Networking for the LHC Program

ANSE: Advanced Network Services for the HEP Community

Harvey B Newman

California Institute of Technology

LHCONE Workshop

Paris, June 17, 2013

ANSE: Advanced Network Services for Experiments

- **ANSE is a project funded by NSF’s CC-NIE program**
  - **Two years funding, started in January 2013, ~3 FTEs**
- **Collaboration of 4 institutes:**
  - **Caltech (CMS)**
  - **University of Michigan (ATLAS)**
  - **Vanderbilt University (CMS)**
  - **University of Texas at Arlington (ATLAS)**
- **Goal: Enable strategic workflow planning including network capacity as well as CPU and storage as a co-scheduled resource**
- **Path Forward: Integrate advanced network-aware tools with the mainstream production workflows of ATLAS and CMS**
  - **In-depth monitoring and Network provisioning**
  - **Complex workflows: *a natural match and a challenge for SDN***
- ***Exploit state of the art progress in high throughput long distance data transport, network monitoring and control***

ANSE - Methodology

- Use agile, managed bandwidth for tasks with levels of priority along with CPU and disk storage allocation.
  - Allows one to define goals for time-to-completion, with reasonable chance of success
  - Allows one to define metrics of success, such as the rate of work completion, with reasonable resource usage efficiency
  - Allows one to define and achieve “consistent” workflow
- Dynamic circuits a natural match
  - As in DYNES for Tier2s and Tier3s
- Process-Oriented Approach
  - Measure resource usage and job/task progress in real-time
  - If resource use or rate of progress is not as requested/planned, diagnose, analyze and decide if and when task replanning is needed
- Classes of work: defined by resources required, estimated time to complete, priority, etc.

Tool Categories

- *Monitoring (Alone):*
  - Allows *Reactive* Use: React to “events” (State Changes) or Situations in the network
    - Throughput Measurements ➡ Possible Actions:

      (1) Raise Alarm and continue (2) Abort/restart transfers (3) Choose different source
    - Topology (+ Site & Path performance) Monitoring ➡ possible actions:

      (1) Influence source selection

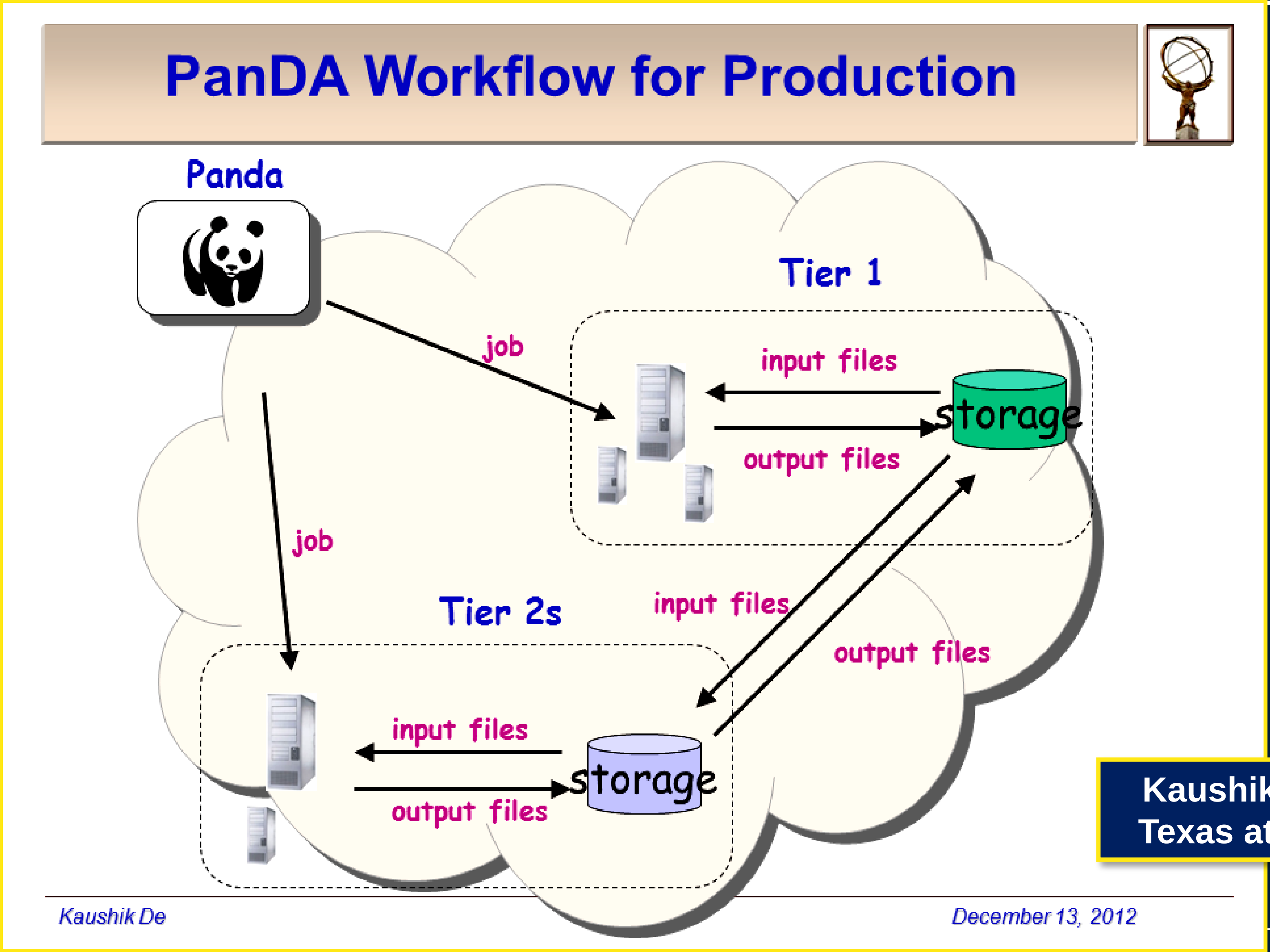
      (2) Raise alarm (e.g. extreme cases such as site isolation)
- *Network Control: Allows Pro-active Use*
  - Reserve Bandwidth Dynamically: prioritize transfers, remote access flows, etc.
  - Co-scheduling of CPU, Storage and Network resources
  - Create Custom Topologies ➡ optimize infrastructure to match operational conditions: deadlines, workprofiles
    - e.g. during LHC running periods vs reconstruction/re-distribution

