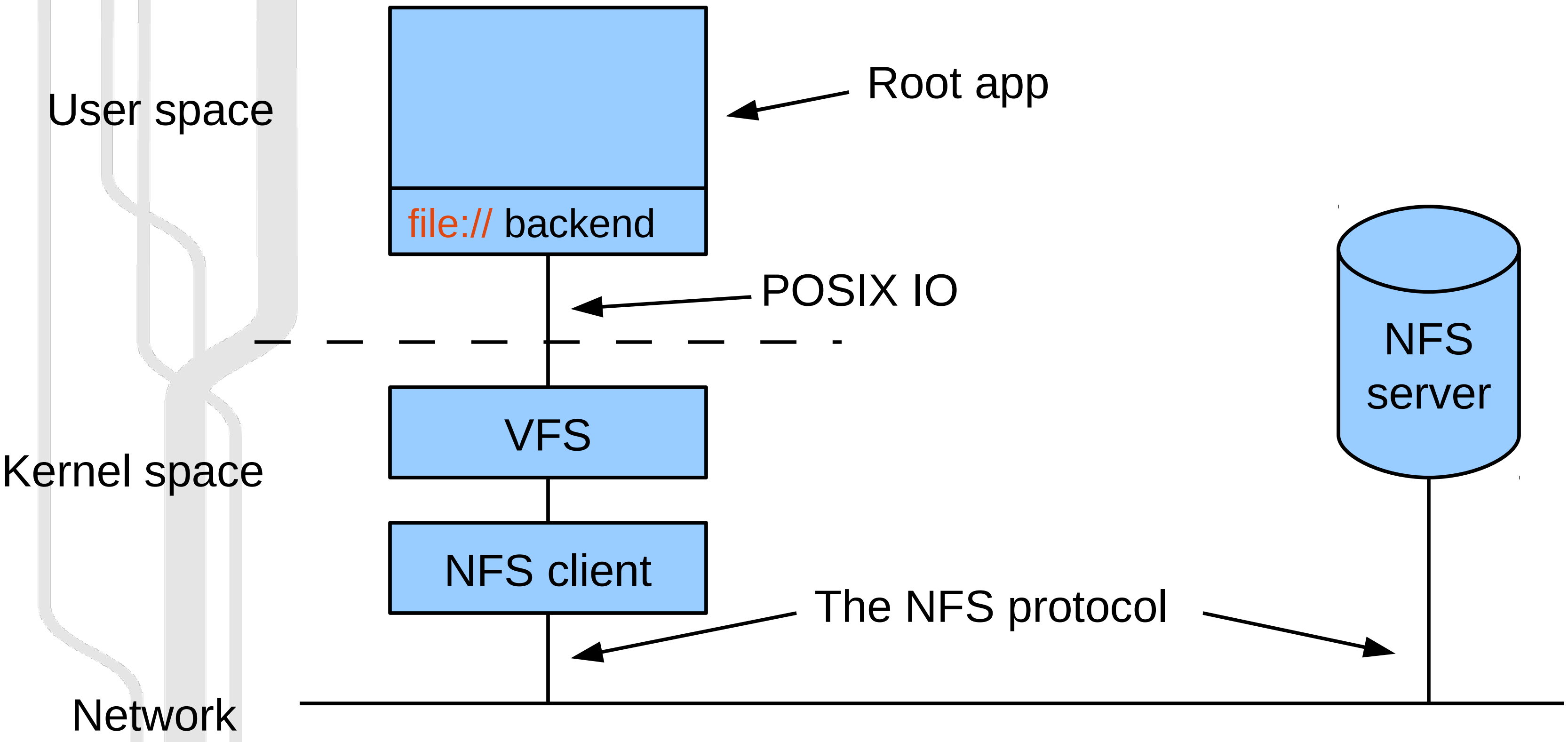
NFS 4.1
11 Reasons You Should Care

Jean-Philippe Baud, gLite
Gerd Behrmann, NDGF
Patrick Fuhrmann, dCache.org
Yves Kemp, DESY
Tigran Mkrtchyan, dCache.org

Thanks to Rene Brun, ROOT

What are we talking about?



