LHCONE Point-to-Point Service Workshop - CERN Geneva

Eric Boyd, Internet2

Slides borrowed liberally from Artur, Inder, Richard, and other workshop presenters

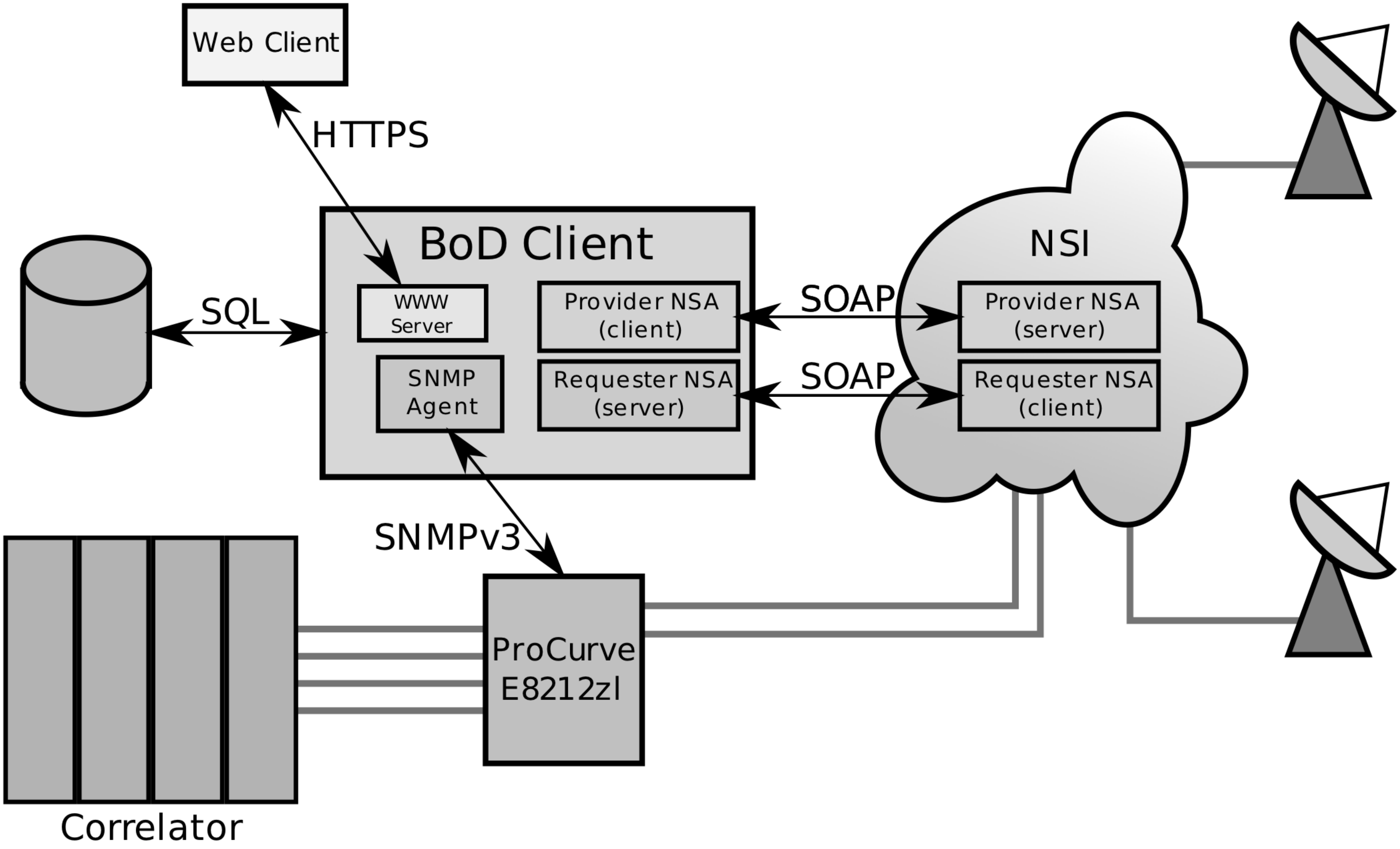
Who was there?

- Network Operators
- LHCONE Application Developers
- https://indico.cern.ch/conferenceDisplay.py?confId=215393

First Part: Bandwidth on Demand

- Introduction to BoD Concepts – Inder Monga
- NSI – Jerry Sobieski
- Circuit Service Deployments
  - North America et al – Eric Boyd
  - GEANT – Tangui Coulouarn
- Example of what Data Intensive Science can do with BoD: JIVE – Paul Boven

NEXPReS NSI Agent



Experience with NSI so far

- Standard is still in development

- There is no ‘NSI-cloud’ yet
  - Every new connection has to be provisioned
  - Lots of work for ‘first user’

- Layer 2 service: think about your IP assignments and routing
  - Limited number of ‘labels’ (VLAN tags)

- Not a production service yet
  - Testbeds have limited bandwidth
  - Extra connections in/out of testbeds
  - Often no bandwith enforcement