ATLAS Computing: PanDA

- PanDA Workflow Management System (ATLAS): A Unified System for organized production and user analysis jobs
  - Highly automated; flexible; adaptive
  - Uses asynchronous Distributed Data Management system
  - Eminently scalable: to > 1M jobs/day (on many days)
- DQ2, Rucio: Register & catalog data; Transfer data to/from sites, delete when done; ensure data consistency; enforce ATLAS computing model
- PanDA basic unit of work: A Job
  - Physics tasks split into jobs by ProdSys layer above PanDA
- Automated brokerage based on CPU and Storage resources
  - Tasks brokered among ATLAS “clouds”
  - Jobs brokered among sites

➡ Here’s where Network information will be most useful!

ATLAS Production Workflow



PanDA Workflow for Production

Kaushik De

December 13, 2012

Kaushik De, Univ. Texas at Arlington

CMS Computing: PhEDEx

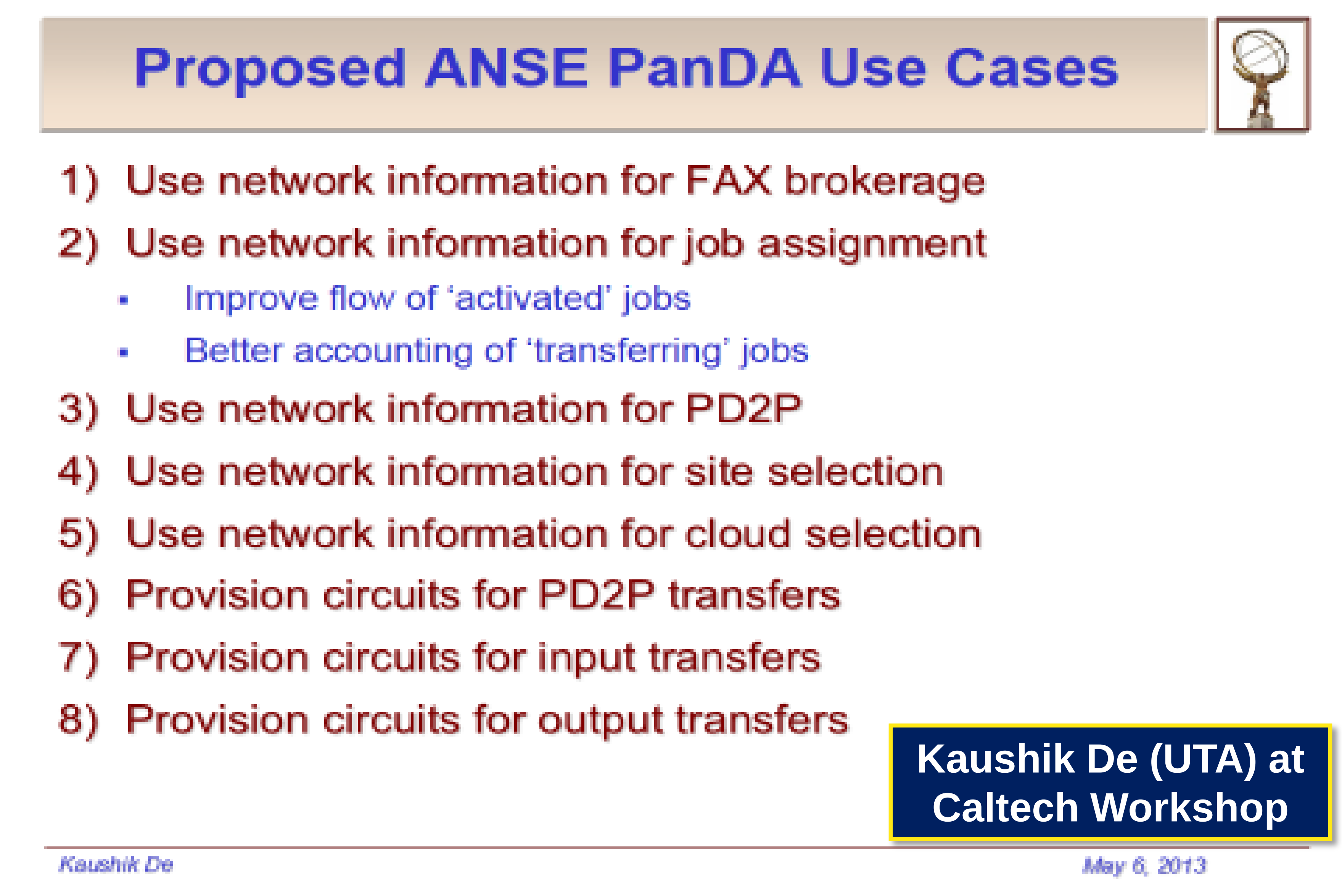
- PhEDEx is the CMS data-placement management tool
  - a reliable and scalable dataset (fileblock-level) replication system
  - With a focus on robustness
- Responsible for scheduling the transfer of CMS data across the grid
  - using FTS, SRM, FTM, or *any other transport package*
- PhEDEx typically queues data in blocks for a given src-dst pair
  - From tens of TB up to several PB
- Success metric so far is the volume of work performed; No explicit tracking of time to completion for specific workflows
  - Incorporation of the time domain, network awareness , and reactive actions, could increase efficiency for physicists
- Natural candidate for using dynamic circuits
  - could be extended to make use of a dynamic circuit API like NSI

