StoRM installation and configuration

Luca Magnoni and Riccardo Zappi INFN-CNAF

SRM2.2 deployment workshop - Edinburgh

13 November 2007

- StoRM requirements
- Getting StoRM with YAIM
- StoRM service configuration

StoRM guides

- Installation and configuration guides for StoRM can be found in the documentation section of the StoRM web site: **http://storm.forge.cnaf.infn.it**
- All information reported in this presentation can be found on the guides.

Pre-requisites

- **Scientific Linux CERN** edition or **Red Hat Enterprise Linux** version 3.X.
- **Host certificate**
- **Java 5**
- **ACL** on file system. For generic **Posix** fs, as ext2/3 or xfs, enable it in your /etc/fstab with the specific flag:

  file system mount point type options dump pass
  */dev/hda1 /storage ext3 defaults,***acl** *0 1*

Grid services

The StoRM environment:

- **LCAS/LCMAPS**, with VOMS support, configured on the machine[s] hosting the Backend components.
- **Local user** and **pool account** configured for the VO you want to support, both on frontend an backend machine[s].
- A **GridFTP** server, by default it is installed on the backend host[s], but a generic GridFTP server will be ok.
- **MySQL server** 4.1.11.

Through the **YAIM** installation of StoRM all this services will be installed and configured **automatically**.

YAIM

- Yet Another Installation Manager (YAIM)
- Install and configure software as defined by specific **profiles**.
- From YAIM 4 it becomes modular.
- All the service **automatically configured**.
- All the service has a **basic configuration**.
- Fine tuning and optimization have to be done by hand.

IG-YAIM

- IG-YAIM is the INFN Grid customization of YAIM.
- It relies on the official YAIM.
- StoRM is part of the official INFN Grid release.
- **IG-YAIM 4.0.0-3** contains all scripts and profiles to install **StoRM v1.3.18**.

APT repositories

To get StoRM with YAIM, first of all check in you apt sources directory for:

- The **Glite** repository.
- The **INFN Grid** repository (to found IG-YAIM and StoRM related packages).
- The **LCG-CA** list for the certificates of Certification Authorities.

Then, you can download and install **YAIM** and **IG-YAIM**.

site-info.def

- The **site-info.def** file is the input for YAIM that should contains all the site specific services variables.
- From **YAIM 4**, there should be the possibility to split the global file in a set of service-specific *.def*, but currently it does not work on IG-YAIM :).
- An examples of files to use with StoRM can be found on */opt/glite/yaim/examples/siteinfo/ig-site-info.def*.

StoRM variables

In the *site-info.def* sapmle there are all the variables needed to configure StoRM:

- **Generic variables** shared with other services:
  - JAVA_LOCATION
  - MYSQL_PASSWORD
  - xxx_REPOSITORY (please pay attention here! Wrong OS_REPOSITORY can generate a lot of caos.)
- **Specific StoRM variables**
  - Service properties configuration
  - Storage configuration

StoRM specific variable: service

This variables used to set the service properties:

- *STORM_HOST=hostname*
- *STORM_PORT=8444*
- *STORM_USER=storm*
- *STORM_DB_USER=storm*
- *STORM_DB_PWD=storm_pwd*
- *STORM_DB_HOST=localhost*

StoRM specific variable: storage

This variables are used to configure StoRM on the desired storage:

- *STORM_AP_LIST="voname1 voname2 voname3 ..."* This is the list of VO supported by StoRM. Subset of (or equal to) VOS.
- *STORM_DEFAULT_ROOT=/gpfs* Default Storage directory.
- *STORM_AUTH_POLICY=[permit—deny—ecar]* The low level authorization mechanism for file access. The usual approach is to have permit here and use the access control at file system level, through the **approchable rules**.

Storage variables per each VO supported

- *STORM_voname1_ROOT=/gpfs/infngrid* The path for the VO storage.
- *STORM_voname1_ACCESSPOINT=/infngrid* How the path above is exposed by SRM. This is used in SURL.
- *STORM_voname1_FSTYPE=[gpfs—xfs—ext3]* File system types.
- *STORM_voname1_FSDRIVER=[gpfs—xfs—posixfs]* ext2/3 and Lustre needs the *posixfs* driver.
- *STORM_voname1_SPCDRIVER=[gpfs—xfs—mock]* Use mock for file system different from GPFS and XFS.
- *STORM_voname1_TOKEN=LHCb_RAW* VO Storage Area Token.
- *STORM_voname1_ACLMODE=[aot—jit]* ACL enforcing mechanism.

Install and configure

Once you have properly configured the **site-info.def** file, you can install and configure StoRM and all the related packages (lcas-lcmaps, gridftp. etc).

