Argonne NATIONAL LABORATORY

Concerted Flows:
Infrastructure for Terabit/s Data Transfer

Raj Kettimuthu, Eun-Sung Jung, Venkatram Vishwanath, Steve Tuecke, Mark Hereld, Mike Papka, Bob Grossman and Ian Foster

U.S. DEPARTMENT OF ENERGY

Exploding data volumes

Astronomy

MACHO et al.: 1 TB

Palomar: 3 TB

2MASS: 10 TB

GALEX: 30 TB

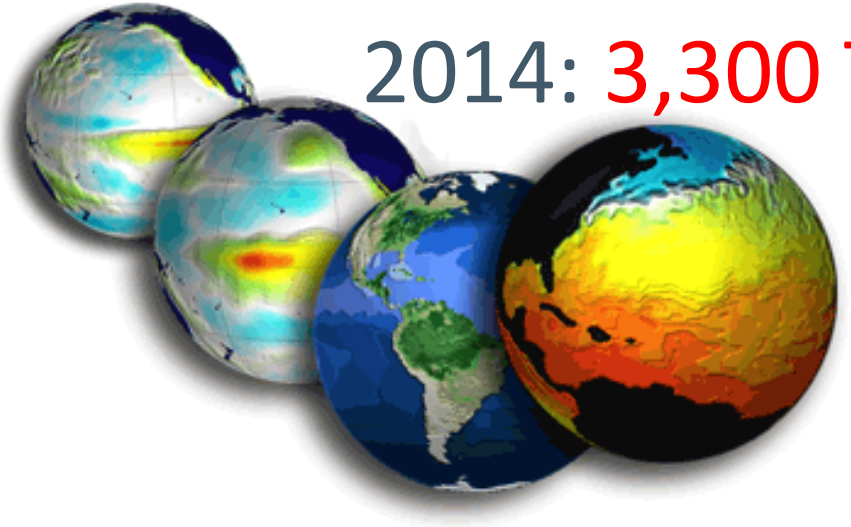
Sloan: 40 TB

Pan-STARRS: 40,000 TB

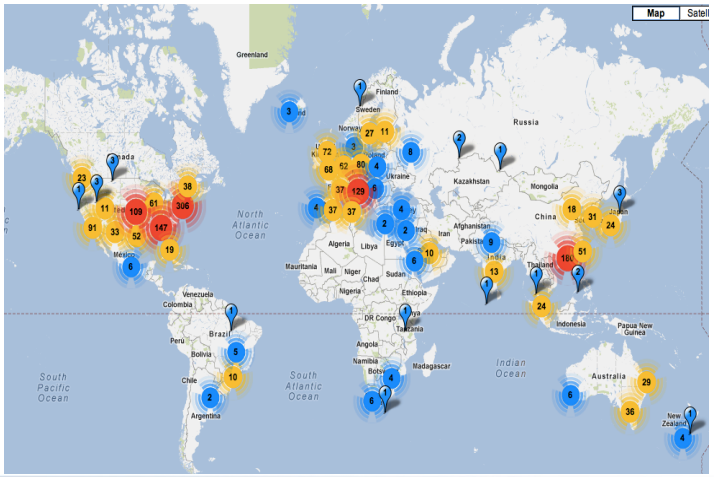
Climate

2004: 36 TB

2014: 3,300 TB

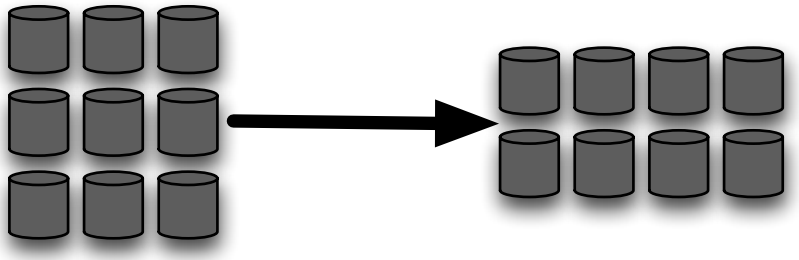