CMS: Possible Approaches within PhEDEx

From T. Wildish, PhEDEx team lead

- There are essentially four possible approaches within PhEDEx for booking dynamic circuits:
  - Do nothing, and let the "fabric" take care of it
    - Similar to LambdaStation (by Fermilab + Caltech)
    - Trivial, but lacks prioritization scheme
    - Not clear the result will be optimal
  - Book a circuit for each transfer-job i.e. per FDT or gridftp call
    - Effectively executed below the PhEDEx level
    - Management and performance optimization not obvious
  - Book a circuit at each download agent, use it for multiple transfer jobs
    - Maintain stable circuit for all the transfers on a given src-dst pair
    - Only local optimization, no global view
  - Book circuits at the Dataset Level
    - Maintain a global view, global optimization is possible
    - Advance reservation

PanDA and ANSE



Proposed ANSE PanDA Use Cases

1) Use network information for FAX brokerage
2) Use network information for job assignment
   - Improve flow of 'activated' jobs
   - Better accounting of 'transferring' jobs
3) Use network information for PD2P
4) Use network information for site selection
5) Use network information for cloud selection
6) Provision circuits for PD2P transfers
7) Provision circuits for input transfers
8) Provision circuits for output transfers

Kaushik De

May 6, 2013

Kaushik De (UTA) at Caltech Workshop

FAX Integration with PanDA

- **ATLAS has developed detailed plans for integrating FAX with PanDA over the past year**
  - **Networking plays an important role in Federated storage**
  - **This time we are paying attention to networking up front**
  - **The most interesting use case – network information used for brokering of distributed analysis jobs to FAX enabled sites**
  - **This is first real use case for using external network information in PanDA**

Kaushik De (UTA)

