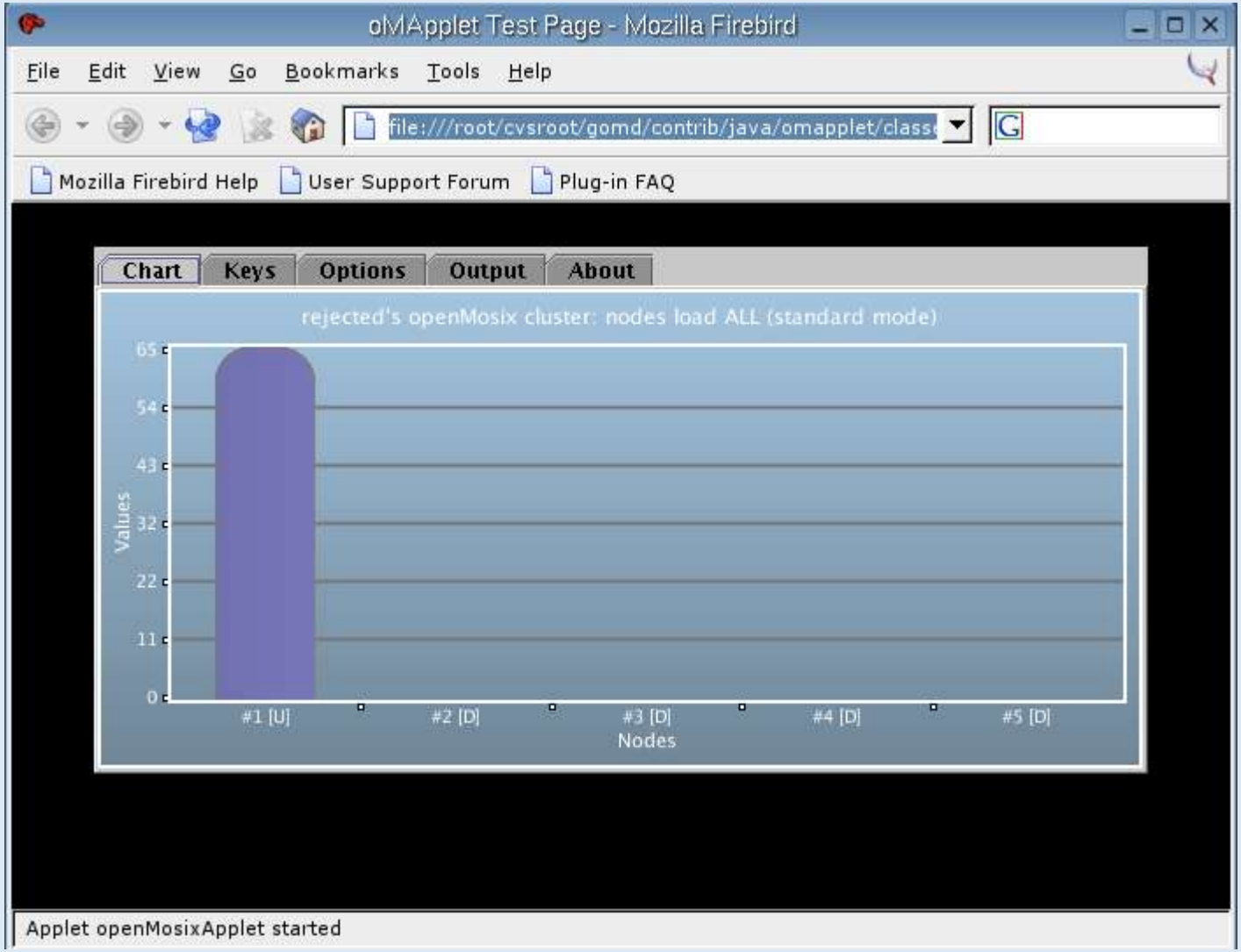
Subprojects (Contrib)

- The key idea for the gomd project is to assure bindings for all the languages.
- Actually we've 3 active sub-projects for this purpose:
  - java2gomd (maintainer: Roel Baardman)
  - perl2gomd (maintainer: Marco Marocco)
  - oracle2gomd (maintainer: Marco Marocco) ☺
- Moreover we've some scripts for interacting with GOMD via Bash. ☺

