



Enabling Grids for E-science

Storage Accounting for Grid Environments

Fabio Scibilia
INFN - Catania
08.03.2007

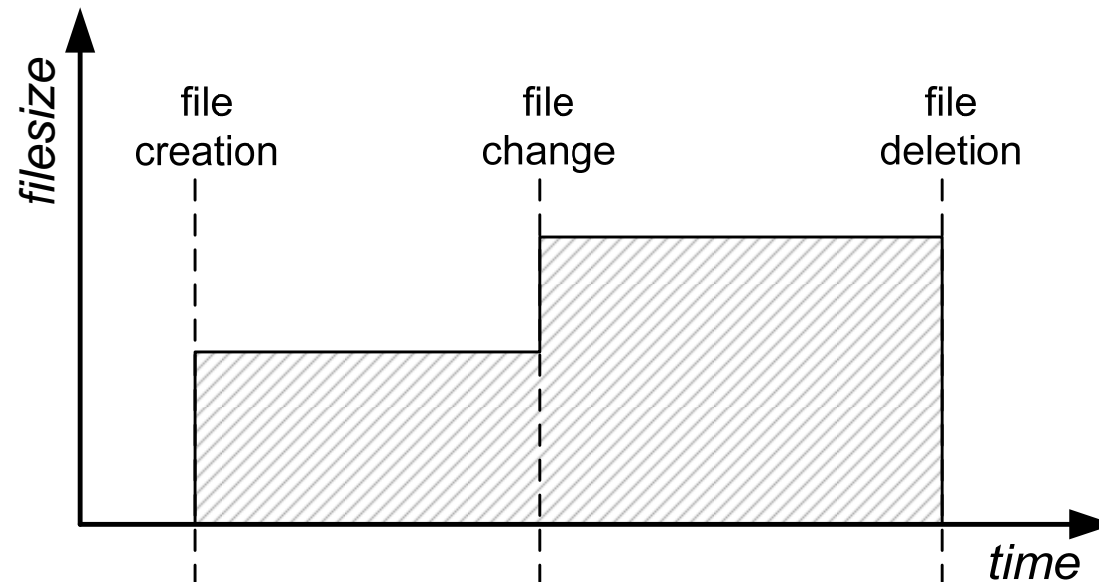
www.eu-egee.org
www.glite.org



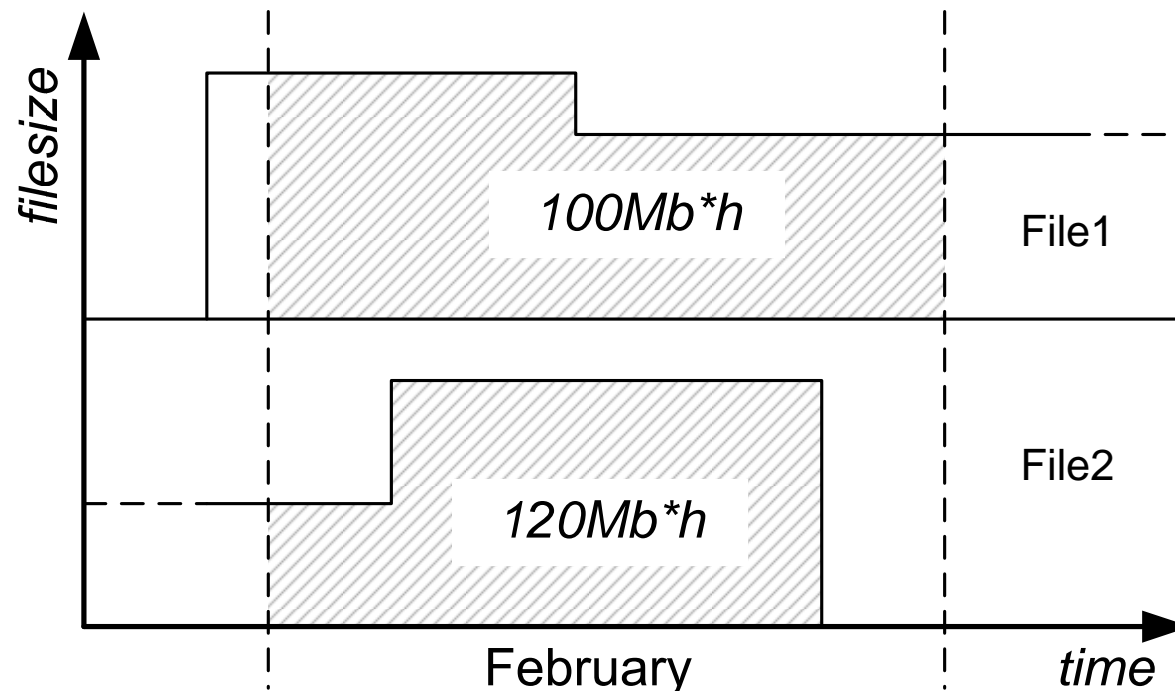
- **Storage Accounting for Grid Environments (SAGE)**
- **System to collect usage metering information on Storage Elements**
- **C++, mysqlclient, API of DPM, openssl**
- **Will be integrated in DGAS at the Usage Metering level**
- **Works over DPM-based SE. However**
 - Most of the software can be reused for other systems
 - DPM is not aware of being accounted
- **Provides for local usage information**
- **Defines novel reports to the users**