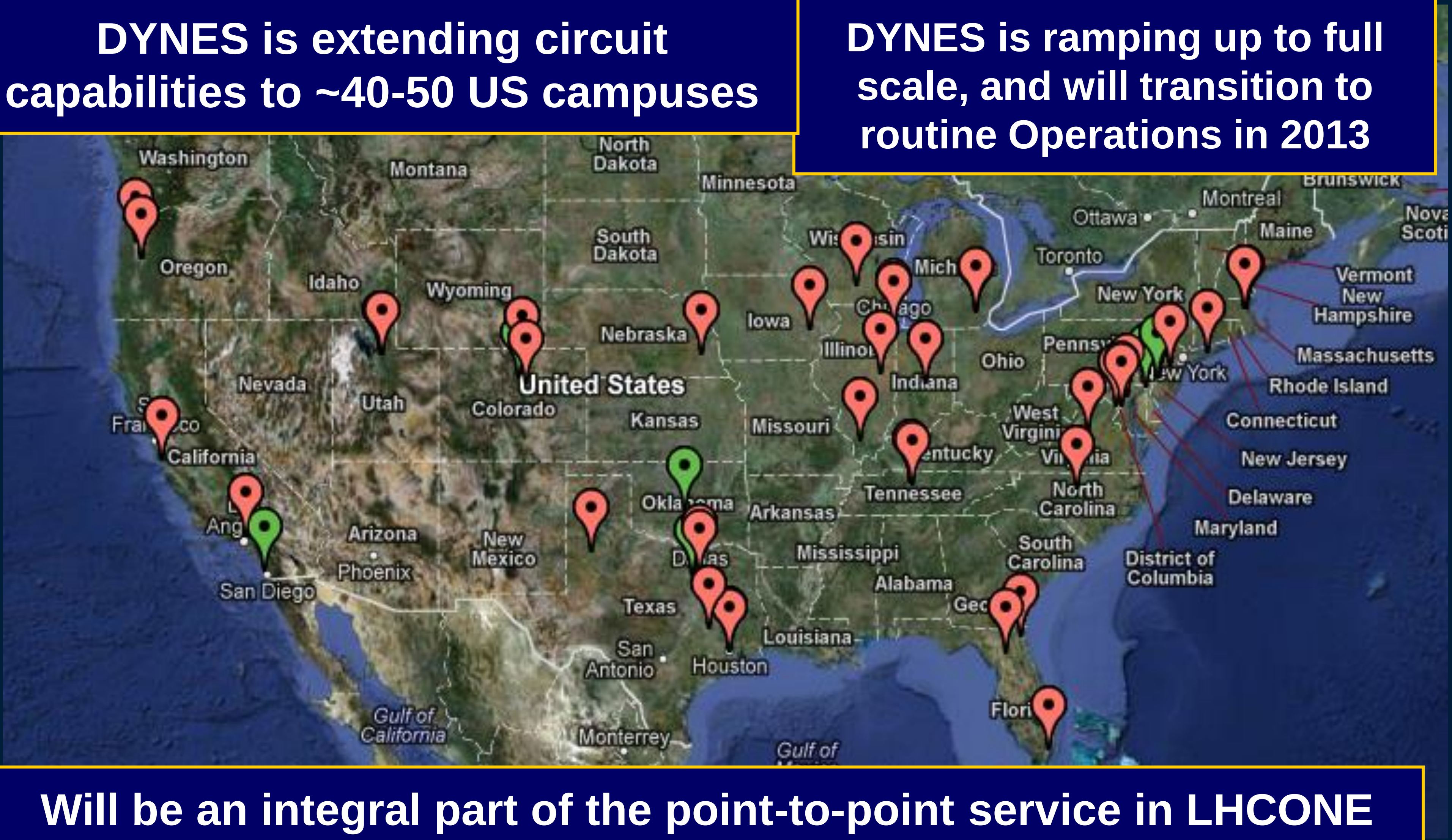
PD2P – How LHC Model Changed

- PD2P = PanDA Dynamic Data Placement
- PD2P is used to distribute data for user analysis
  - For production PanDA schedules all data flows
  - Initial ATLAS computing model assumed pre-placed data distribution for user analysis – PanDA sent jobs to data
  - Soon after LHC data started, we implemented PD2P
- Asynchronous usage based data placement
  - Repeated use of data → make additional copies
  - Backlog in processing → make additional copies
  - Rebrokerage of queued jobs → use new data location
  - Deletion service removes less used data
  - Basically, T1/T2 storage used as cache for user analysis
- This is perfect for network integration
  - Use network status information for site selection
  - Provisioning - usually large datasets are transferred, known volume

ANSE - Relation to DYNES

- In brief, DYNES is an NSF funded project to deploy a cyberinstrument linking up to 50 US campuses through Internet2 dynamic circuit backbone and regional networks
  - based on ION service, using OSCARS technology
- DYNES instrument can be viewed as a production-grade ‘starter-kit’
  - comes with a disk server, inter-domain controller (server) and FDT installation
  - FDT code includes OSCARS IDC API ➡ reserves bandwidth, and moves data through the created circuit
    - “Bandwidth on Demand”, i.e. get it now or never
    - routed GPN as fallback
- The DYNES system is naturally capable of advance reservation
- We need the right agent code inside CMS/ATLAS to call the API whenever transfers involve two DYNES sites

DYNES Sites



DYNES is extending circuit capabilities to ~40-50 US campuses

DYNES is ramping up to full scale, and will transition to routine Operations in 2013

Will be an integral part of the point-to-point service in LHCONE

ANSE Current Activities

- Candidate initial sites: UMich, UTA, Caltech, Vanderbilt, UVIC
- Monitoring information for workflow and transfer management
  - Define path characteristics to be provided to FAX and PhEDEx
  - On a NxN mesh of source/destination pairs
  - use information gathered in perfSONAR
  - Use LISA agents to gather end-system information
- Dynamic Circuit Systems
  - Working with DYNES at the outset
    - monitoring dashboard, full-mesh connection setup and BW test
  - Deploy a prototype PhEDEx instance for development and evaluation
    - integration with network services
  - Potentially use LISA agents for pro-active end-system configuration

ATLAS Next Steps

- UTA and Michigan are focusing on getting estimated bandwidth along specific paths available for PanDA (and PhEDEx) use.
  - Site A to SiteB; NxN mesh
- Michigan (Shawn McKee, Jorge Batista) is working with the perfSONAR-PS metrics which are gathered at OSG and WLCG sites.
  - The perfSONAR-PS toolkit instances provide a web services interface open for anyone to query, but we would rather use the centrally collected data that OSG and WLCG will be gathering.
  - The "new" modular dashboard project (https://github.com/PerfModDash) has a collector, a datastore and a GUI.
  - Batista is working on creating an estimated bandwidth matrix for WLCG sites querying the data in the new dashboard datastore.
  - Also extending the datastore to gather/filter traceroute from perfSONAR

Shawn McKee
Michigan

ATLAS Next Steps: University of Texas at Arlington (UTA)