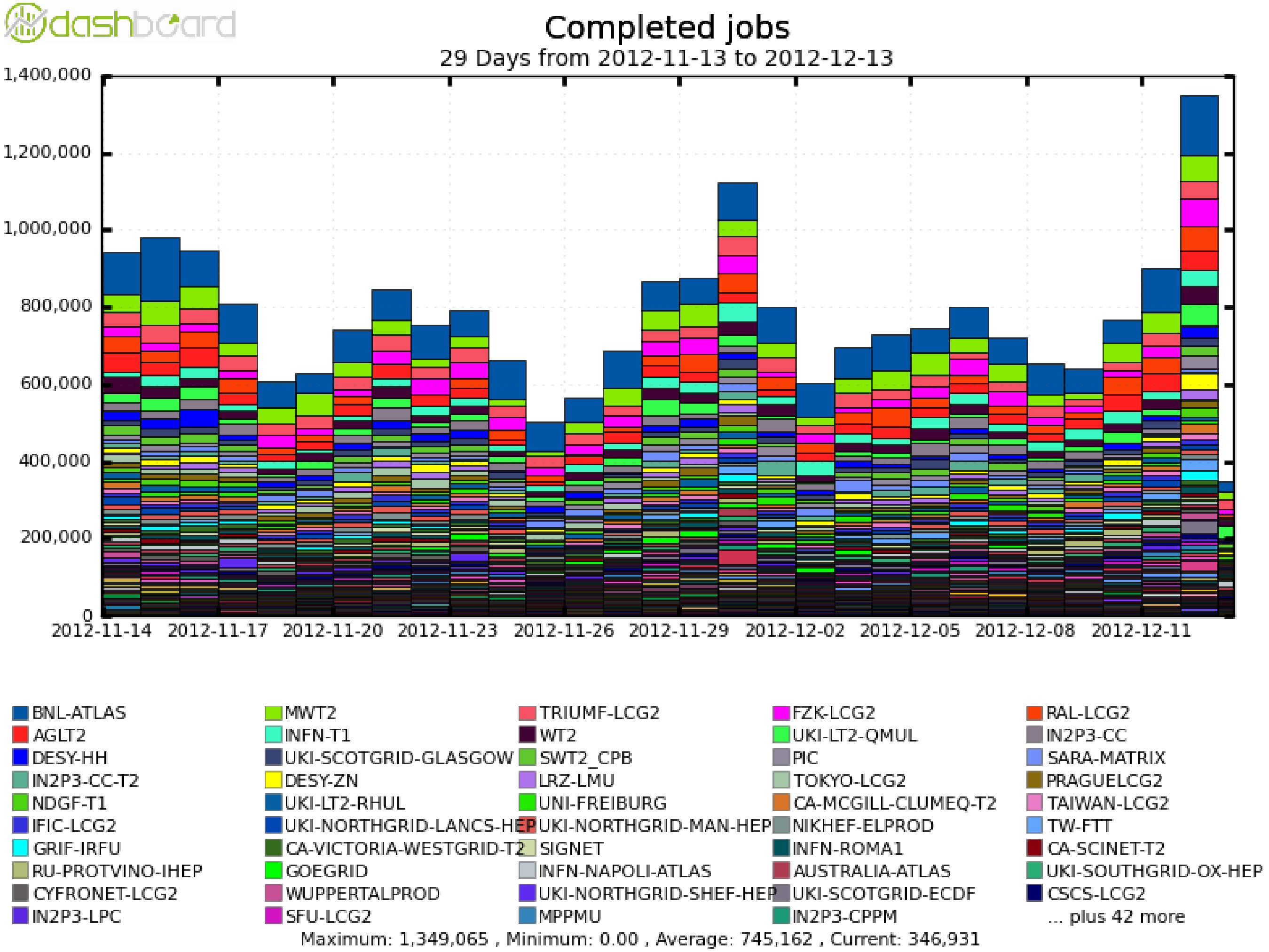
- Different versions of NSI standard and software
  - AutoBahn, OpenDRAC, OpeNSA, NEXPReS client

- Very good support from NRENs, GÉANT

Second Part: LHC Computation Middleware and Workflow

- Networking and Workload Management – Kaushik De
- ATLAS and CMS Data Management Tools and Federated Data Store Implementations – Daniele Bonacorsi
- ALICE Data Access Model – Costin Grigoras
- ANSE Project Overview – Artur Barczyk

PanDA Scale



Number of Analysis Users: (unique)