- **User activities accounting information**
 - Actions taken by a user against one of his/her files
 - Putting, modifying, retrieving and deletion of a file are user activities
 - Each activity consists of an action, a file, the number of bytes affected, the time it started/stopped, the user credential and so on.
 - Will be integrated in DGAS HLR
- **Disk Usage information**
 - Is accounted in terms of space and time
 - Is accounted user by user and VO by VO
 - Is evaluated considering user activities
 - We defined the *disk energy* function to create reports

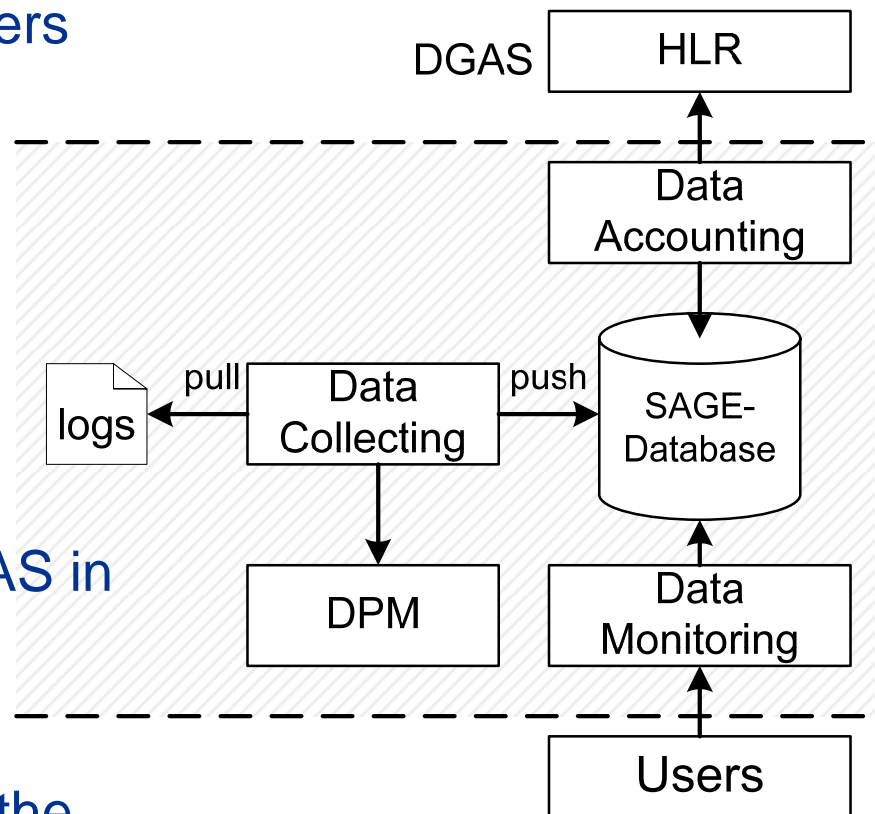
- Defined as “*The integral of the size of the file along the time*”
- In the figure is the slashed area
- Can easily evaluated at any time just knowing all events that affected that file
- Expressed in *Mbytes*hours*