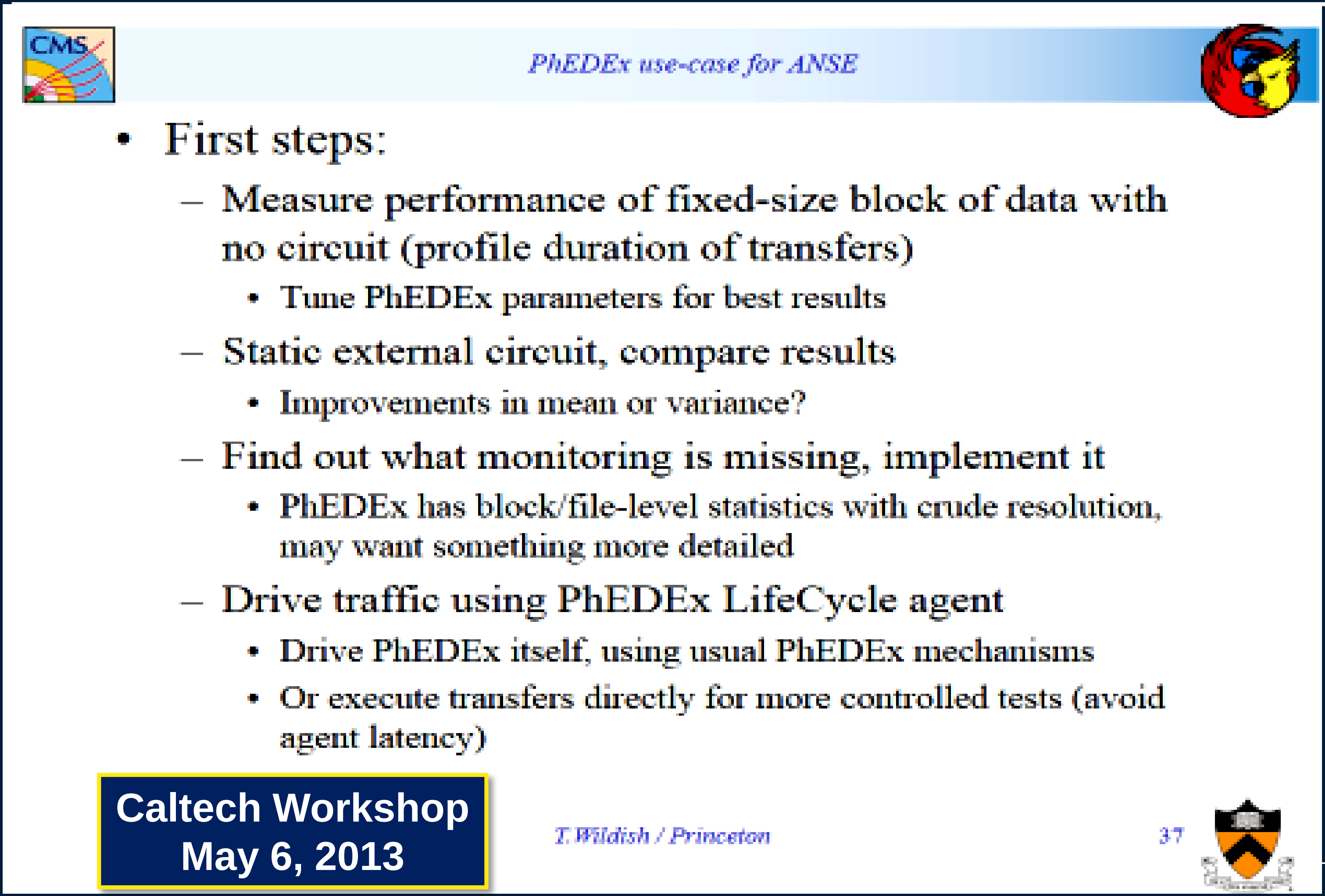
- **UTA is working on PanDA network integration.**
- **Artem Petrosyan, a new hire with experience in the PanDA team before coming to UTA is preparing a web page with the best sites in each cloud for PanDA.**
  - **He will collect data from existing ATLAS SONAR (data transfer) tests and load them into the AGIS (ATLAS information) system.**
  - **Needs to implement the schema for storing the data appropriately**
  - **Next he will create metadata for statistical analysis & prediction**
  - **He will then build a web UI to present the best destinations for job and data sources.**
- **In parallel with AGIS and web UI work, Artem will be working with PanDA to determine the best locations to access and use the information he gets from SONAR as well as the information Jorge gets from perfSONAR-PS.**
- **Also coordinated with the BigPanDA (OASCR) project at BNL (Yu)**

CMS PhEDEx Next Steps

- **Prototype installation of PhEDEx for ANSE/LHCONE**
  - **Use Testbed Oracle instance at CERN**
  - **Install website (need a box to put it on)**
  - **Install site agents at participating sites**
    - One box per site, (preferably) at the site
    - Possible to run site-agents remotely, but need remote access to SE to check if file transfers succeeded/delete old files
  - **Install management agents**
    - One box somewhere, anywhere – could even be CERN
  - **For a few sites, could all be run by 1 person**
  - **Will provide a fully-featured system for R&D**
    - Can add nodes as/when other sites want to participate

