



PROOF and ROOT Grid Features

Fons Rademakers



PROOF - Parallel ROOT Facility

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



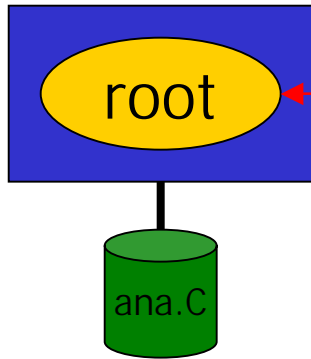
Parallel ROOT Facility

- The PROOF system allows:
 - parallel execution of scripts
 - parallel analysis of chains of treeson clusters of heterogeneous machines
- Its design goals are:
 - transparency, scalability, adaptivity
- Prototype developed in 1997 as proof of concept (only for simple queries resulting in 1D histograms)



Parallel Script Execution

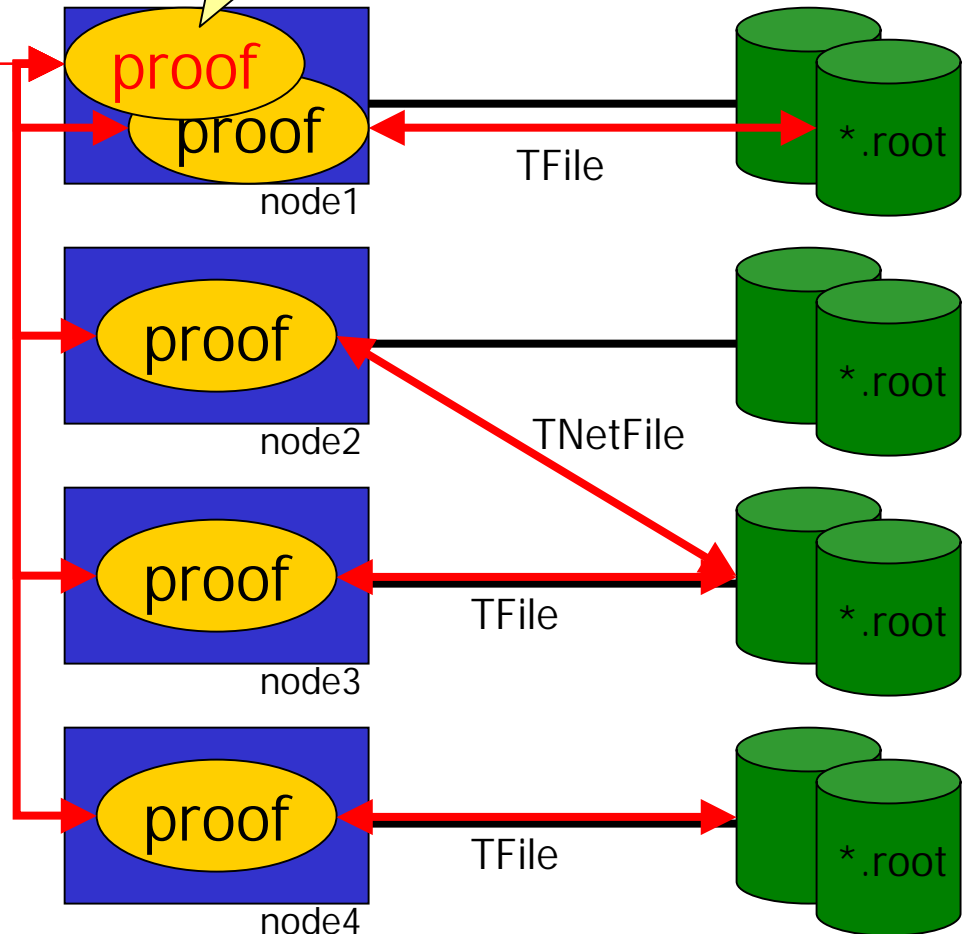
Local PC



← stdout/dbj
ana.C →

Remote Proof Cluster

```
#proof.conf
slave node1
slave node2
slave node3
slave node4
```



```
$ root
root [0] .x ana.C
root [1] gROOT->Proof("remote")
root [2] gProof->Exec(".x ana.C")
```

proof = master server
proof = slave server

PROOF Aware ROOT Script



```
void ana {
    if (gROOT->IsProofServ()) {
        if (gProofServ->IsMaster()) {
            printf("Macro running on master server\n");
            // single remote init
        } else {
            printf("Macro running on %d of %d\n", gProofServ->GetGroupId(),
                gProofServ->GetGroupSize());
            // parallel remote init
        }
    } else {
        printf("Macro running in local ROOT session\n");
        // local init
    }
    ...
}
```