- **Install services**: */opt/glite/yaim/bin/ig_yaim -i -s site-info.def -m ig_SE_storm_mysql*
- **Configure services**: */opt/glite/yaim/bin/ig_yaim -c -s site-info.def -n ig_SE_storm_mysql*

Common errors

- **Repositories misconfiguration**. YAIM repositories different from the ones already configured on the host.
- **Java** not correctly installed. (Look at StoRM FAQ).
- **ACL not enabled** on the file systems.
- **Permission on StoRM** directory. Be sure the *storm* user will have all the correct permission on files and directories.

Frontend configuration

The Frontend installation directory is */opt/srmv2storm/*.
The Frontend configuration is based on:

- file */opt/srmv2storm/etc/sysconfig/srmv2storm.nsconfig*
- that contains all information for database connection:
  db_user/db_password
  @frontend_host

Backend configuration

The Backend installation directory is */opt/storm/*.
The Backend configuration is based on two files in */opt/storm/etc*:

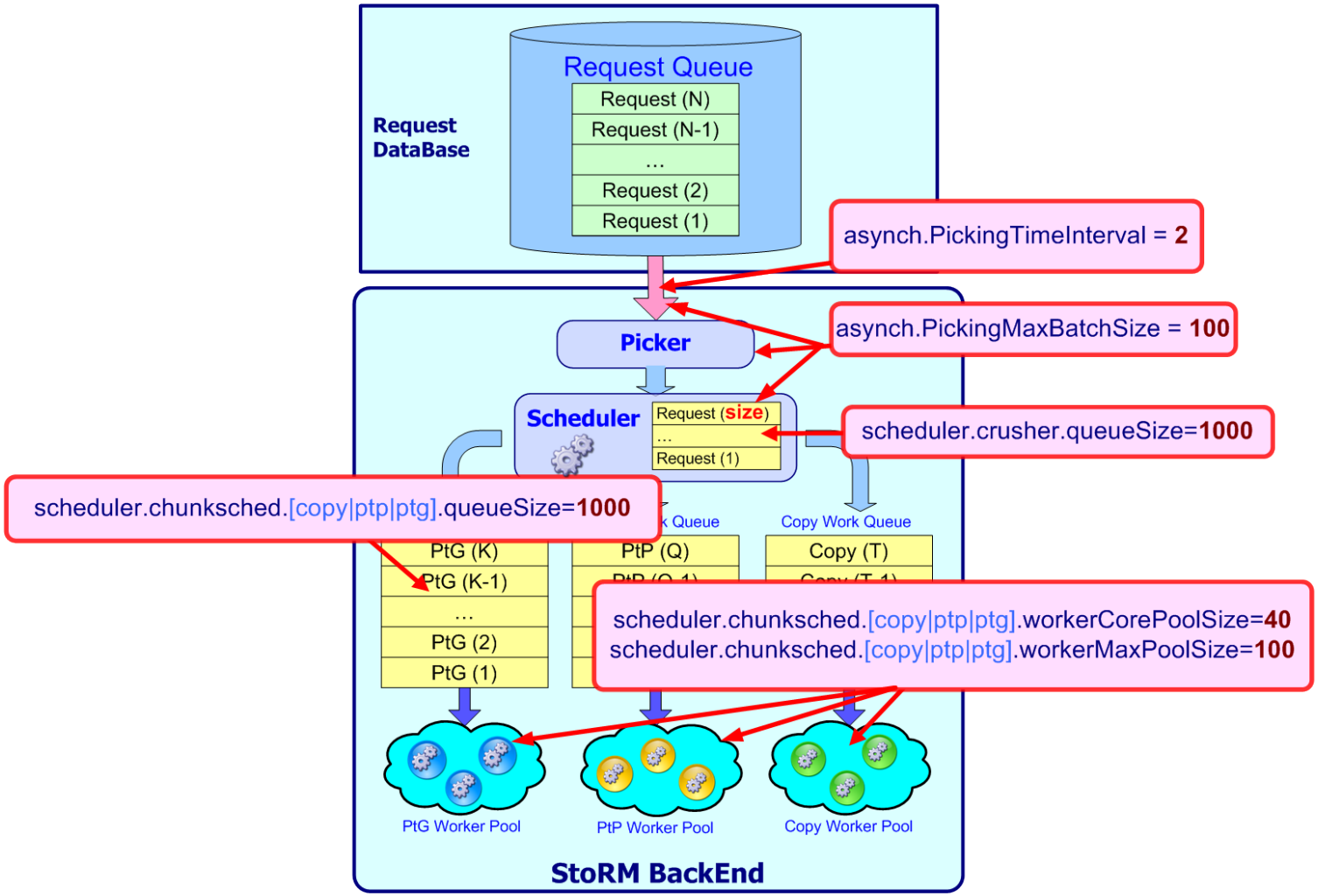
- **storm.properties**
- **namespace.xml**