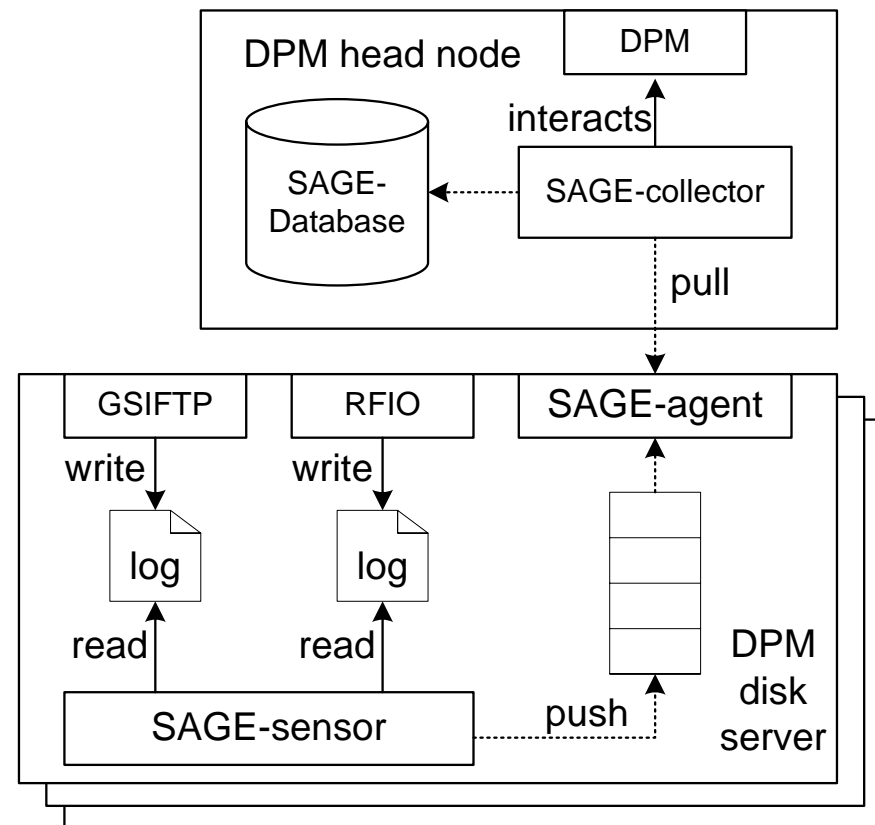
- Are related to a user or to a VO or to a couple (user/VO)
- Refer to specific period in time
- In the example, the user consumed (220Mb*h) of disk energy with his 2 files.



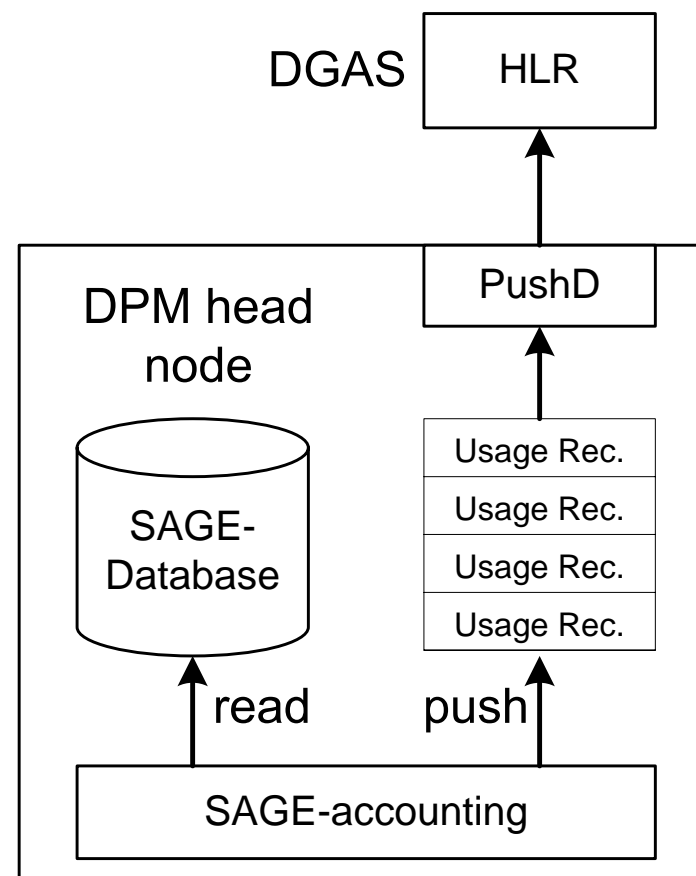
- **Data Collecting**
 - To collect data from disk servers related to user activities
- **SAGE-Database**
 - To store collected data and reports on the usage of the resource
- **Data Accounting**
 - To integrated SAGE with DGAS in the future
- **Data Monitoring**
 - To provide for an interface to the users and a system for reporting