Parallel Tree Analysis

```
root [0] .! ls -l run846_tree.root
-rw-r-r--  1  rdm      cr   598223259  Feb 1  16:20  run846_tree.root

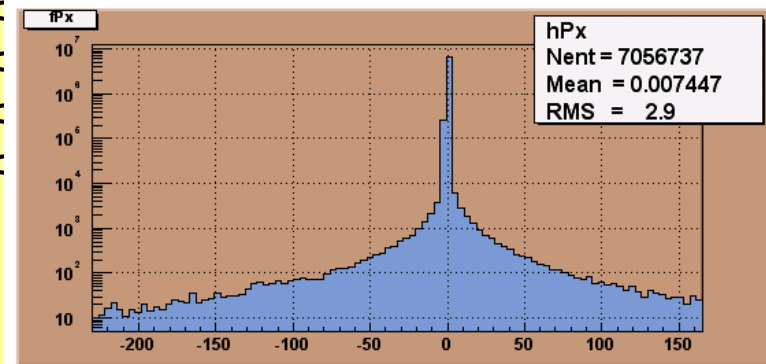
root [1] TFile f("run846_tree.root")

root [2] gROOT->Time()

root [3] T49->Draw("fPx")
Real time 0:0:11, CP time 10.860

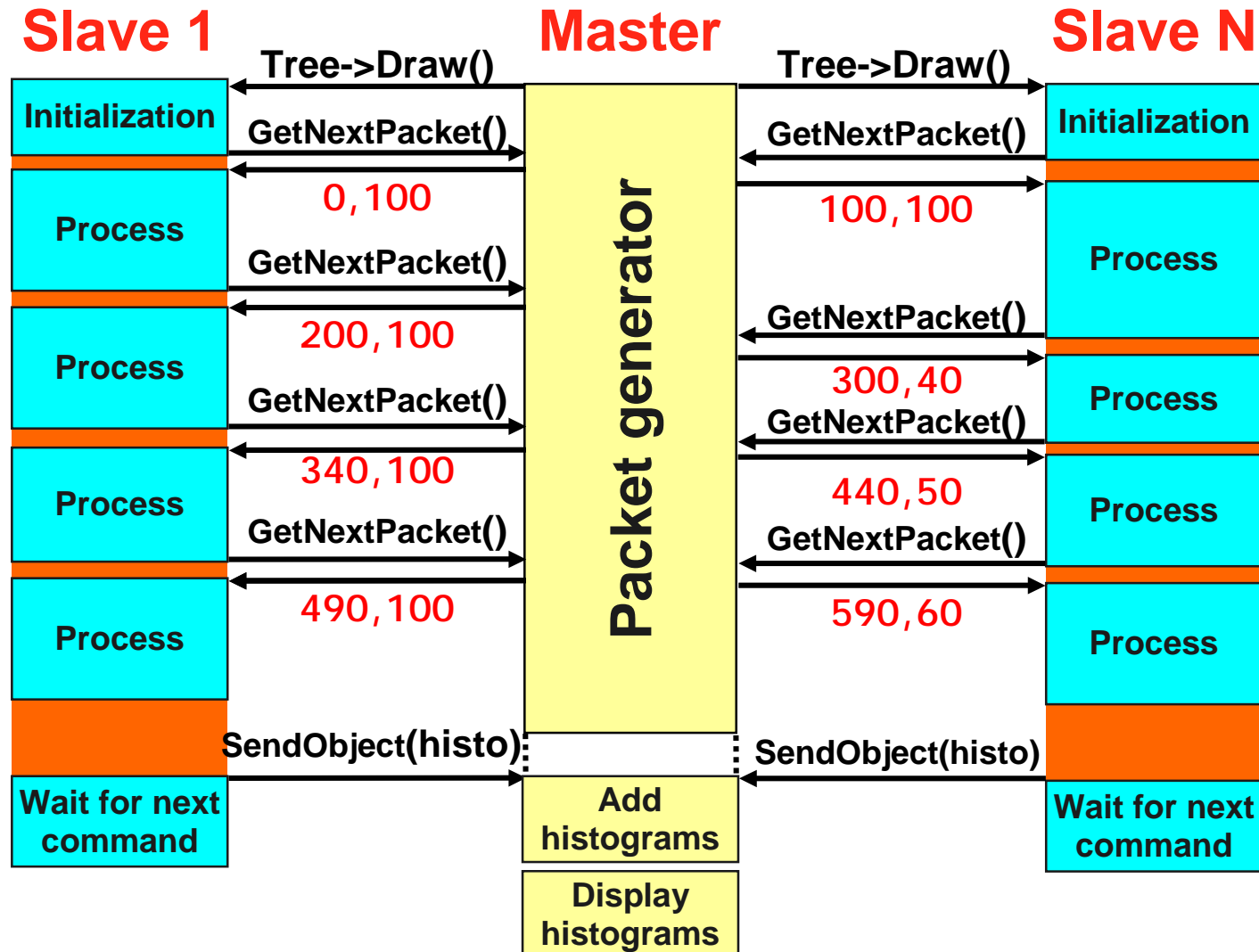
root [4] gROOT->Proof()
*** Proof slave server : pcna49a.cern.ch started ***
*** Proof slave server : pcna49b.cern.ch started ***
*** Proof slave server : pcna49c.cern.ch started ***
*** Proof slave server : pcna49d.cern.ch started ***
*** Proof slave server : pcna49e.cern.ch started ***
Real time 0:0:4, CP time 0.140

root [5] T49->Draw("fPx")
Real time 0:0:3, CP time 0.240
```





