PROOF - Parallel ROOT Facility

Maarten Ballintijn

http://root.cern.ch

Bring the KB to the PB not the PB to the KB

Agenda

- Architecture
  - TSelector
  - TDSet
  - Environment
- Implementation
  - TProofPlayer
  - TPacketizer
  - Dynamic Binning Histograms
  - Merge API

TSelector – The algorithms

- Basic ROOT TSelector + small changes

```
// Abbreviated version
class TSelector : public TObject {
Protected:
    TList *fInput;
    TList *fOutput;
public
    void Init( TTree* );
    void Begin( Ttree* );
    Bool_t Process(int entry);
    void Terminate();
};
```

TDSet – The data

- Specify a collection of TTrees or files with objects

```
[] TDSet *d = new TDSet("TTree", "tracks", "/");
[] TDSet *d = new TDSet("TEvent", "", "/objs");
[] d->Add("root://rcrs4001/a.root", "tracks", "dir",
first, num);
…
[] d->Print("a");
[] d->Process("mySelector.C", nentries, first);
```

- Returned by DB or File Catalog query etc.
- Use logical filenames ("lfn:...")

Sandbox – The Environment

- Each slave runs in its own sandbox
  - Identical, but independent
- Multiple file spaces in a PROOF setup.
  - Shared via NFS, AFS, multi CPU node
- File transfers are minimized
  - Cache
  - Packages

Sandbox – The Cache

- Minimize the number of File transfers
  - One Cache per file space
- Locking to guarantee consistency
- File identity and integrity ensured using
  - MD5 digest
  - Time stamps
- Transparent via TProof::Sendfile()

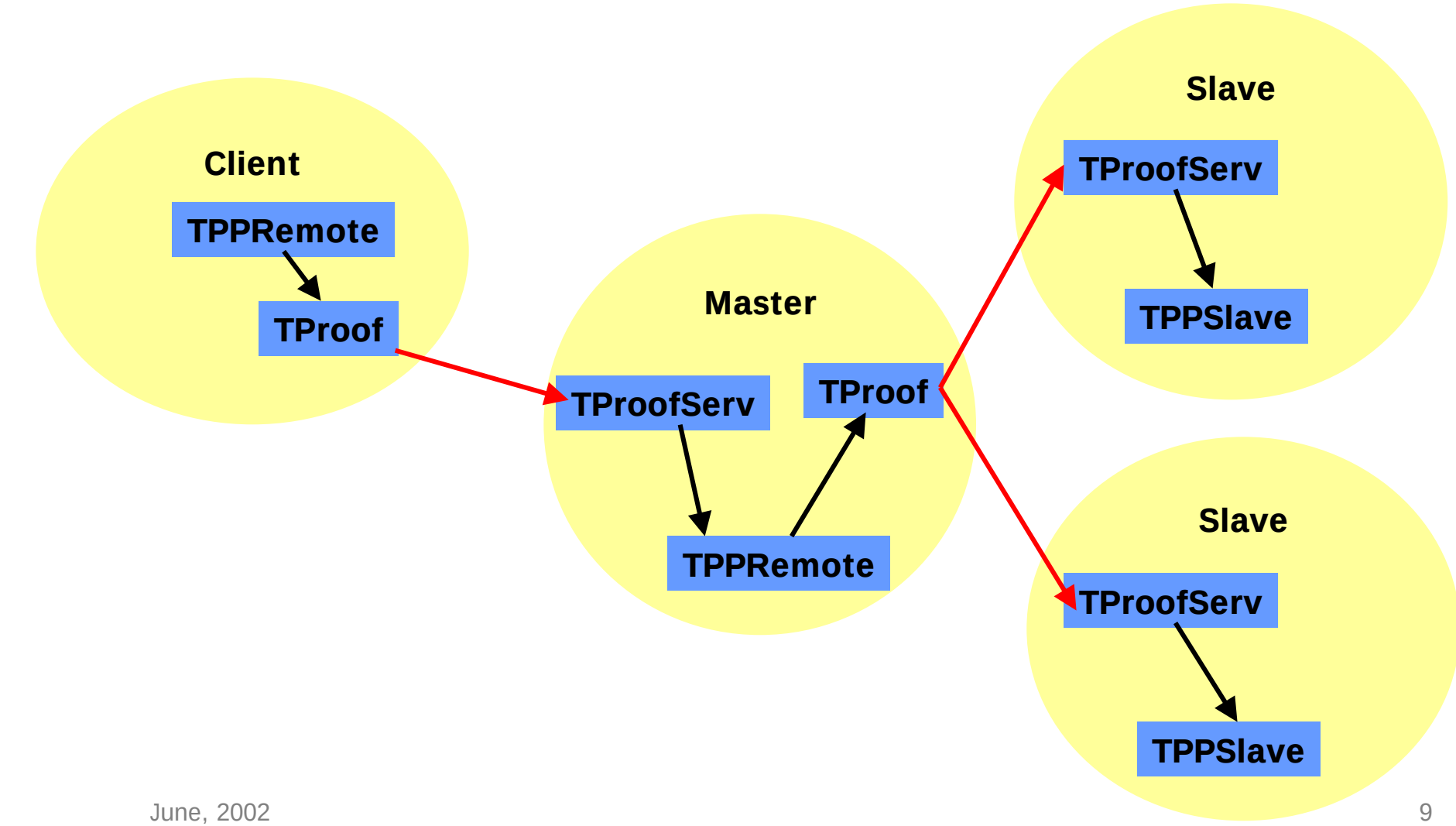
Sandbox – Package Manager

- Provide a collection of files in the sandbox
- Binary or Source packages
- PAR files: Proof ARchive. Like Java jar
  - Tar file, ROOT-INF directory
  - BUILD.C or BUILD.sh
  - SETUP.C, per slave setting
- API manage and activate packages