Reason 1

High latency link performance

- Components
  - Allows batching of several commands, e.g. open, read, read, read, into one round-trip
- Delegations
  - Further reduces number of over the wire operations
  - Uses bidirectional RPC for notifications

Reason 2

Proper authentication and authorization

- Kerberos
  - But other schemes can be substituted
  - x509 is under evaluation
- ACLs

Reason 3

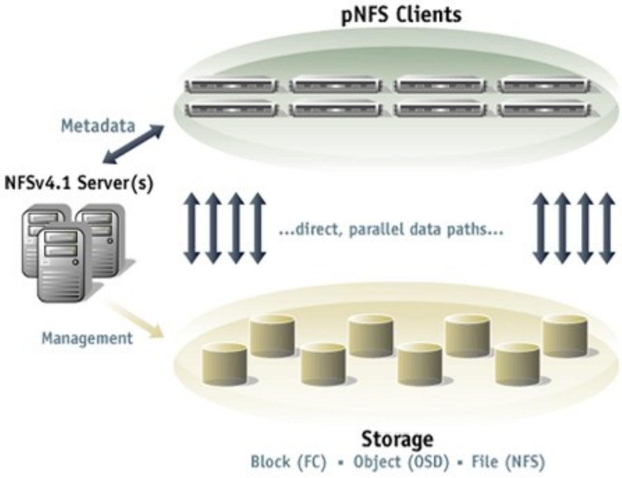
Sessions

- Introduced in NFS 4.1
- Decouples transport from client
- Exactly ones semantics
  - Due to duplicate request cache
- Mount over TCP and data optionally over alternative channels (like RDMA)

Reason 4

Parallel NFS

- Introduced in NFS 4.1
- Facilitates direct connections between clients and data nodes in distributed storage servers!
- Allows striping
  - e.g. concurrent read from multiple replicas


pNFS Clients
Metadata
NFSv4.1 Server(s)
...direct, parallel data paths...
Management
Storage
Block (FC) • Object (OSD) • File (NFS)

Reason 5

Standardization

- RFC 5661: Network File System (NFS) Version 4 Minor Version 1 Protocol
- IETF Proposed Standard
- No more proprietary protocol zoo
- Unified client stack for all the different servers

Reason 6

Backed by industry heavyweights

- A potential path to using off the shelf solutions in the future

NetApp™
Go further, faster
Microsoft®
ORACLE®
IBM®
panasas®
EMC²®
where information lives®

Reason 7

Client availability

- Linux client since 2.6.32
  - Parallel NFS client probably in 2.6.36
- Solaris driver available, but not shipped with Solaris yet
- Windows driver exists, but not published yet
- Redhat has builds for Fedora 12, 13, rawhide with pNFS
- Redhat Enterprise Linux is expected to have pNFS in 6.1

Reason 8

Server availability

- Industry
  - Netapp, Panasas, Oracle, EMC, IBM and others have hardware products in the pipeline
  - Waiting for broad client availability
- WLCG
  - dCache ships with NFS 4.1 now
  - DPM prototype before CHEP

Reason 9

Clients provided by industry

- In-kernel client provides real POSIX IO
- State-of-the-art caching is provided by the OS, tuned for a wide range of use cases by experts in the field
- No need to modify apps (you use the file:// protocol)

Reason 10

Funding

- Secured for next three years; after that explicit funding should not be necessary.
- EMI funds implementation of NFS 4.1 in DPM and continued improvement of NFS in dCache
- HGF (Helmholtz Alliance - Physics at the Terascale) funds implementation of NFS 4.1 in dCache