Workflow For Tree Analysis



PROOF Session Statistics



```
root [6] T49->Print("p")
Total events processed:                10585
Total number of packets:               147
Default packet size:                  100
Smallest packet size:                  20
Average packet size:                   72.01
Total time (s):                        2.78
Average time between packets (ms):     10.93
Shortest time for packet (ms):         99
Number of active slaves:                5
    Number of events processed by slave 0: 1890
    Number of events processed by slave 1: 2168
    Number of events processed by slave 2: 2184
    Number of events processed by slave 3: 2667
    Number of events processed by slave 4: 1676
```



PROOF Error Handling

- Handling death of PROOF servers
 - death of master
 - fatal, need to reconnect
 - death of slave
 - master will resubmit packets of death slave to other slaves
- Handling of ctrl-c
 - OOB message is send to master, and forwarded to slaves, causing soft/hard interrupt



PROOF Authentication

- PROOF supports secure and un-secure authentication mechanisms
 - Un-secure
 - mangled password send over network
 - Secure
 - SRP, Secure Remote Password protocol (Stanford Univ.), public key technology
 - Soon: Globus authentication



PROOF Grid Interface

- PROOF can use Grid Resource Broker to detect which nodes in a cluster can be used in the parallel session
- PROOF can use Grid File Catalogue and Replication Manager to map LFN's to chain of PFN's
- PROOF can use Grid Monitoring Services