Tony Wildish
Princeton

PhEDEx & ANSE, Next Steps



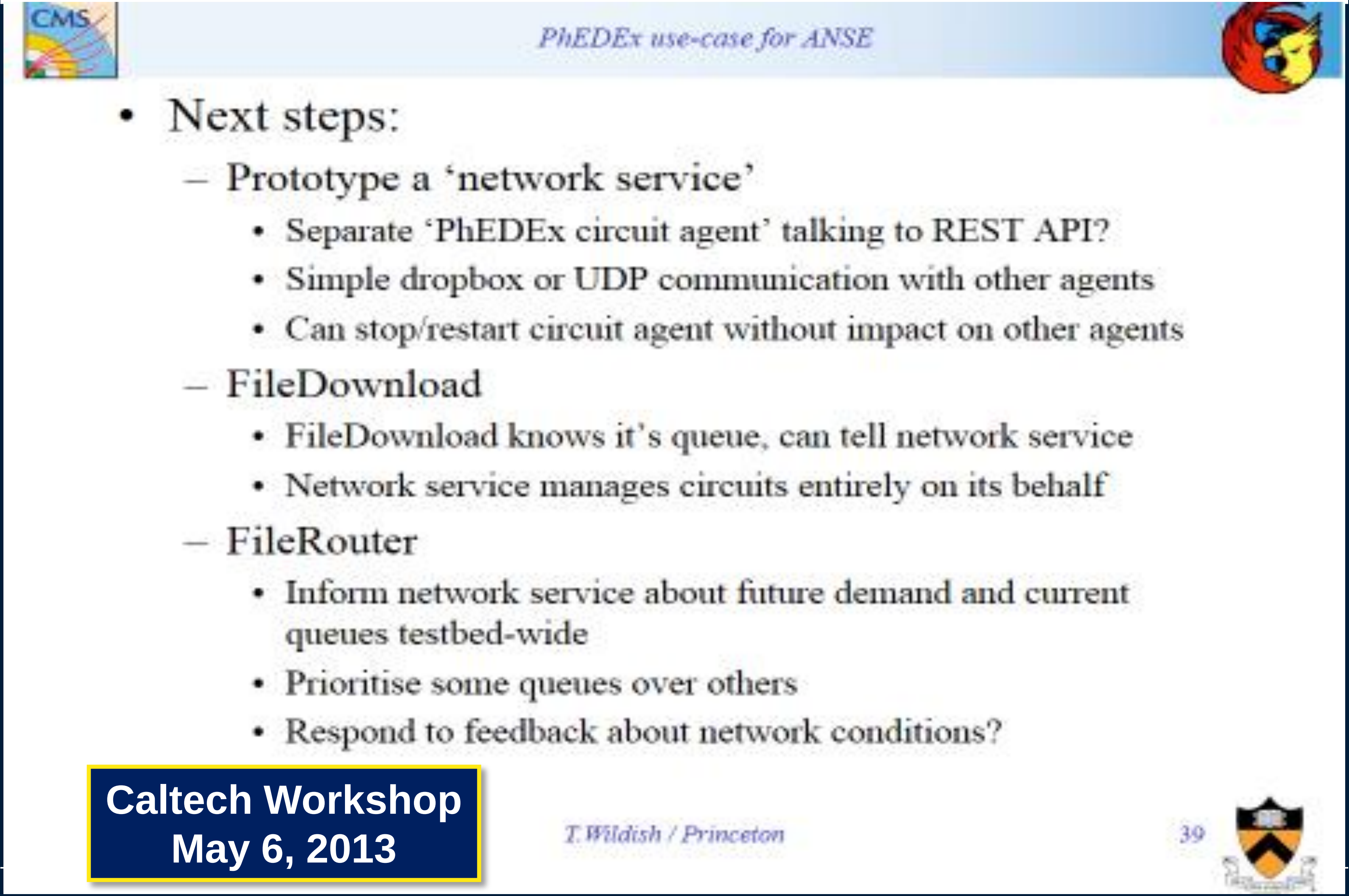
PhEDEx use-case for ANSE

- First steps:
  - – Measure performance of fixed-size block of data with no circuit (profile duration of transfers)
    - Tune PhEDEx parameters for best results
  - – Static external circuit, compare results
    - Improvements in mean or variance?
  - – Find out what monitoring is missing, implement it
    - PhEDEx has block/file-level statistics with crude resolution, may want something more detailed
  - – Drive traffic using PhEDEx LifeCycle agent
    - Drive PhEDEx itself, using usual PhEDEx mechanisms
    - Or execute transfers directly for more controlled tests (avoid agent latency)

T. Wildish / Princeton

Caltech Workshop
May 6, 2013

PhEDEx & ANSE, next steps



PhEDEx use-case for ANSE

- Next steps:
  - Prototype a 'network service'
    - Separate 'PhEDEx circuit agent' talking to REST API?
    - Simple dropbox or UDP communication with other agents
    - Can stop/restart circuit agent without impact on other agents
  - FileDownload
    - FileDownload knows it's queue, can tell network service
    - Network service manages circuits entirely on its behalf
  - FileRouter
    - Inform network service about future demand and current queues testbed-wide
    - Prioritise some queues over others
    - Respond to feedback about network conditions?

Caltech Workshop
May 6, 2013

ANSE: Summary and Conclusions

- The ANSE project will integrate advanced network services with the LHC Experiments' mainstream production SW stacks
- Through interfaces to
  - Monitoring services (PerfSONAR-based, MonALISA)
  - Bandwidth reservation systems (NSI, IDCP, DRAC, etc.)
- By implementing network-based task and job workflow decisions (cloud and site selection; job & data placement) in
  - PanDA system in ATLAS
  - PhEDEx in CMS
- The goal is to make deterministic workflows possible
  - Increasing the working efficiency of the LHC Discovery program
  - Profound implications for future networks and other fields
- A challenge and an opportunity for Software Defined Networks
  - With new scale and scope: optimization of global systems

THANK YOU !

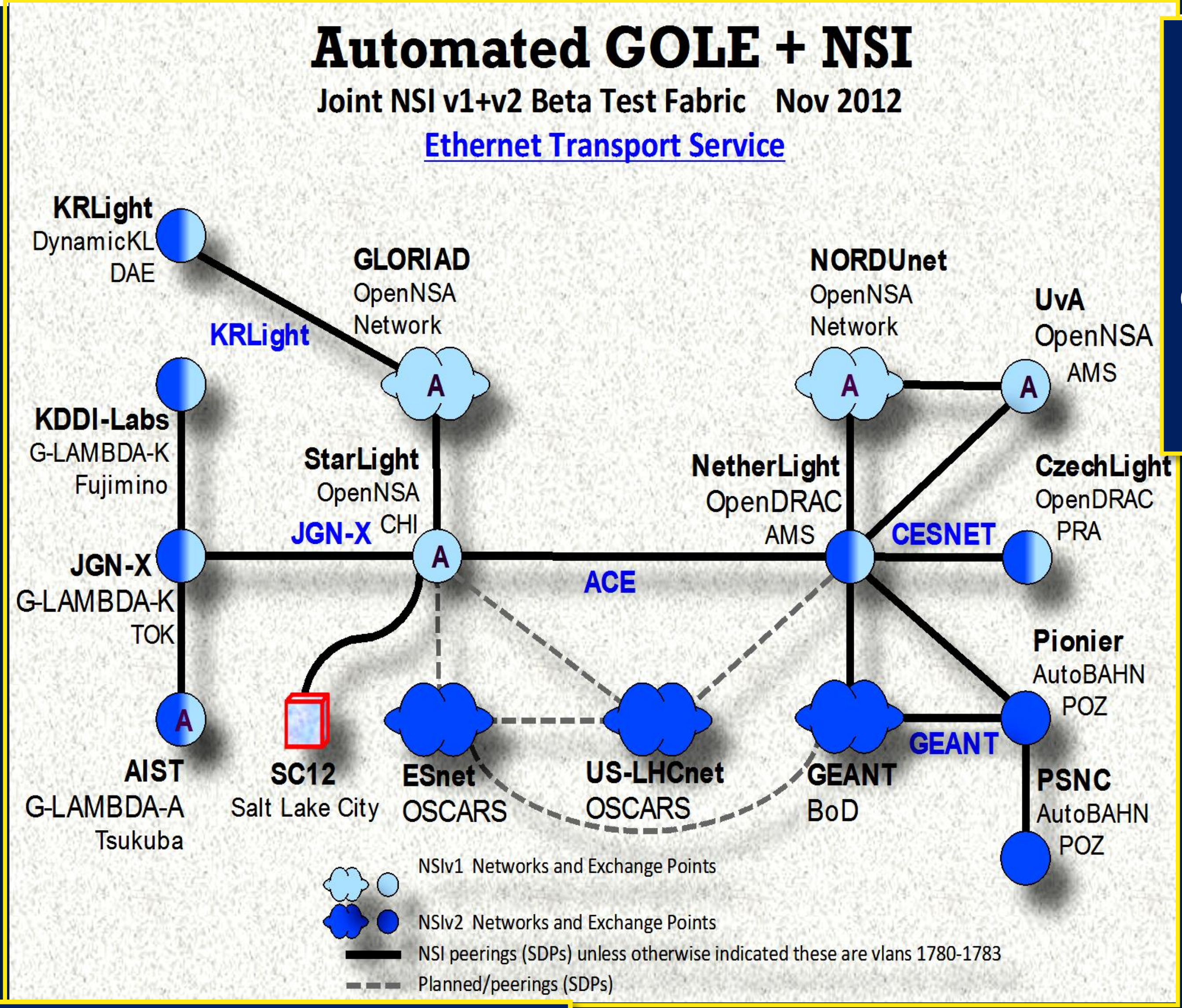
QUESTIONS ?