- **SAGE-sensor**
 - Reads info from logs of GSIFTP and RFIO
 - Creates and queues this info
 - Can be easily extended to other protocols
- **SAGE-agent**
 - Make this info available to the collector
- **SAGE-collector**
 - Periodically polls all the agents of the pool and pulls new info
 - Interact with DPM to complete all missing information

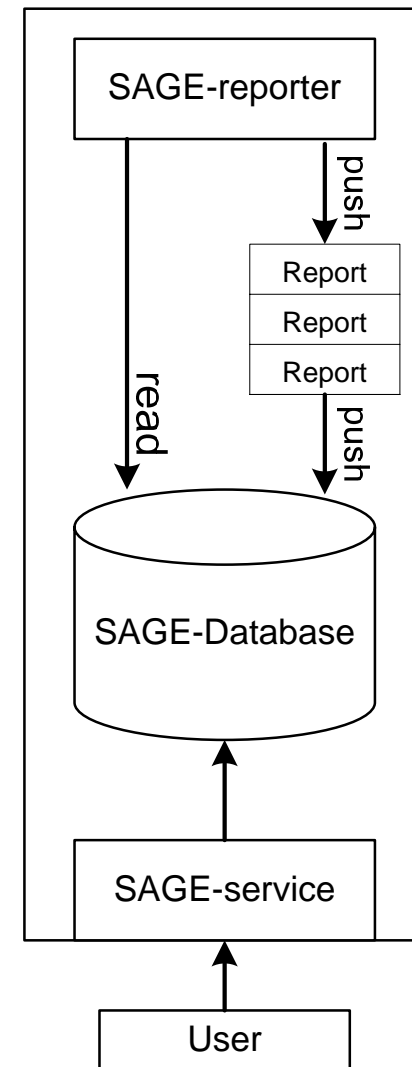


- **SAGE-Accounting**
 - Reads data from SAGE-database
 - Creates and queues Usage Records
- **PushD**
 - Pushes Usage Records to the HLR of DGAS
 - Wakes up periodically
- **Usage Record**
 - Not yet defined for storage accounting
 - Under discussion!!!



- **SAGE-Reporter**
 - Wakes up periodically
 - Reads status of all current files
 - Creates reports
 - Pushes back these reports to the Database