Genomics

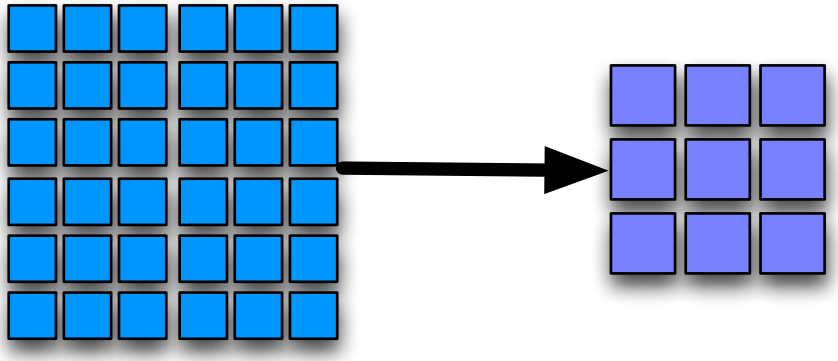


10^5 increase in data volumes in 6 years

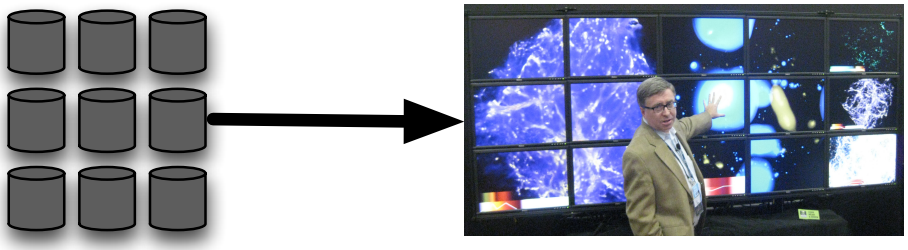
Data movement trends



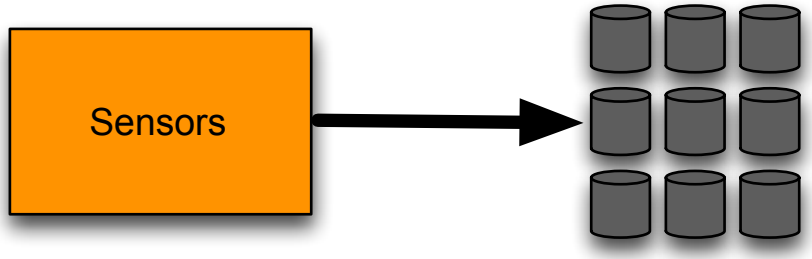
Disk-to-Disk Transfers



Memory-to-Memory Transfers



Disk-to-Memory Transfers



Memory-to-Disk Transfers

Data Movement is being increasingly characterized by Parallel M-to-N Data Flows

Characteristics of application flows

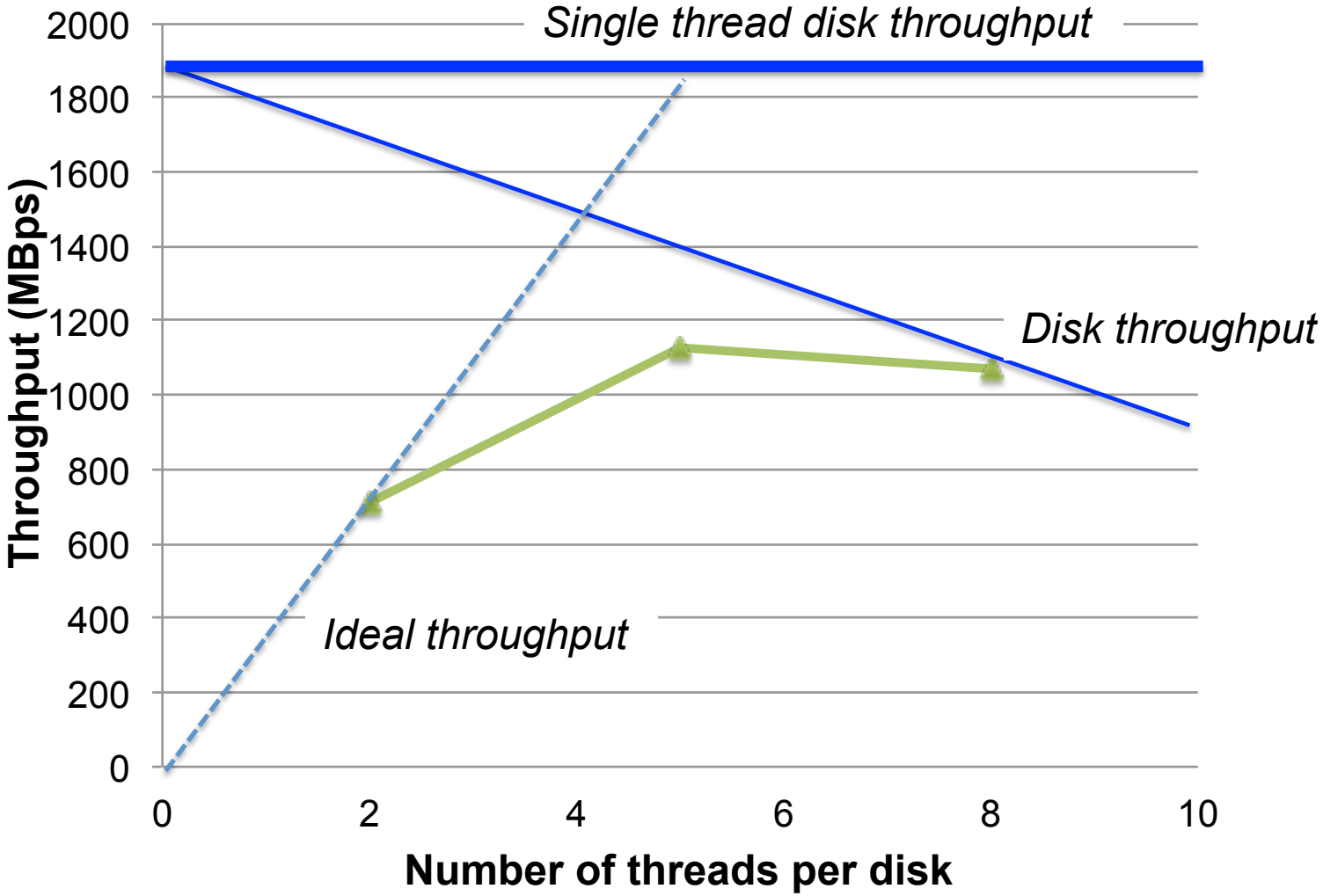
App	Type of Flow	# of Flows	BW	Latency	Burstiness	Size	Protocol
Globus Online	Data	1 per node	High	N	Y	Large	TCP, UDT
	Control	1 per session	Low	Y	Y	Small	TCP
APS	Data	1 per detector	High	N	Y	Large	TCP
	Control	1 per app	Low	Y	Y	Small	TCP
FLASH Simulation-time Analysis	Data	1 per core	High	N*	Y	Variable	TCP, RDMA
	Control	1 per app	Low	Y	y	Small	TCP, RDMA
ENZO Remote Viz	Data	1 per display	High	Y	N	Large	TCP, UDP
	Control	1 per app	Low	Y	Y	Small	TCP

A mechanism to characterize and model an application's data movement behavior will be critical to better architect future networks