storm.properties

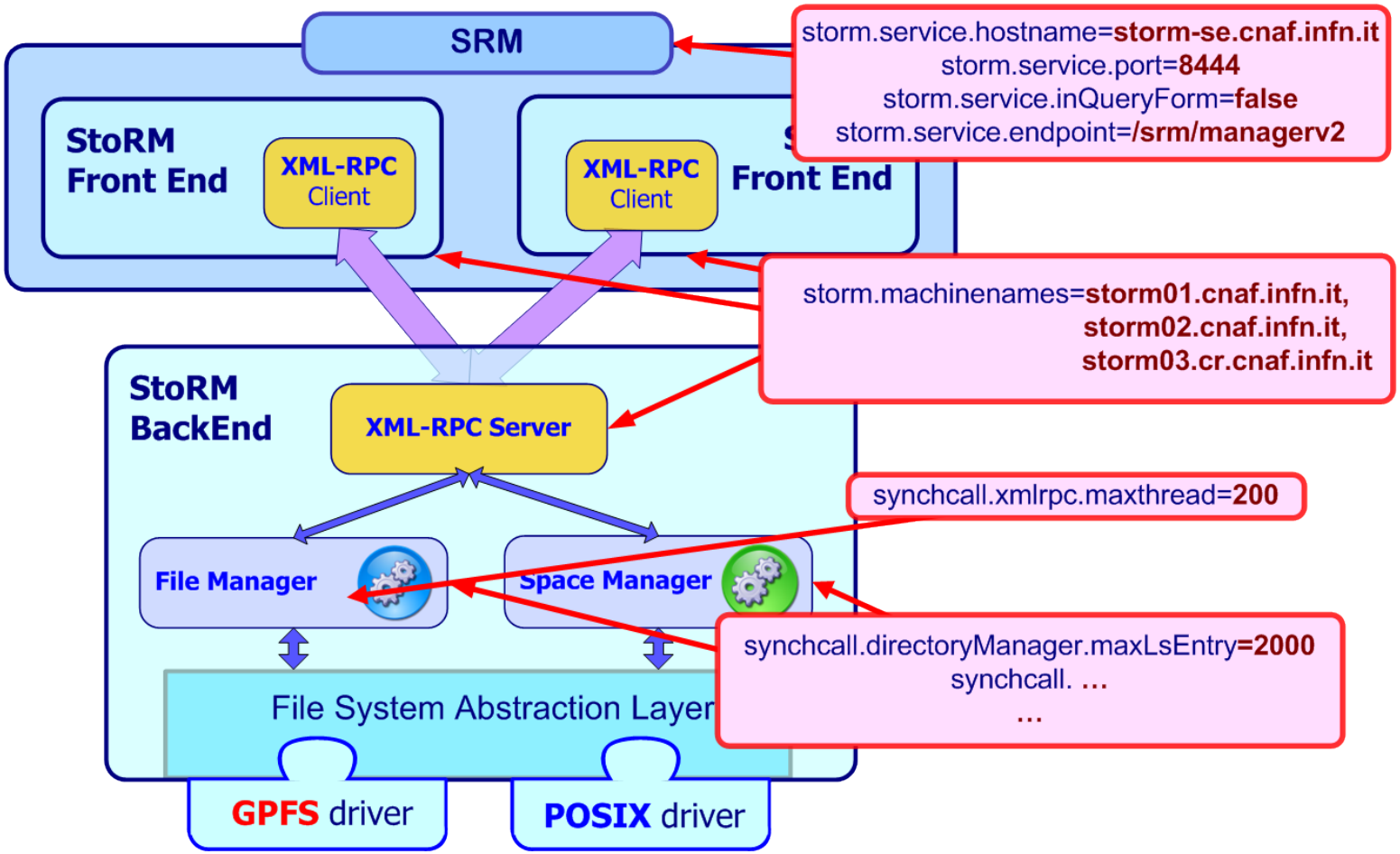
The **storm.properties** is a standard JAVA properties file, a set of **key=value** entries. An example:

- *storm.service.hostname=storm01.cr.cnaf.infn.it*
- *storm.service.port=8444*
- *storm.service.inQueryForm=true*
- *storm.service.endpoint=/srm/managerv2*
- *persistence.db.host = localhost*
- *persistence.db.username = storm*
- *persistence.db.passwd = storm*
- *persistence.db.pool = true*
- *persistence.db.pool.maxActive = 10*
- *persistence.db.pool.maxWait = 50*
- *...*