European Middleware Initiative

PHYSICS AT THE TERA SCALE

Helmholtz Alliance

Reason 11

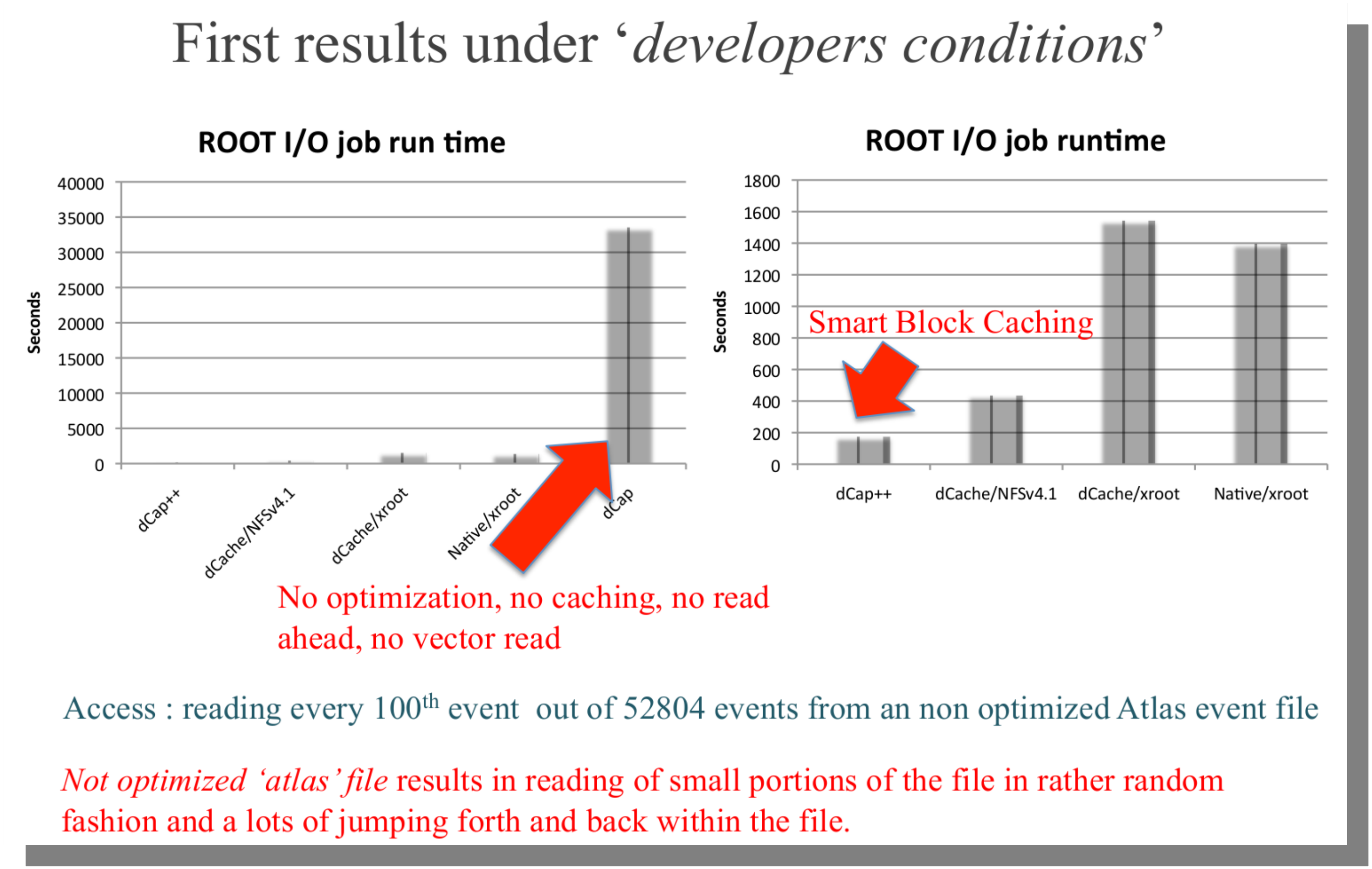
Simple migration path

- Clients use file://

  Unifies access to dCache, DPM, GPFS+Storm, etc.
- No data migration
- Full access to all existing features such as scheduling, SRM
- Legacy app support through the classic proprietary protocols like DCAP and RFIO

