Distributed storage works update

SuperB Collaboration Meeting
Pisa, Sept. 20th 2012

Paolo Franchini (CNAF)
for the distributed storage group

Summary

- Overview on WAN environment
- HTTP data access test
- Data access library development

Wide Area Network environment

- Distributed resources
  - Resources comparable with ATLAS and CMS experiments for the 2011 runs
  - Several computing sites in Europe and North America, under a Grid infrastructure
  - LHC Computing Grid architecture adopted
  - LHC Tier classification: few Tier1s and several Tier2s
- Distributed storage R&D
  - Exploit WAN data access during the data model definition
  - Dynamic and remote data access vs data driven computation paradigm

WAN data access

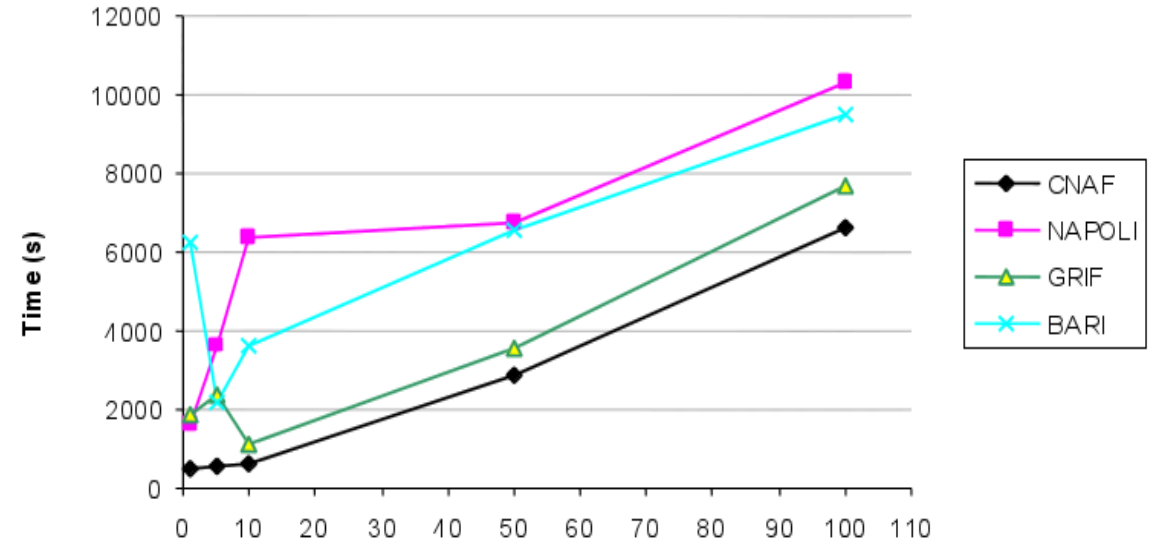
- Experiment use cases:
  - interactive usage of SuperB data
  - analysis code writing and debugging
  - analysis tasks executed on non SuperB resources
  - job execution on small sites like Tier3s
  - safeness in case of storage failure
- Network protocols
  - xrootd and HTTP:
    - support posix-like calls
    - capabilities of work through routers and firewalls
    - caching and pre-fetching mechanism
    - supported by ROOT framework

HTTP data access test

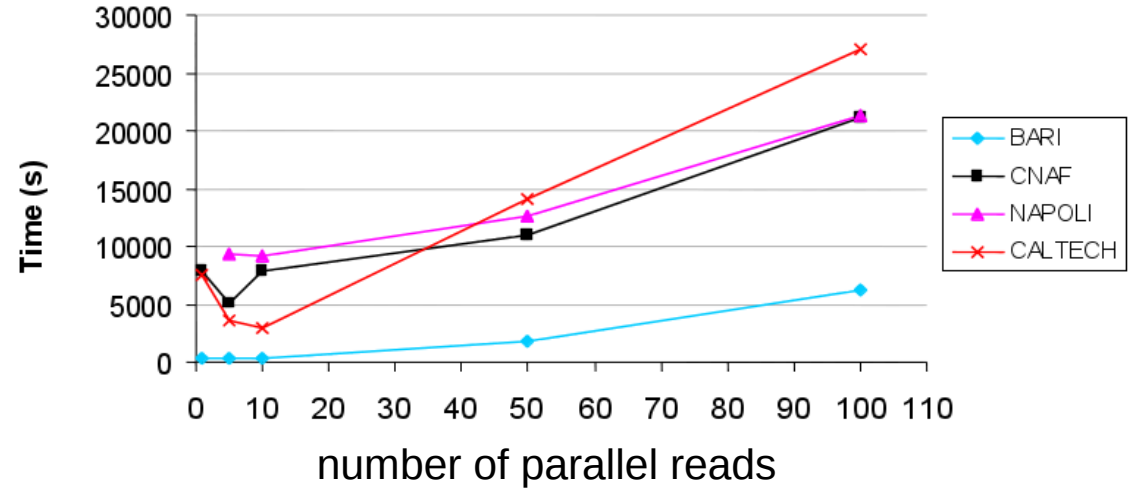
- Test goal:
  - measure the latency period due to the increase number of parallel read stream
  - measure the latency period due to the increase of round trip time elapsed between source and destination
- Test layout definition:
  - 1, 5, 10, 50 and 100 parallel set of read streams
  - each stream reads a random files according to a trace file obtained from an analysis application
  - 250 compressed root files, 500 MB each
  - sources: INFN-T1 and INFN-Bari
  - destinations: INFN-T1, INFN-Bari, INFN-Napoli, GRIF, Caltech
  - measured the time of the cURL execution

HTTP data access test: results

Data source: INFN-T1



Data source: INFN-BARI



HTTP data access test: results

- The network latency influences the read stream operations for all the routes
- The link congestions affect the case of single read-stream also on short routes
- The dips of the curves can be the effects of a specific link-to-link overload

Data access library

- R&D for a library permitting to shape the data access
- Features:
  - intelligent pre-fetching and buffering algorithms
  - logical file name map with different physical storage URI
  - possibility of support to unsupported ROOT storage protocols
  - read-head buffer and caching mechanism in order to solve the overhead

Library approaches

- **High level**: library wrapper of ROOT data access/download methods
  - pros: simple deployment inside the experiment framework
  - cons: users need to uses new access methods
- **Low level**: new file protocol developing a ROOT class
  - pros: no impact for the final user
  - cons: need an ad-hoc ROOT implementation
- **Configuration driven**: need a ROOT configuration interface in order to change the data access according to a set of parameters.

Library state-of-art: `libSbNet`

- The library input is the catalog name (`lfn://`) of the file that must be used in the analysis
- The library output is the local file name (`file://`) that ROOT can use in the analysis
- In order to obtain the output the library first checks the default storage element, defined in the environment variable `VO_SUPERBVO_ORG_DEFAULT_SE`.
- If this SE returns a valid `file://` TURL the work is done…

Library state-of-art: `libSbNet`

- If the default SE has some problem the library uses the lcg API (`lcg_lr`) to obtain the list of the file replicas, sorts them in network speed order, then asks every SRM a valid `file://` TURL
- If the `file://` protocol is not available, then falls down to slower protocols, first `http://` (not yet implemented) and eventually `gsiftp://`
- In the `gsiftp://` case the lib copies the file locally and returns his full local path at the caller.
- It’s up to the lib caller to delete this temporary file at the end of its use.