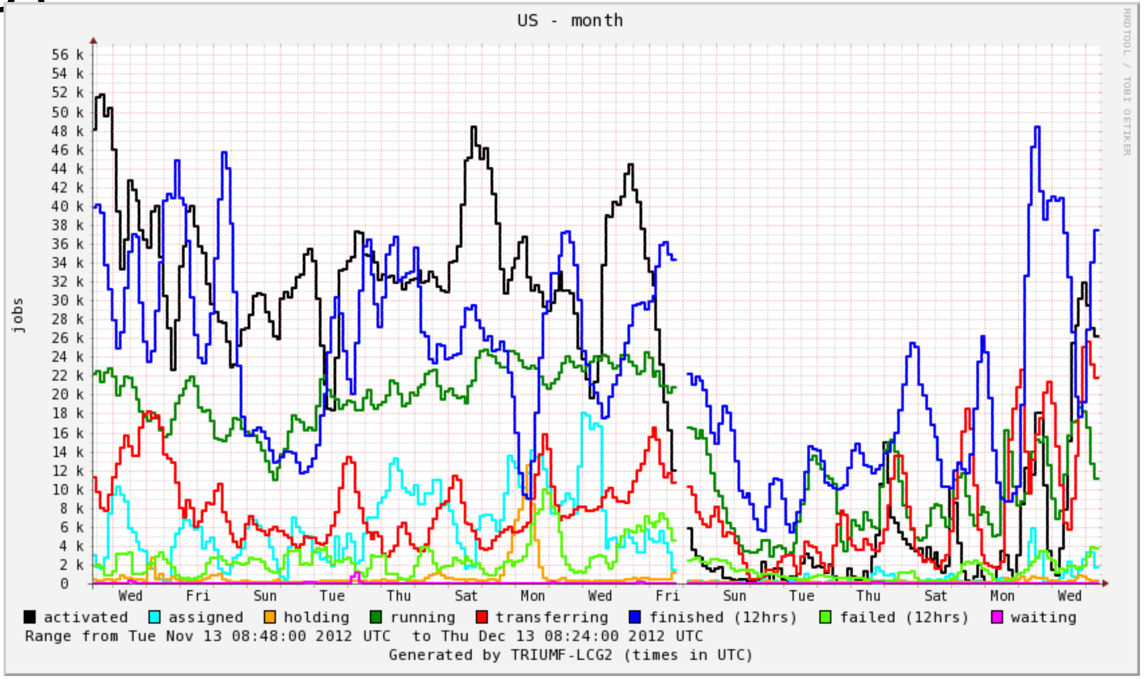
Users in the last 3 days : **458;** 7: **623;** 30: **941;** 90: **1240;** 180: **1547;**

PanDA Philosophy

- PanDA WMS design goals:
  - Achieve high level of automation to reduce operational effort
  - Flexibility in adapting to evolving hardware and network capabilities
  - Support diverse and changing middleware
  - Insulate user from hardware, network, middleware, and all other complexities of the underlying system
  - Unified system for organized production and user analysis
  - Incremental and adaptive software development
- PanDA and DDM
  - PanDA uses a independent and asynchronous Distributed Data Management system (DDM) called DQ2 in ATLAS
  - DDM is tightly coupled to networking – will not address here

Job States

- Panda jobs go through a succession of steps tracked in DB
  - Defined
  - Assigned
  - Activated
  - Running
  - Holding
  - Transferring
  - Finished/failed



Assigned Jobs

- Assigned -> Activated workflow
  - Group of jobs are assigned to a site by PanDA brokerage
  - For missing input files, data transfer is requested asynchronously
  - PanDA waits for "transfer completed" callback from DDM system to activate jobs for execution
  - Network data transfer plays crucial role in this workflow
- Can network technology help assigned->activated transition?
  - Can we use network provisioning in this step?
  - Jobs are reassigned if transfer times out (fixed duration) – can knowledge of network status help reduce the timeout?
  - Can modification of network path help?

Transferring Jobs

- Transferring state
  - After job execution is completed, asynchronous data transfer is requested from DDM
  - Callback is required for successful job completion
- How can network technology help?
  - Similar questions as assigned state
  - Very long timeout delays completion – can network status info help
  - Can we balance CPU resource vs Network resource
  - At what point can we give up on transfer and rerun the job?

Summary

- In the past WMS assumed:
  - Network is available and ubiquitous
  - As long as we implement timeouts, workflow will progress smoothly
  - Computing models can tell us how to design workflows
- What we learned from the LHC:
  - Flexibility in WMS design is more important than computing model
  - Network evolution drives WMS evolution
  - We should start thinking about Network as resource
  - WMS should use network information actively to optimize workflow
  - Resource provisioning could be important for the future
- The future:
  - **A**dvanced **N**etwork **S**ervices for **E**xperiments (ANSE), NSF funded (Caltech, Michigan, Vanderbilt and U Texas Arlington)
  - Next Generation Workload Management and Analysis System for Big Data, PANDA integration with networking, DOE funded (BNL, U Texas Arlington)

Second Part: LHC Computation Middleware and Workflow

- Networking and Workload Management – Kaushik De
- ATLAS and CMS Data Management Tools and Federated Data Store Implementations – Daniele Bonacorsi
- ALICE Data Access Model – Costin Grigoras
- ANSE Project Overview – Artur Barczyk

Differences in DM implementation choices (or status)

ATLAS handles a larger **number of files** in the DM system than CMS

- ATLAS injects all files in their system, including in particular all user file. As a result, ATLAS transfers many more files with DDM than CMS does with PhEDEx, and have a lot of small files in the system

In ATLAS **a file can belong to multiple datasets**, while in CMS the datasets contain unique files

- CMS deletes on the dataset level or data block level, while ATLAS deletes on the file level

ATLAS and CMS use different **catalogues** for mapping purposes

- ATLAS currently uses a central catalog (LFC) to perform LFN-to-PFN mapping. This is going to change in Rucio during LS1, it will use implicit LFN-to-PFN conversion rules, more similar to what CMS does

ATLAS is going towards a less MONARC-hierarchical and "full **mesh**" transfer model like CMS