One more thing

HEPIX 2010



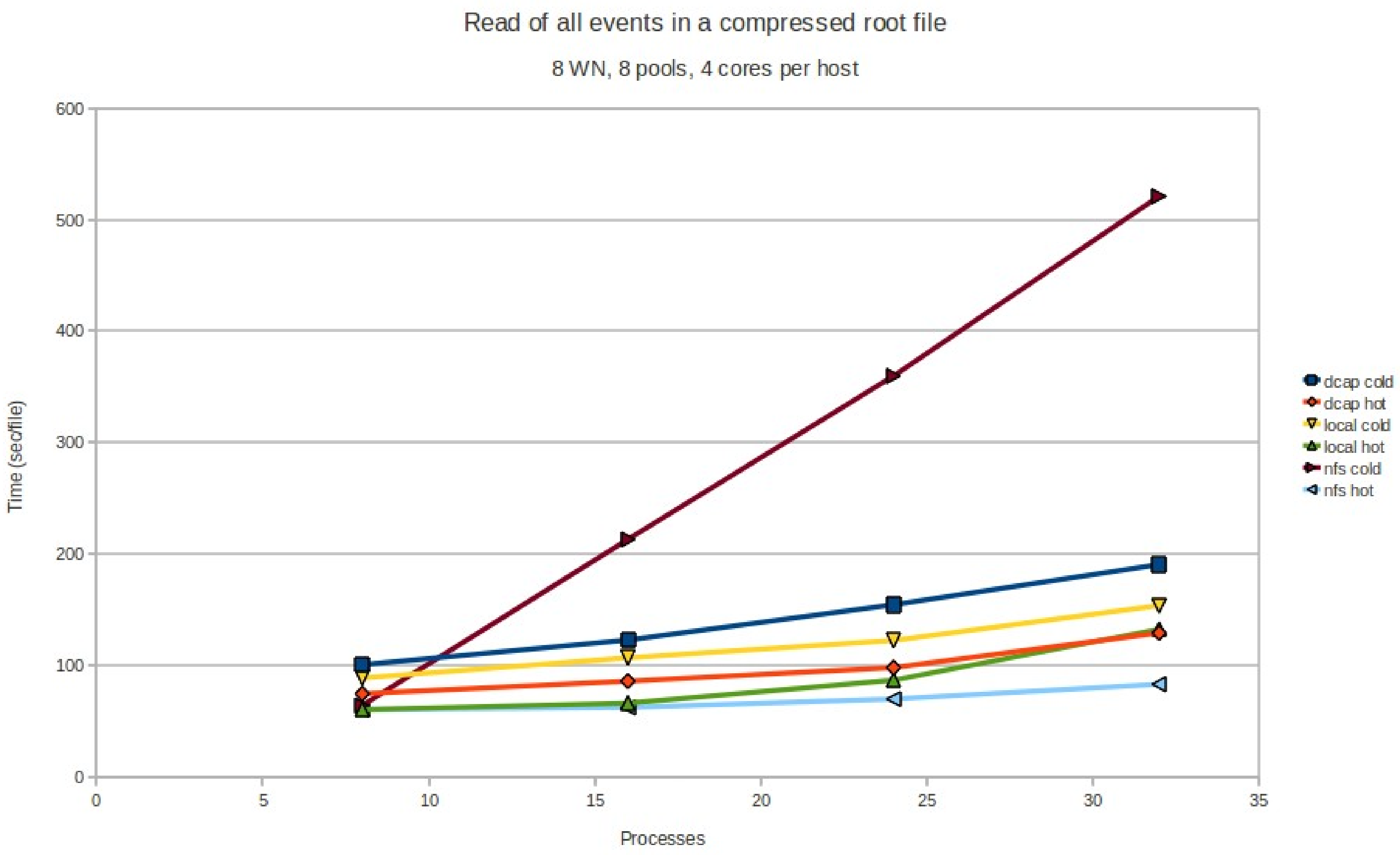
First results under ‘*developers conditions*’