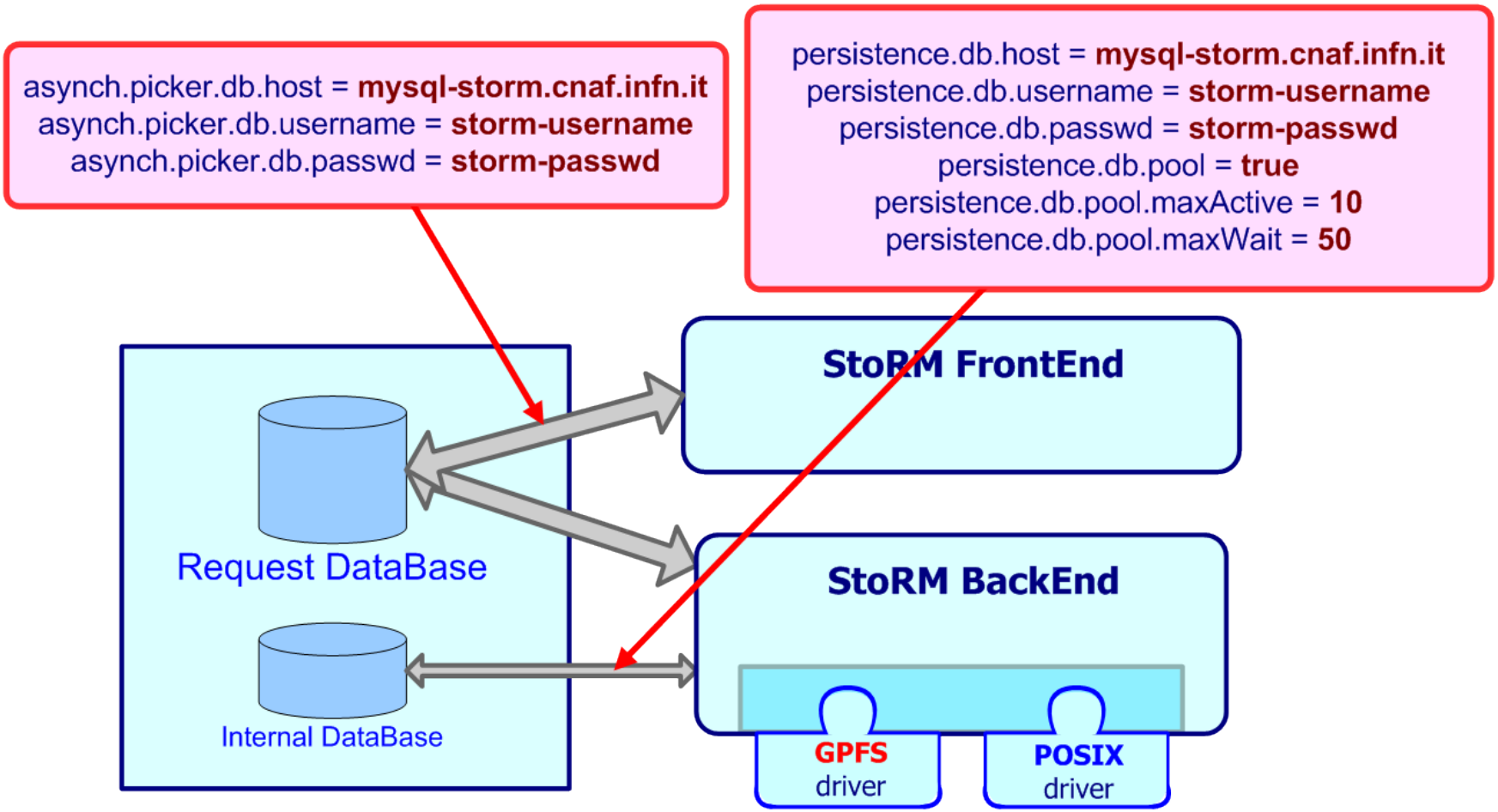
storm.properties



storm.properties



storm.properties



Log files

StoRM log files can be found:

- **Frontend log**:
  - **/opt/srmv2storm/var/log/srmv2storm.log**.
- **Backend log**:
  - **/opt/storm/var/log/storm-backend.[log—stdout—stderr]**.

log4j

The Backend uses *log4j* to structure the log operations.

- Different level of logging: **DEBUG, INFO, WARN, ERROR, FATAL**.
- Configured by the file *log4j.properties*.
- Depending on the log level specified, **the StoRM Backend will be more or less verbose**.
- The DEBUG mode is really verbose, but can help a lot in case of error.

log4j

In the file *storm-backend.log* you can find all log information, in *storm-backend.stderr* only the system erro. The log are structured:

- Log example: **2007-11-09 19:21:04,565 INFO [pool-1-thread-5] Log message...**
-
  - Date
  - Log level
  - Thread name
  - Log message