- ATLAS uses a more hierarchical model of T0→T1→T2 transfers still, similar to what CMS had in ~2008. Many T2s are still restricted to sharing data only with their "regional" T1, though now ATLAS is also moving to a "full mesh" like CMS and there are also many T2s that transfer data to/from any T1 (so-called "T2D"s). Following from the previous point, ATLAS has enabled multi-hop transfers to get data from T2_X to T2_Y through some T1

CMS is now moving to **disk vs tape separation** like ATLAS (and others) already did

- CMS has not yet finalized a full separation among disk and tape resources at the T1 level, still sending most of the data to tape backend. ATLAS has separated "T1 Disk" from "T1 tape" through space tokens a long time ago

ATLAS system is somehow more "**dynamic**" than CMS, soon doing something similar

- A large fraction of ATLAS data placement is automated, with dynamic subscription and deletion of datasets based on data popularity rather than relying on human action on requests. CMS has equipped their system with data popularity evaluation mechanisms, and will soon (2013) enable deletions, but so far the system is less "dynamic"

ATLAS operates systems more **centrally**, CMS has a **distributed** deployment of components

- ATLAS DDM is operated centrally at CERN for all sites, talking to the site SEs only with "remote" tools (i.e. SRM, FTS). CMS PhEDEx is a distributed transfer system with central agents at CERN and site agents at each site, managed by the CMS contacts on site (despite it has been demonstrated years ago that technically all agents could be run elsewhere)

Not all items are relevant to the topic of this workshop, but a couple are.

Transfers: network-awareness? [1/2]

Where data management could become network-aware?

Level 1: “*high*”-level i.e. **activity planning**

- in some sense, above both Data and Workload Management
- the planning (e.g. dependencies, completion times, ..) drive workflow scheduling and executions
  - network bandwidth reservation could be triggered in advance based on planning details/needs

Level 2: “*medium*”-level i.e. **transfer “routing”**

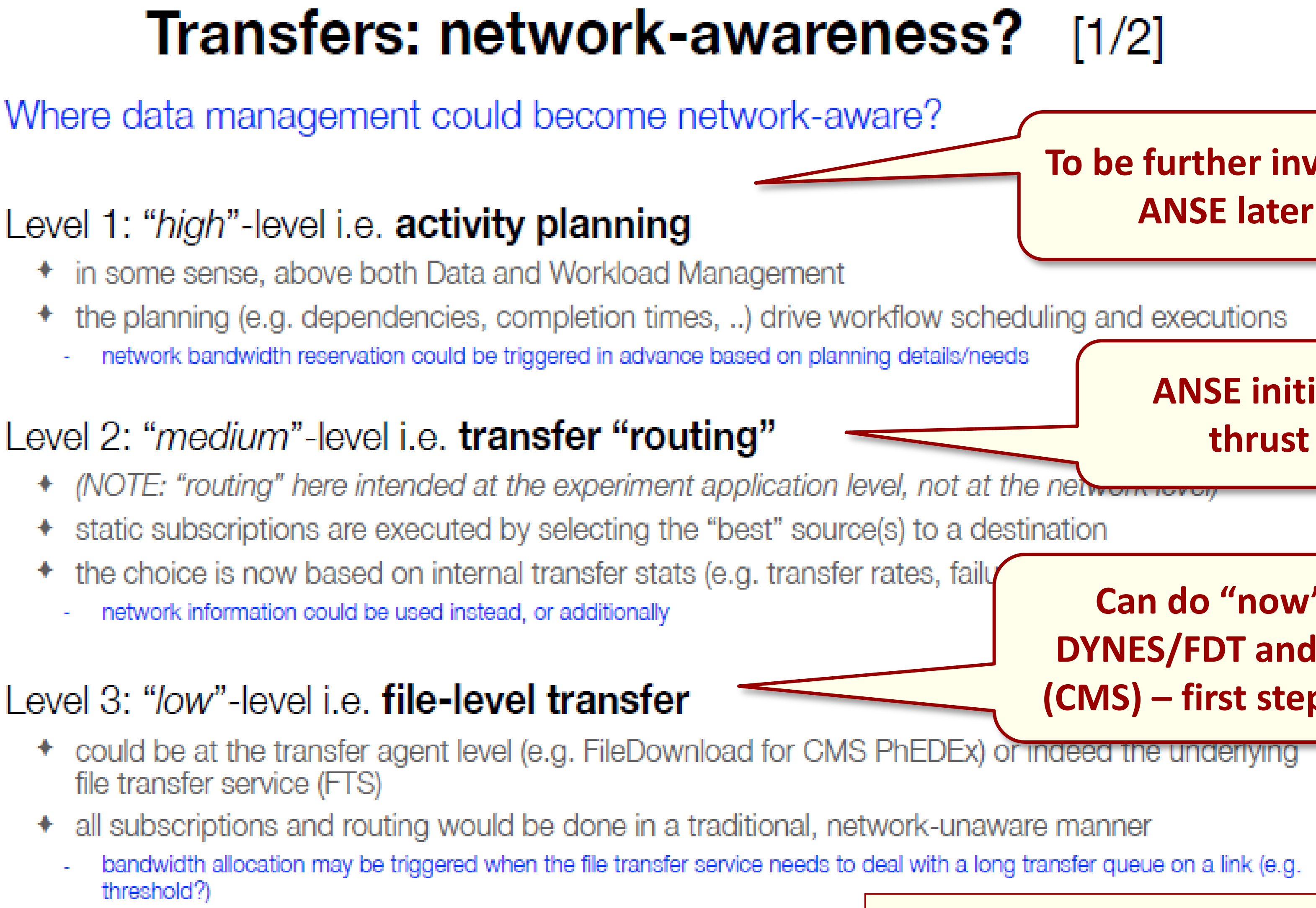
- *(NOTE: “routing” here intended at the experiment application level, not at the network level)*
- static subscriptions are executed by selecting the “best” source(s) to a destination
- the choice is now based on internal transfer stats (e.g. transfer rates, failures, .. over last days/hrs)
  - network information could be used instead, or additionally