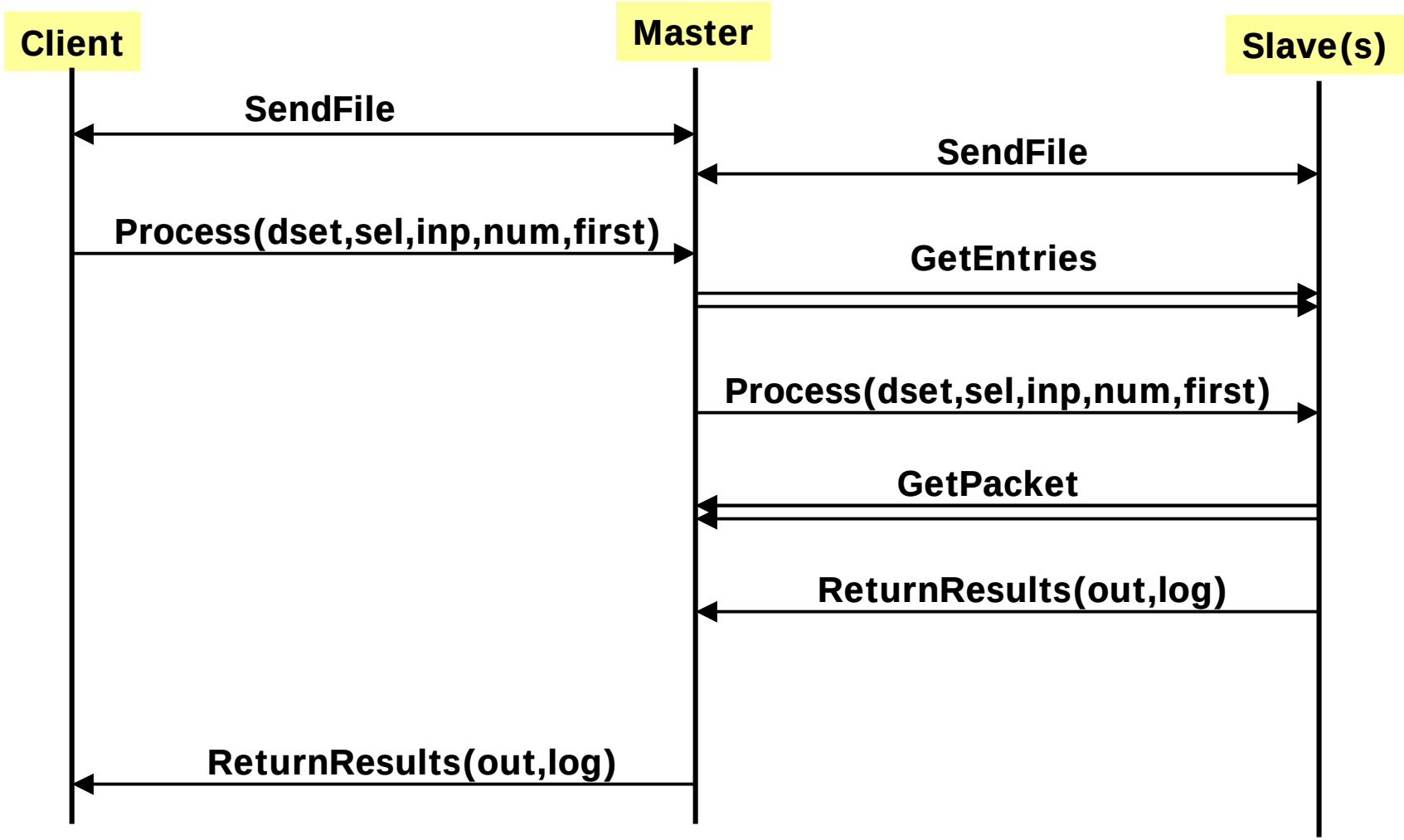
Implementation Highlights

- TProofPlayer Class hierarchy
  - Basic API to process events in Proof
  - Implement Event Loop
  - Implement proxy for remote execution
- TEventIter
  - Access to TTree or TObject derived collection
  - Cache File, Directory, Tree

TProofPlayer



Simplified Message Flow



Dynamic Histogram Binning

- Implemented by extending THLimitsFinder
- Avoid synchronization between Slaves
- Keep score-board in master
  - Use histogram name as key
  - First slave posts limits
  - Others use these values

Merge API

- Collect output lists in master server
- Objects are identified by name
- Combine partial results
- Member function: Merge(TCollection *)
  - Executed via CINT, no inheritance required
- Standard implementation for Histograms
- Otherwise return the individual objects

Near Future

- Few more weeks testing in Phobos
- Beta test with a few other experiments
- Basic documentation
  - Install and Configure guide
  - User HowTo
- First Release in the next major ROOT Release

Future

- Ongoing development
- Event lists
- Friend Tree
- Scalability to O(100) nodes
- Multi site PROOF sessions
- The GRID

Demo!

- The H1 example analysis code
  - Use output list for Histograms
  - Move fitting to client
- 15 fold H1 example dataset at CERN
  - 4.1 Gbyte
  - 4.3 Million Events
- 4 fold H1 example dataset at MIT

Demo!

- Client machine
  - PIII @ 1GHz / 512 MB
  - Standard IDE disk
- Cluster with 15 nodes at CERN
  - Dual PIII @ 800 MHz / 384 MB
  - Standard IDE disk
- Cluster with 4 nodes at MIT
  - Dual AthlonMP @ 1.4GHz / 1GB
  - Standard IDE disk