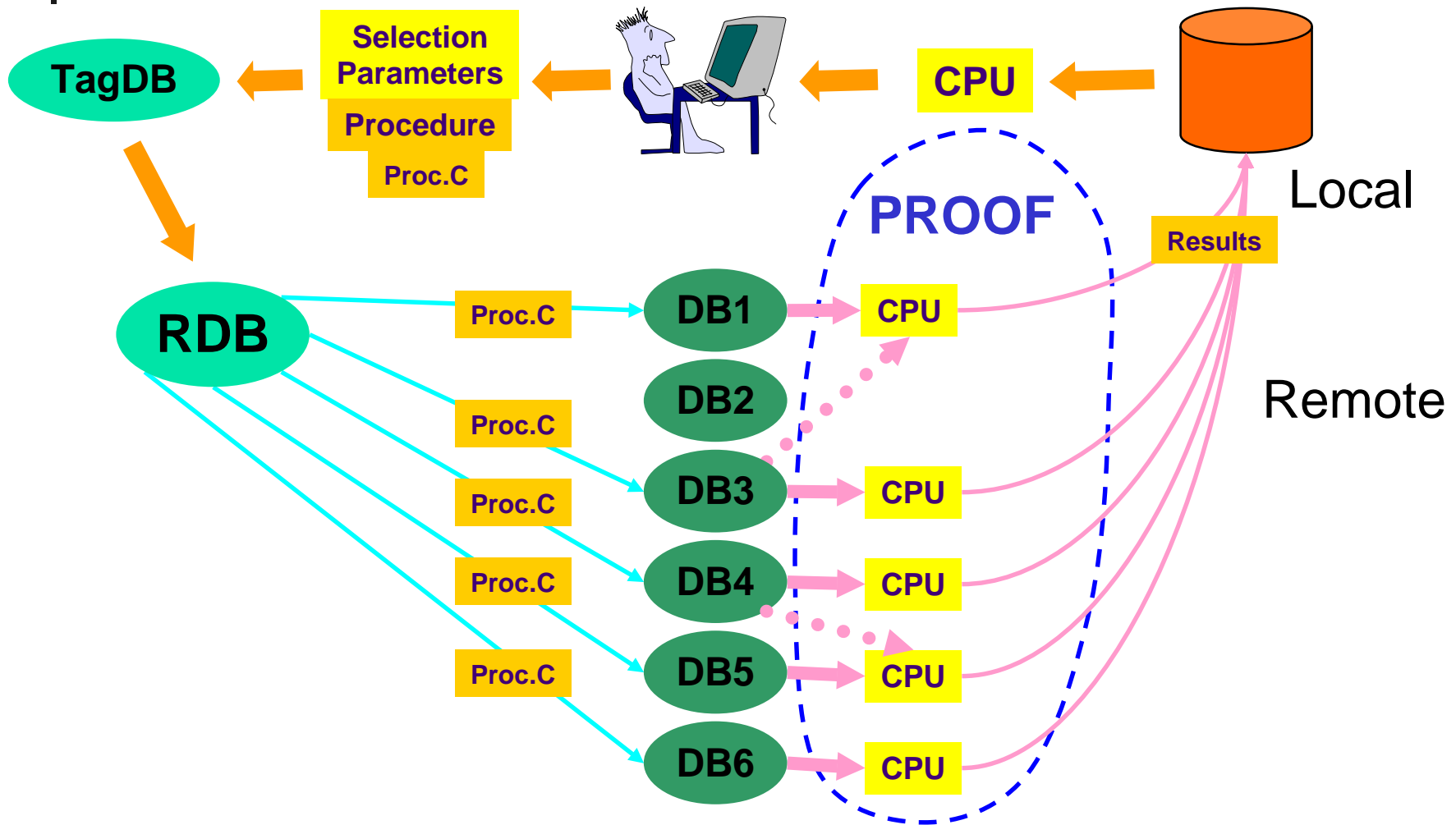


Setting Up PROOF

- Install ROOT system
- For automatic execution of daemons add proofd and rootd to /etc/inetd.conf and /etc/services (not mandatory, servers can be started by users)
 - The rootd (1094) and proofd (1093) port numbers have been officially assigned by IANA
- Setup proof.conf file describing cluster
- Setup authentication files (globally, users can override)



PROOF and the GRID





New Grid Features in ROOT



Main Grid Issues

- Distributed computing over wide area networks (WAN's). Requires:
 - efficient use of WAN pipes
 - user authentication
 - file catalogue and file replication
 - resource allocation and brokering
 - resource monitoring
 - etc.



Long Fat Pipes

- Long fat pipes are WAN links with a large bandwidth*delay product
- For optimal performance keep pipe full
- By default this is not the case
 - maximum TCP buffer size is 64KB
 - for a pipe with a 192KB bandwidth*delay product the pipe is empty 60% of the time





TCP Window Scaling (RFC 1323)

- A solution is to use a TCP buffer size equal to the bandwidth*delay product
- This support for large TCP buffers (window scaling) is described in RFC 1323



- Problem: system administrators are needed to change maximum TCP buffer sizes on source and destination machines, e.g. for Linux:
 - `echo 200000 > /proc/sys/net/core/rmem_max`



Parallel Sockets

- Buffer is striped over multiple sockets in equal parts
- Ideal number of parallel sockets depends on bandwidth*delay product (assuming default 64KB TCP buffer size). No system manager needed to tune network



- Same performance as with large buffers



New Grid Features in ROOT

- Parallel socket classes, TPSTSocket and TPSTServerSocket, that derive from TSocket and TServerSocket
- TNetFile and rootd daemon modified to use parallel sockets
- New TFTP class using parallel sockets and rootd daemon



Parallel FTP

- Parallel FTP via the TFTP class and the rootd daemon
- Uses the TPSSocket class
- Supports all standard ftp commands
- Anonymous ftp
- Performance, CERN - GSI:
 - wu-ftp: 1.4 MB/s
 - TFTP: 2.8 MB/s



Coming soon...

- Interface to Grid authentication service
- Interface to Grid file catalog
- Interface to Grid resource broker