Concerted flows

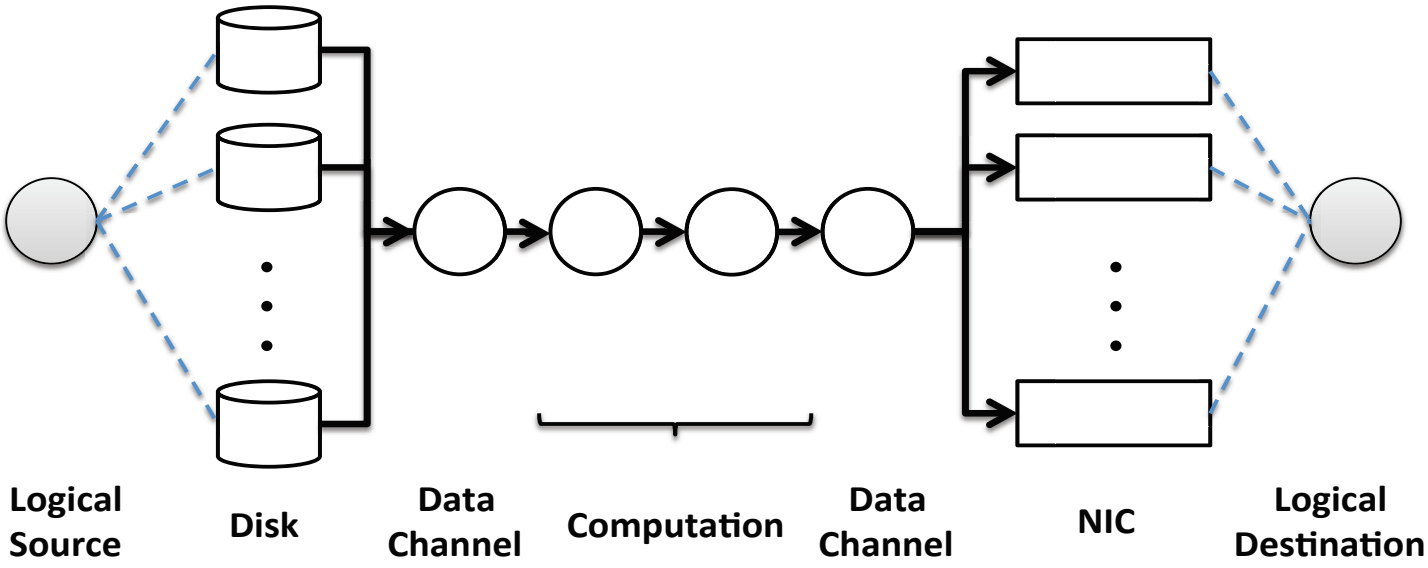
- **Problem**: Traditional data transfer mechanisms fail to scale, fail to satisfy diverse applications needs
- **Goal**: Develop new tools that are
  - **Adaptive:** Leverages the characteristics of various components in the end-to-end path, feedback from network agents etc. to optimize transfers
  - **Composable:** Captures the diverse flow characteristics and requirements of applications
- **Results**: Tools that optimize individual transfers, efficient scheduling of large number of transfer requests
  - Model based approach for component-specific optimization
  - Data transfer kernels to capture transfer patterns of applications
  - Demonstrated near real-time data movement with 2 applications

Component-specific optimization



Disk-to-disk data transfer: graph model

- The system is modeled as a directed data flow graph:
  - A ***node*** indicates a physical system entity or a software entity.
  - A ***edge*** indicates a connectivity between two entities in terms of data flow.
  - Two attributes are assigned on an edge.
    - **Capacity/Bandwidth**: The maximum amount of data flow on the edge
    - **Cost**: CPU cost