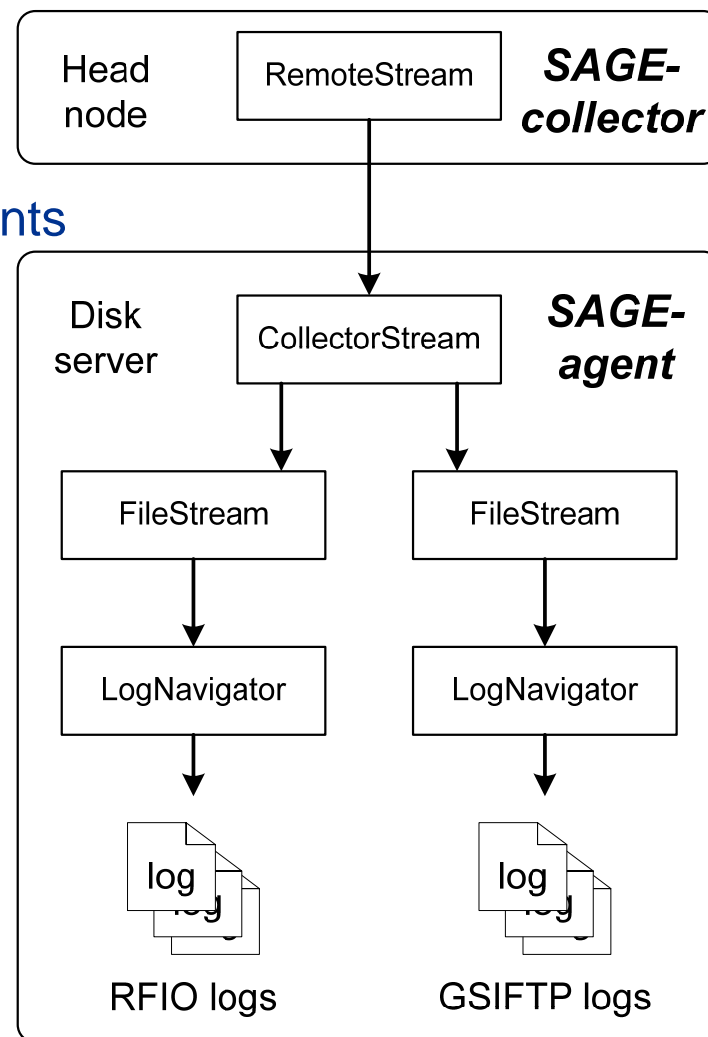
- **SAGE-Service**
 - Let users access their reports
 - Make some other control stuff
 - Is accessible to users
 - Details under definition!!!



- **Interface `sage::sensor::Stream`**
 - Interface with methods to open, read, move and close a log stream.
 - `FileStream`: Gets log information from log files (e.g. `/var/log/rfio.log`)
 - *Requires a parser for specifically for the file (RFIO or GSIFTP)*
 - *Is able to manage log rotation*
 - `CollectorStream`: Manages more streams as in a collection
 - *Sorts info into the stream chronologically*
 - `RemoteStream`: To access to a stream remotely
 - *The SAGE-collector and the the SAGE-agent use this stream to communicate*
 - Can be combined in more ways
- **Interface `sage::sensor::Parser`**
 - To parse log files
 - GSIFTP
 - RFIO
 - DPNS

- **Is a library coded by us:**
 - To access information of log files as it was a stream of data
- **Is includes following interfaces**
 - `sage::sensor::Parser`
 - Parser for log files
 - Three implementations: GSIFTP, RFIO and DPNS
 - `sage::sensor::LogNavigator`
 - Allow to move within more log files as they were a unique file (e.g: `/var/log/rfio.X` where `X=0 . . .`)
 - `sage::sensor::Stream`
 - Treats log information as in a stream
 - Three implementations: `FileStream`, `CollectorStream`, `RemoteStream`
 - Implementations can be combined

- **FileStream: Stream**
 - Reads log info from log files
 - Two instances: GSIFTP and RFIO
 - Captures and manages log rotation events
 - Uses a Parser to parse log lines
- **CollectorStream: Stream**
 - Extracts log data from more streams chronologically
 - One instance used by the Agent
- **RemoteStream: Stream**
 - Access through on open channel to a remote stream
 - The agent works to open the a SSL channel with mutual authentication
 - One instance for each disk server



- **Data Collecting is about to be ready**
 - The SAGE-sensor and SAGE-agent are ready
 - The SAGE-collector is about to be ready
 - In next week we will deploy it on our GILDA testbed
- **SAGE-Database**
 - Data model is ready
 - Database deployed on my laptop 😊 !
- **Data Monitoring**
 - We are about to start working while we test Data Collecting
 - Some stuff is under definition
 - use GT4???
 - Data report model
 - Report policies etc.