Level 3: “*low*”-level i.e. **file-level transfer**

- could be at the transfer agent level (e.g. FileDownload for CMS PhEDEx) or indeed the underlying file transfer service (FTS)
- all subscriptions and routing would be done in a traditional, network-unaware manner
  - bandwidth allocation may be triggered when the file transfer service needs to deal with a long transfer queue on a link (e.g. threshold?)

Examples? See next slide.

Where to Attach?



Transfers: network-awareness? [1/2]

Where data management could become network-aware?

Level 1: "*high*"-level i.e. **activity planning**

- in some sense, above both Data and Workload Management
- the planning (e.g. dependencies, completion times, ..) drive workflow scheduling and executions
  - network bandwidth reservation could be triggered in advance based on planning details/needs

Level 2: "*medium*"-level i.e. **transfer "routing"**

- *(NOTE: "routing" here intended at the experiment application level, not at the network level)*
- static subscriptions are executed by selecting the "best" source(s) to a destination
- the choice is now based on internal transfer stats (e.g. transfer rates, failu
  - network information could be used instead, or additionally

Level 3: "*low*"-level i.e. **file-level transfer**

- could be at the transfer agent level (e.g. FileDownload for CMS PhEDEx) or indeed the underlying file transfer service (FTS)
- all subscriptions and routing would be done in a traditional, network-unaware manner
  - bandwidth allocation may be triggered when the file transfer service needs to deal with a long transfer queue on a link (e.g. threshold?)

To be further investigated in ANSE later stage

ANSE initial main thrust axis

Can do "now" with DYNES/FDT and PhEDEx (CMS) – first step in ANSE

D. Bonacorsi, CMS, at LHCONE P2P Service Workshop, Dec. 2012

Transfers: network-awareness? [2/2]

Level 1: "*high*"-level i.e. **activity planning**

- ✦ *subscriptions in Rucio may be an interesting candidate for a choice at this level?*
  - replica management based on **Replication Rules** defined on datasets/containers. Each rules is owned by a Rucio "**account**", and defines the minimum # of replicas that have to be available on a Rucio Storage Element (**RSE**), i.e. a storage space with attributes. RSEs can be grouped in logical ways (e.g. CLOUD=US, or Tier=1). Accounts manage (and are charged) for their own data with replication rules defined on datasets/containers and lists of RSEs
  - *Could a translation of such a rule into a concrete list of transfer tasks be engineered to be optimized on the basis of network-aware information? (e.g. naively: "choose the source RSE with best connection to the destination RSE"?)*

Level 2: "*medium*"-level i.e. **transfer "routing"**

- ✦ *ATLAS Site Services or PhEDEx FileRouter could use network info at this level?*

Level 3: "*low*"-level i.e. **file-level transfer**

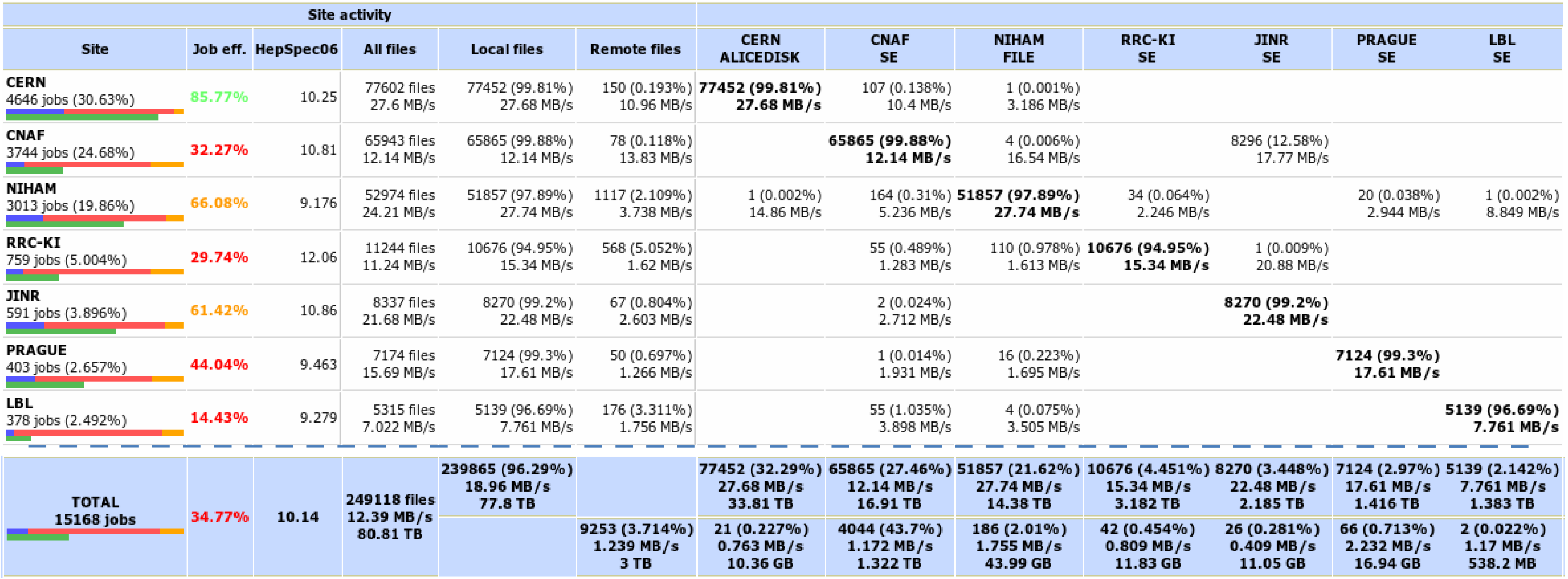
- ✦ *e.g. FDT used as the backend in the FileDownload agent in PhEDEx on the /Debug instance on just one link may be an existing proof of concept of a choice at this level?*

