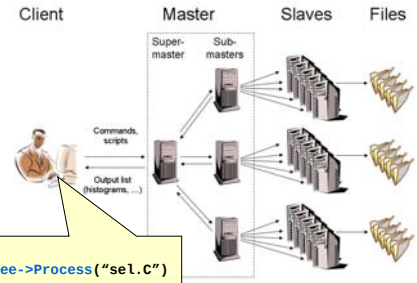
PROOF – Parallel ROOT Facility

The PROOF system allows:
- parallel analysis of trees in a set of files
- parallel analysis of objects in a set of files
- parallel execution of scripts

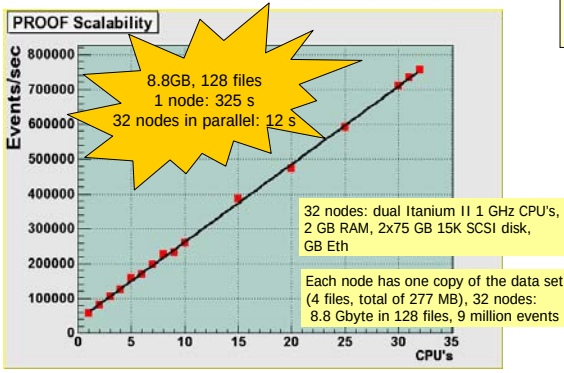
on clusters of heterogeneous machines

Its main design goals are:
- Transparency:
  - input objects copied from client, output objects merged, returned to client
- Scalability and Adaptability:
  - Varying packet size (depends on number of workers and their relative performance)

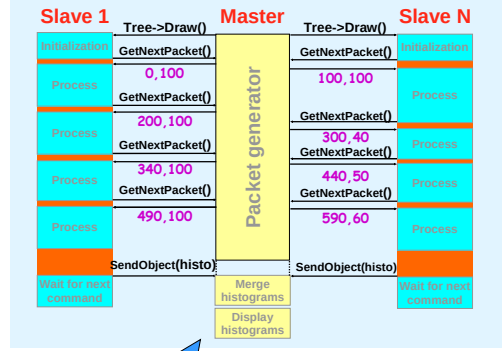
Multi-tier architecture




```
$ root
root[0] tree->Process("sel.C")
root[1] TProof proof("remote")
root[2] chain->Process("sel.C")
```



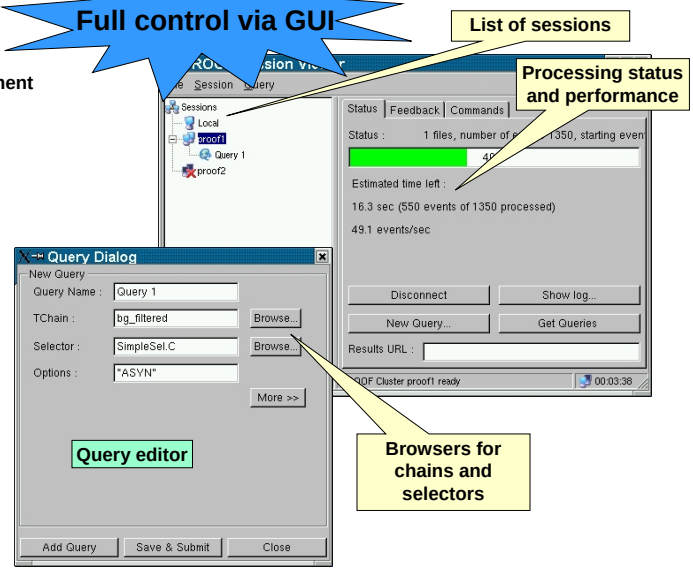
PULL Architecture — Master-Worker Workflow — Dynamic Load Balancing

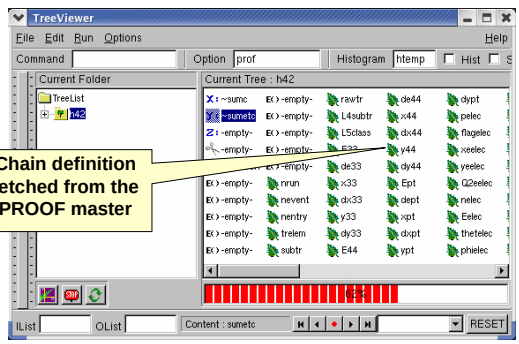


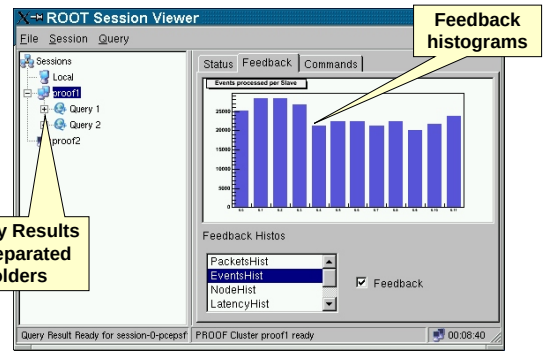
PROOF Data Access Strategies
- Assign to each worker node data in local files
- If no (more) local data, get remote data via daemon data servers (needs good LAN)
- In case of SAN/NAS just use round robin strategy

Features
- User runs in unique sandbox
  - Flexible authentication framework (pass-based, Krb5, GSI)
- Package manager for easy installation of user's environment
  - Additional shared libraries, data files, etc.
- Support for multiple sessions
  - Queries can be run concurrently on different sessions
- Support for asynchronous, non-blocking, running mode
  - Set of queries submitted to the cluster and processed sequentially in the *background*
- Query manager for easy handling of results
- Retrieve and Archive functionalities
  - results can be saved by the master on any mass storage accessible via `TFile::Open()`
- Full tree viewer functionality, including `TChain::Draw()`

Full control via GUI







Grid Interface — AliEn Supported
- Access via an abstract interface class
  - use Resource Broker to locate available nodes
  - use File Catalogue and Storage Index to map LNF's to a chain of PFN's
  - use Monitoring Services
- AliEn concrete implementation

PROOF is a collaboration between the CERN core ROOT team and the MIT Heavy Ion Group