newman@hep.caltech.edu

BACKUP SLIDES FOLLOW

newman@hep.caltech.edu

Components for a Working System: Dynamic Circuit Infrastructure



Automated GOLE + NSI

Joint NSI v1+v2 Beta Test Fabric Nov 2012

Ethernet Transport Service

NSIv1 Networks and Exchange Points

NSIv2 Networks and Exchange Points

NSI peerings (SDPs) unless otherwise indicated these are vlans 1780-1783

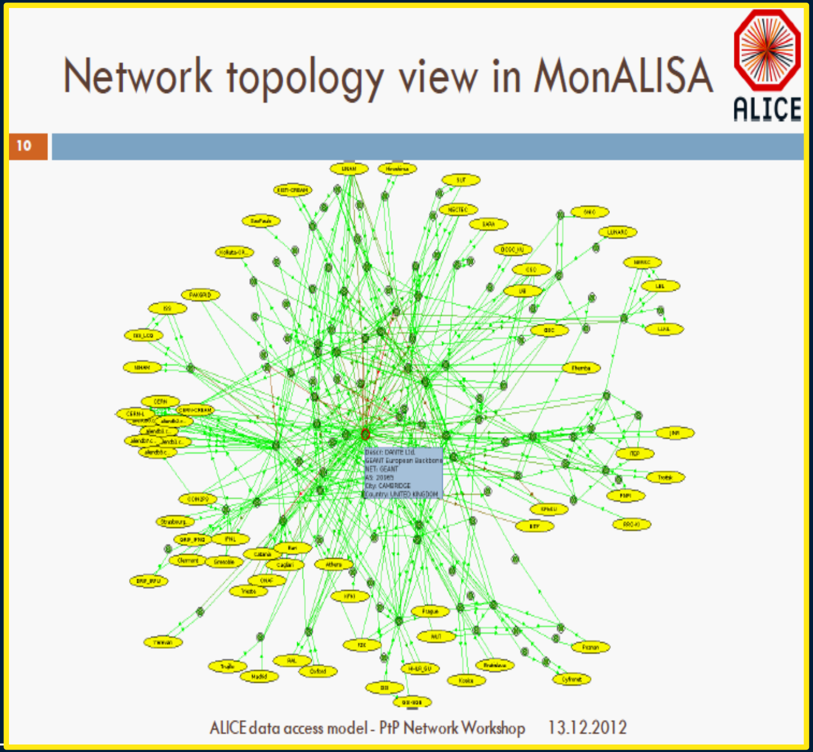
Planned/peerings (SDPs)

To be useful for the LHC and other communities,
it is essential to build on current & emerging standards, deployed on a global scale

Jerry Sobieski, NORDUnet

Components for a working system:
In-Depth Monitoring

- Monitoring: PerfSONAR and MonALISA
- All LHCOPN and many LHCONE sites have PerfSONAR deployed
  - Goal is to have all LHCONE instrumented for PerfSONAR measurement
- Regularly scheduled tests between configured pairs of end-points:
  - Latency (one way)
  - Bandwidth
- Currently used to construct a dashboard
- Could provide input to algorithms developed in ANSE for PhEDEx and PanDA
- ALICE and CMS experiments are using the MonALISA monitoring framework
  - Real time
  - Accurate bandwidth availability
  - Complete topology view
  - Higher level services