Food for thoughts...

Second Part: LHC Computation Middleware and Workflow

- Networking and Workload Management – Kaushik De
- ATLAS and CMS Data Management Tools and Federated Data Store Implementations – Daniele Bonacorsi
- ALICE Data Access Model – Costin Grigoras
- ANSE Project Overview – Artur Barczyk

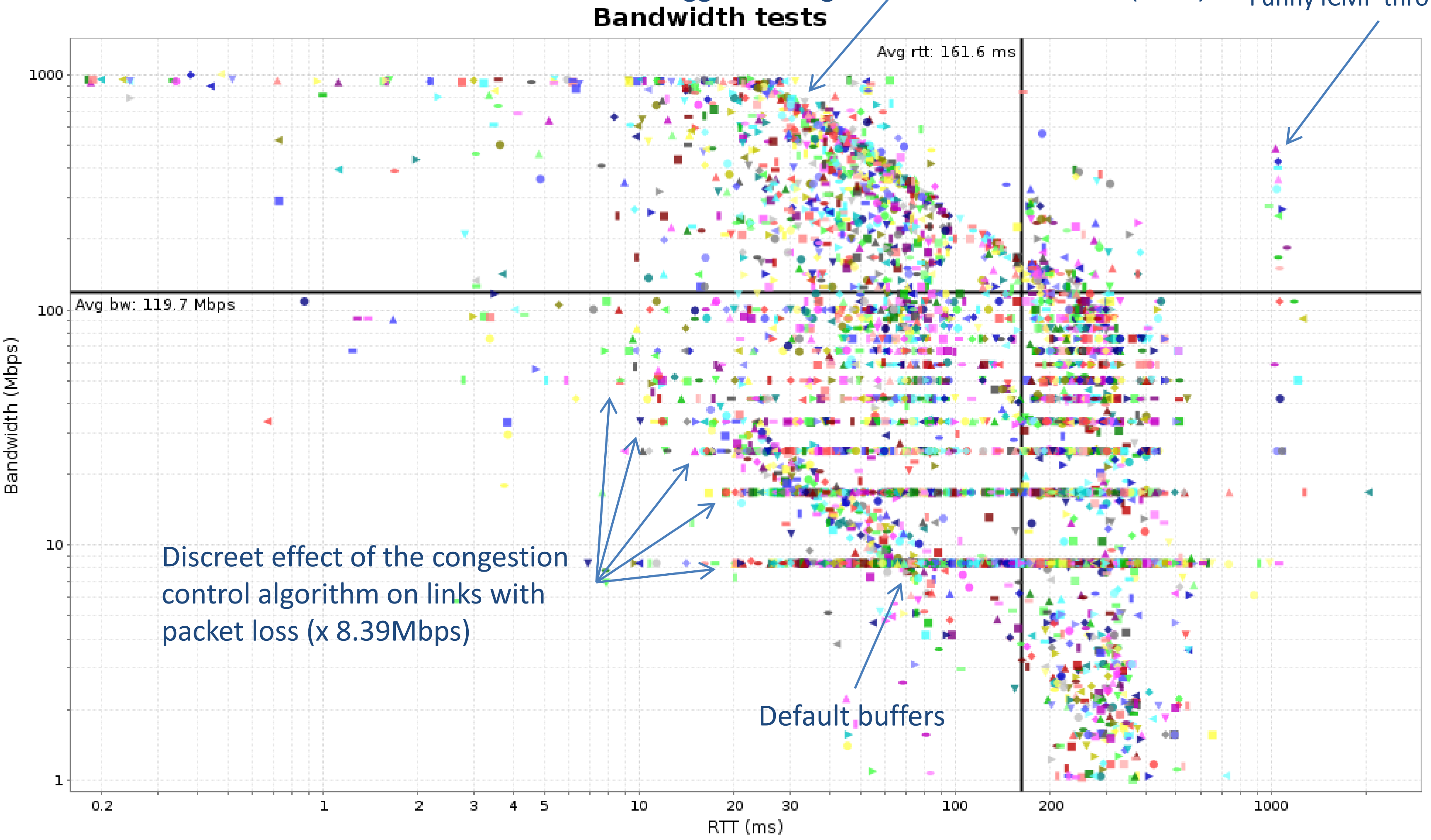
A particular analysis task …



Site activity												
Site	Job eff.	HepSpec06	All files	Local files	Remote files	CERN ALICEDISK	CNAF SE	NIHAM FILE	RRC-KI SE	JINR SE	PRAGUE SE	LBL SE
CERN 4646 jobs (30.63%)	85.77%	10.25	77602 files 27.6 MB/s	77452 (99.81%) 27.68 MB/s	150 (0.193%) 10.96 MB/s	**77452 (99.81%) 27.68 MB/s**	107 (0.138%) 10.4 MB/s	1 (0.001%) 3.186 MB/s				
CNAF 3744 jobs (24.68%)	32.27%	10.81	65943 files 12.14 MB/s	65865 (99.88%) 12.14 MB/s	78 (0.118%) 13.83 MB/s		**65865 (99.88%) 12.14 MB/s**	4 (0.006%) 16.54 MB/s		8296 (12.58%) 17.77 MB/s		
NIHAM 3013 jobs (19.86%)	66.08%	9.176	52974 files 24.21 MB/s	51857 (97.89%) 27.74 MB/s	1117 (2.109%) 3.738 MB/s	1 (0.002%) 14.86 MB/s	164 (0.31%) 5.236 MB/s	**51857 (97.89%) 27.74 MB/s**	34 (0.064%) 2.246 MB/s		20 (0.038%) 2.944 MB/s	1 (0.002%) 8.849 MB/s
RRC-KI 759 jobs (5.004%)	29.74%	12.06	11244 files 11.24 MB/s	10676 (94.95%) 15.34 MB/s	568 (5.052%) 1.62 MB/s		55 (0.489%) 1.283 MB/s	110 (0.978%) 1.613 MB/s	**10676 (94.95%) 15.34 MB/s**	1 (0.009%) 20.88 MB/s		
JINR 591 jobs (3.896%)	61.42%	10.86	8337 files 21.68 MB/s	8270 (99.2%) 22.48 MB/s	67 (0.804%) 2.603 MB/s		2 (0.024%) 2.712 MB/s			**8270 (99.2%) 22.48 MB/s**		