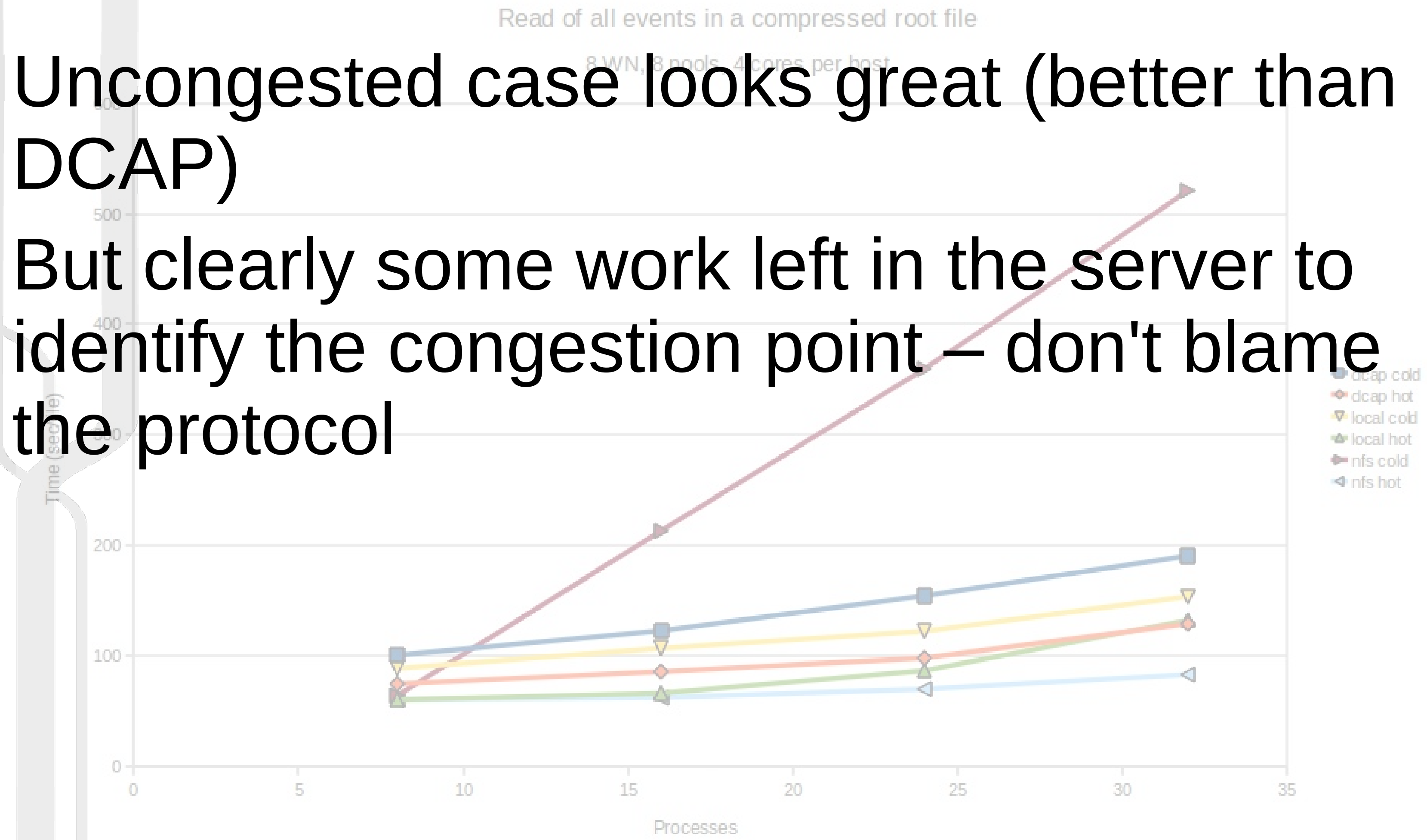
No optimization, no caching, no read ahead, no vector read

Access : reading every 100th event out of 52804 events from an non optimized Atlas event file

Not optimized ‘atlas’ file results in reading of small portions of the file in rather random fashion and a lots of jumping forth and back within the file.

Source: Patrick Fuhrmann


Read of all events in a compressed root file
8 WN, 8 pools, 4 cores per host
Time (sec/file)
600
500
400
300
200
100
0
0
5
10
15
20
25
30
35
Processes
dcap cold
dcap hot
local cold
local hot
nfs cold
nfs hot



- Uncongested case looks great (better than DCAP)
- But clearly some work left in the server to identify the congestion point – don't blame the protocol