Libgomd

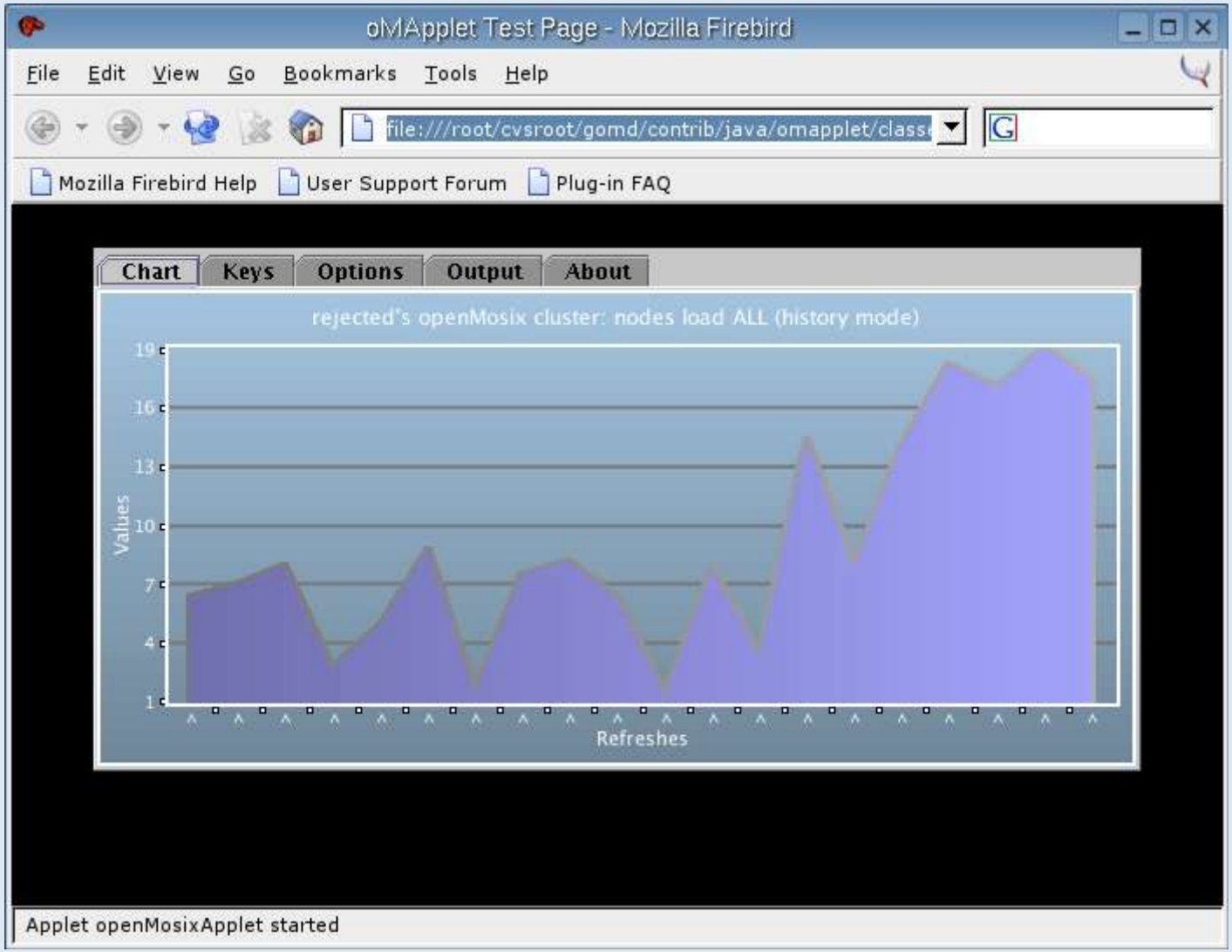
- With the daemon, the GOMD project provides a simple library named **libgomd** for fast creation of client programs.
- Particularly, this library provides some handy functions to contact a remote gomd daemon and receive the answers.
- The library can be used by both C and C++ programmers. Actually it exposes:
  - some quick functions for C users
  - a practical class for C++ programmers
- The daemon itself uses this library for daemon-to-daemon communication (for statistics/information exchange, …). ☺

OpenMosixApplet 1/2



This is the port for GOMD of the openMosixApplet.
This Java2 applet uses Jason J. Simas' oustanding Chart2D library to monitor the status of an openMosix cluster, in REAL TIME! This is the real-time ("standard") mode ...

OpenMosixApplet 2/2



... while this is the history mode (values are averages of all nodes).
The applet supports all basic openMosix infos (cpus, loads, memory status, ...).
It's provided in a single JAR file for easy deployment.

Join the project!

We are always in search of fresh forces.
Please consider to join the GOMD team.

For more info, visit:
http://www.nongnu.org/gomd

That's all folks!

Thank you! ☺