PRAGUE 403 jobs (2.657%)	44.04%	9.463	7174 files 15.69 MB/s	7124 (99.3%) 17.61 MB/s	50 (0.697%) 1.266 MB/s		1 (0.014%) 1.931 MB/s	16 (0.223%) 1.695 MB/s			**7124 (99.3%) 17.61 MB/s**	
LBL 378 jobs (2.492%)	14.43%	9.279	5315 files 7.022 MB/s	5139 (96.69%) 7.761 MB/s	176 (3.311%) 1.756 MB/s		55 (1.035%) 3.898 MB/s	4 (0.075%) 3.505 MB/s				**5139 (96.69%) 7.761 MB/s**
TOTAL 15168 jobs	**34.77%**	**10.14**	**249118 files 12.39 MB/s 80.81 TB**	**239865 (96.29%) 18.96 MB/s 77.8 TB**		**77452 (32.29%) 27.68 MB/s 33.81 TB**	**65865 (27.46%) 12.14 MB/s 16.91 TB**	**51857 (21.62%) 27.74 MB/s 14.38 TB**	**10676 (4.451%) 15.34 MB/s 3.182 TB**	**8270 (3.448%) 22.48 MB/s 2.185 TB**	**7124 (2.97%) 17.61 MB/s 1.416 TB**	**5139 (2.142%) 7.761 MB/s 1.383 TB**
					9253 (3.714%) 1.239 MB/s 3 TB	**21 (0.227%) 0.763 MB/s 10.36 GB**	**4044 (43.7%) 1.172 MB/s 1.322 TB**	**186 (2.01%) 1.755 MB/s 43.99 GB**	**42 (0.454%) 0.809 MB/s 11.83 GB**	**26 (0.281%) 0.409 MB/s 11.05 GB**	**66 (0.713%) 2.232 MB/s 16.94 GB**	**2 (0.022%) 1.17 MB/s 538.2 MB**

- IO-intensive analysis train run
- Small fraction of files accessed remotely
  - With the expected penalty
- However the external connection is the lesser issue …

Available bandwidth per stream



Suggested larger-than-default buffers (8MB)

Funny ICMP throttling

Discreet effect of the congestion control algorithm on links with packet loss (x 8.39Mbps)

Default buffers

Second Part: LHC Computation Middleware and Workflow

- Networking and Workload Management – Kaushik De
- ATLAS and CMS Data Management Tools and Federated Data Store Implementations – Daniele Bonacorsi
- ALICE Data Access Model – Costin Grigoras
- ANSE Project Overview – Artur Barczyk