Network topology view in MonALISA

ALICE

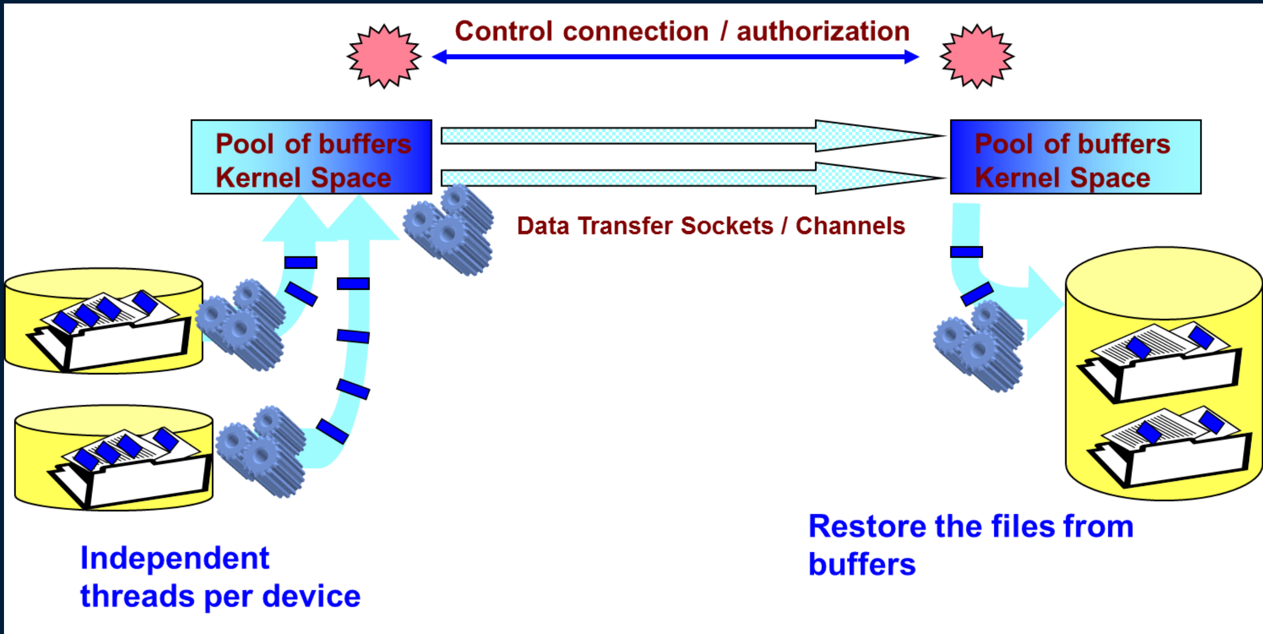
10

ALICE data access model - PtP Network Workshop 13.12.2012

Fast Data Transfer (FDT)

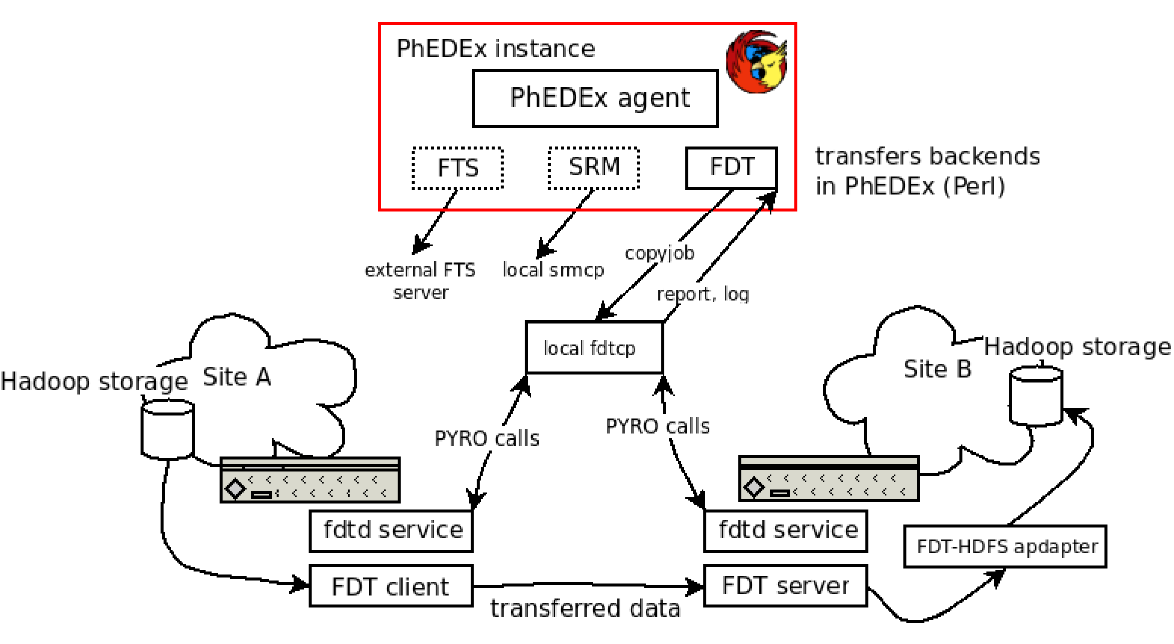
- DYNES instrument includes a storage element, FDT as transfer application
- FDT is an open source Java application for efficient data transfers
- Easy to use: similar syntax with SCP, iperf/netperf
- Based on an asynchronous, multithreaded system
- Uses the New I/O (NIO) interface and is able to:
  - stream continuously a list of files
  - use independent threads to read and write on each physical device
  - transfer data in parallel on multiple TCP streams, when necessary
  - use appropriate size of buffers for disk IO and networking
  - resume a file transfer session

FDT uses IDC API to request dynamic circuit connections


Control connection / authorization
Pool of buffers
Kernel Space
Pool of buffers
Kernel Space
Data Transfer Sockets / Channels
Independent threads per device
Restore the files from buffers

DYNES/FDT/PhEDEx

- FDT integrates OSCARS IDC API to reserve network capacity for data transfers
- FDT has been integrated with PhEDEx at the level of download agent
- Basic functionality OK
  - more work needed to understand performance issues with HDFS
- Interested sites are welcome to test
- With FDT deployed as part of DYNES, this makes one possible entry point for ANSE


PhEDEx instance
PhEDEx agent
FTS
SRM
FDT
transfers backends
in PhEDEx (Perl)
external FTS
server
local srmcp
copyjob
report, log
local fdtcp
Hadoop storage
Site A
Site B
Hadoop storage
PYRO calls
PYRO calls
fdtd service
fdtd service
FDT-HDFS apdapter
FDT client
FDT server
transferred data