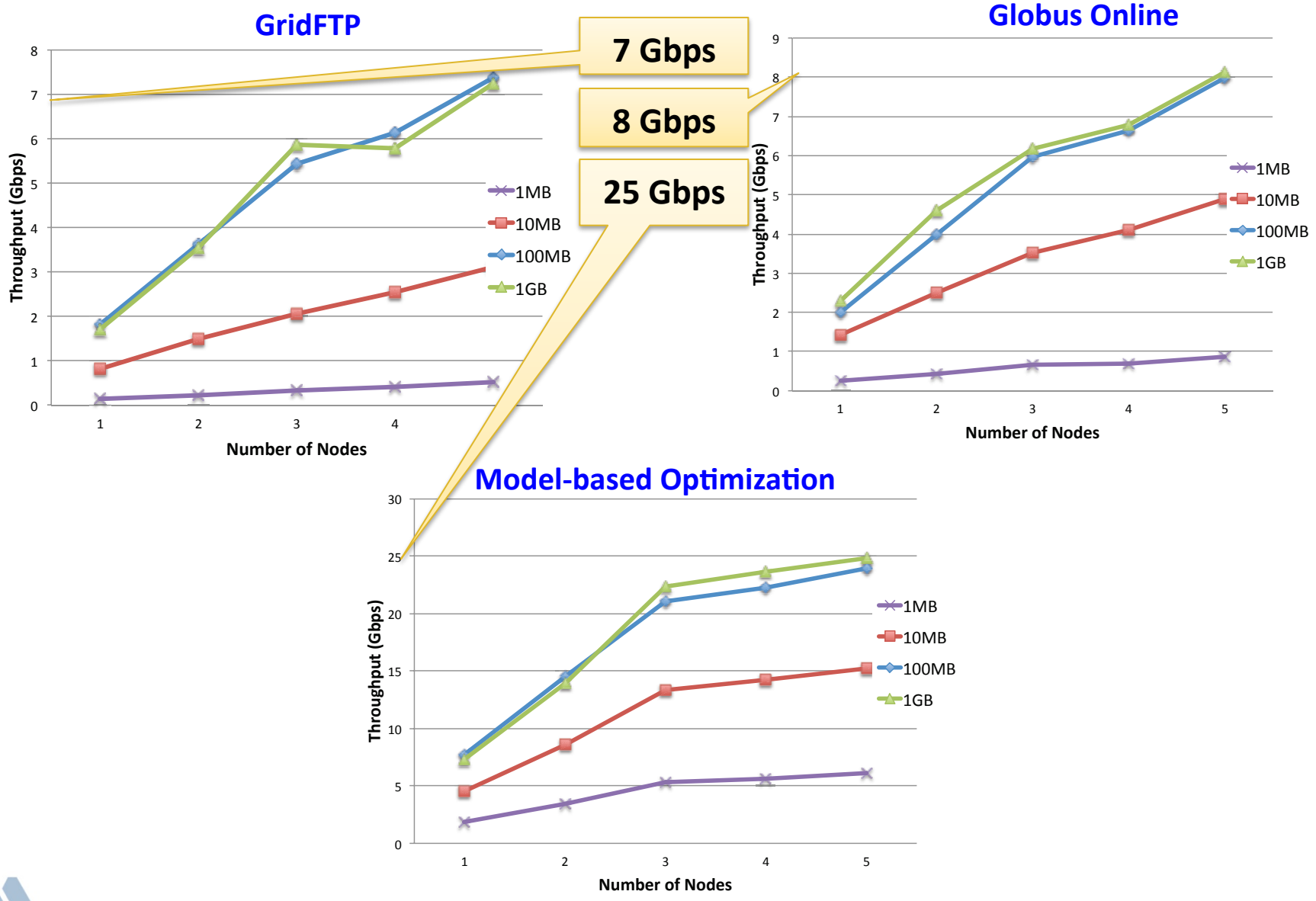


Model accuracy with and without perfSONAR data

Node	Training Error (%)	Validation Error (%)
Gordon	14.4	13.4
Mason	14.1	13.6
NCAR	10.9	14.7
Blacklight	14.2	13.8
Kraken	13.4	14.7

Node	Training Error (%)	Validation Error (%)
Gordon	11.2	10.6
Mason	10.8	11.6
NCAR	9.7	12.2
Blacklight	11.2	10.8
Kraken	10.4	11.3

Evaluation


GridFTP
Throughput (Gbps)
Number of Nodes
1MB
10MB
100MB
1GB
7 Gbps
8 Gbps
25 Gbps
Globus Online
Throughput (Gbps)
Number of Nodes
1MB
10MB
100MB
1GB
Model-based Optimization
Throughput (Gbps)
Number of Nodes
1MB
10MB
100MB
1GB