Objectives and Approach

- **Deterministic, optimized workflow is the goal**
  - **Use network resource allocation along with storage and CPU resource allocation in planning data and job placement**
  - **Improve overall throughput and task times to completion**
- **Integrate advanced network-aware tools in the mainstream production workflows of ATLAS and CMS**
  - **use tools and deployed installations where they exist**
    - i.e. build on previous manpower investment in R&E networks
  - **extend functionality of the tools to match experiments' needs**
  - **identify and develop tools and interfaces where they are missing**
- **Green-Field, but not Terraforming**
  - **Introduce new/recent concepts**
  - **Build on several years of invested manpower, tools and ideas (some since the MONARC era)**

Summary

- ANSE project aims at integration of advanced network services in the LHC experiment's SW stacks
- Through interfaces to
  - Monitoring services (PerfSONAR-based, MonALISA)
  - Bandwidth reservation systems (through protocols like NSI and IDCP)

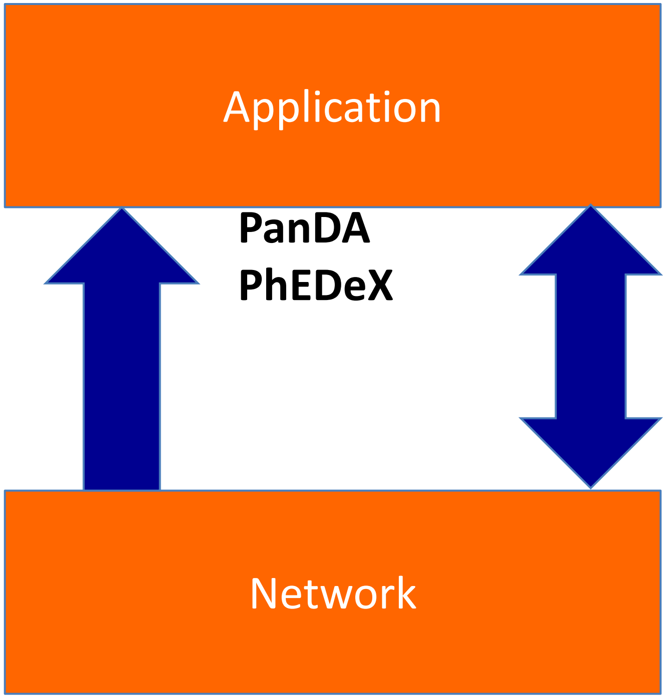
- Working with
  - PanDA system in ATLAS
  - PhEDEx in CMS

- The goal is to make deterministic workflows possible

Inder’s Summary of the discussion

Network Information

- Availability
- Performance
- Capabilities
- Statistics
- Topology?
- …



P2P Network Services

- Resource reservation
- Pt-MP
- Priority
- Modify
- …

Inder: Discussion from P2P workshop: Questions → Opportunities

- Applications need information from the network to help determine
  - What can it provide that will help choose the best Data Transfer Replica?
  - Where should I run my next job (A, A’ or A’’) – co-scheduling requirement
    - Is it better to move storage to compute or compute to storage?
  - Federated Storage Redirection
    - Choice of the source of traffic is just-in-time
    - Throughput monitoring, can tell the network when something is not working as expected
  - Application based routing of flows
    - If A → C is busy or blocked, can I move it from A → B → C
  - Middleware like Workflow Managers
    - Can we provide an aggregated view to the network

Inder: Network Services Questions

- Point to Multi-point data replication (or Multi-point to single point)
- Granularity of the Network Service request
  - Service limitations of the network
  - Can that be discovered end-to-end?
- Circuit-blocking response
  - What happens when network cannot provide the circuit?
  - Alternate suggestions from the network rather than yes/no answer
- Prioritization of various circuit requests
  - Bump one vs the other?
- Should the applications be multi-domain aware or agnostic
  - Network as a single black box or more visible?
- Ability to modify network paths – more duration or bandwidth
- How should applications model the network: Network as a resource or Network as a service

Takeaways – Richard’s Thoughts

- Experiments need to be able to manage the network & its resources and to interact with Panda and FedX.
- Users need the authority to allocate net resources
  need authi & authz mechanisms
- BOD usage
  - May need strict policing or floor with excess marked as LBE
  - Concern re integration time/duty cycle for policing; need for shaping of flows and effect of buffer over runs
  - May wish to lower the bandwidth of a BoD link
  - Need to have tools to know what possible BW/path can be requested both now and at a future time, then user will determine if reservation is useful (FedX).
    Network needs to return this information on request.
  -

Takeaways – Richard’s Thoughts

- Users (and networkers) want to know WHY a reservation failed or had poor performance
  - Also what to do about it
  - Need enough info to tell the net people what was wrong so can look at it.
- Is the network a black box?
- Need a global view of the network to be able to organise storage and access to data – not just moving 1 file but eg which replica to use?
- Topology info and “normal routes” important
  eg decide to move data lon-chi, not gen-chi but actually it flows lon-gen-chi.
- Chain or tree for NSI BoD?
  - Problems in the past with trees failing – need for info about each step
  - App could decide on the path
  - Will Client APIs talk only to local NREN?