Data transfer skeletons

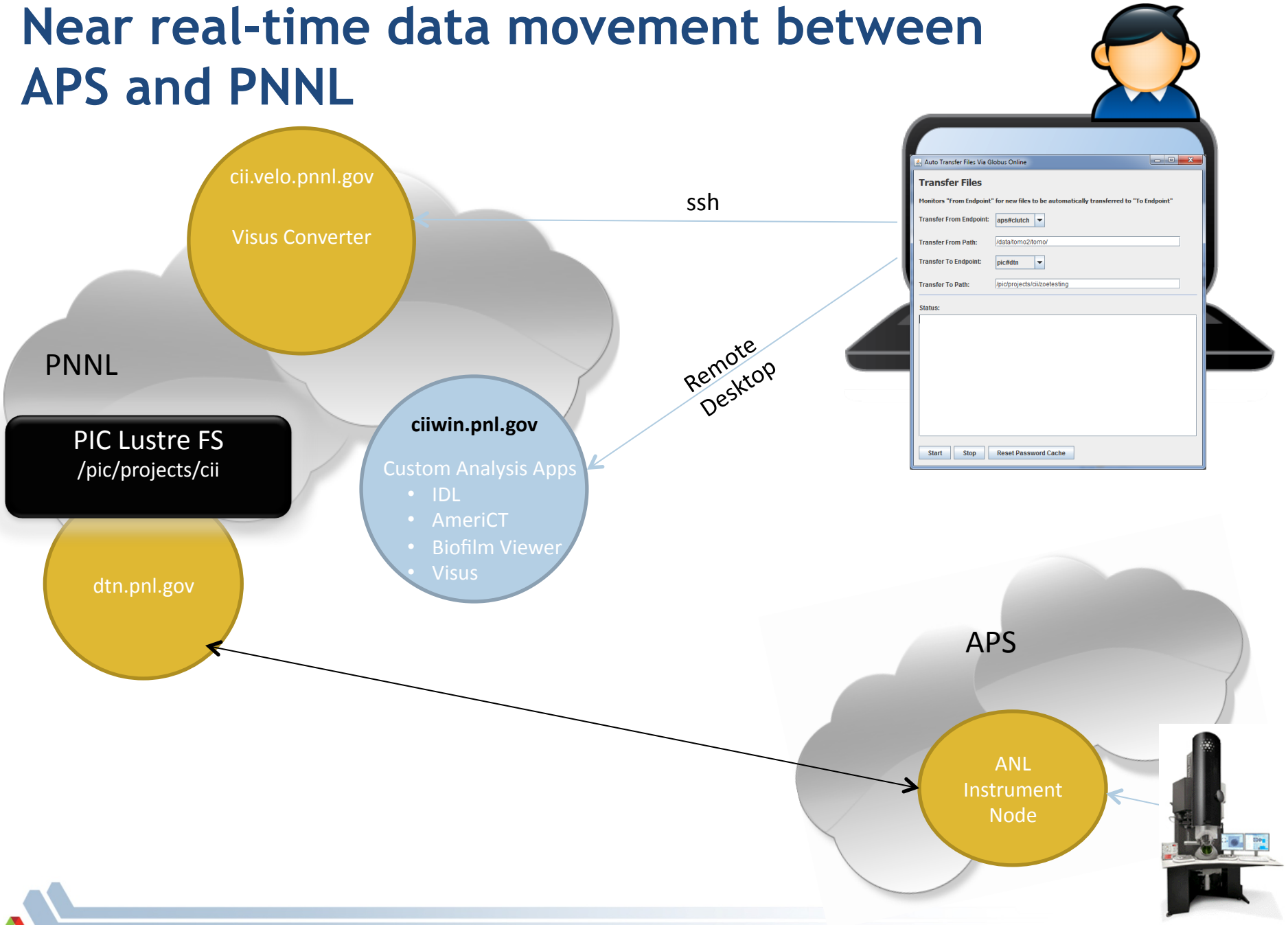
```
1.  :InFile = 100 // in MB
2.  ::InFileRate = 100 // in MB/s
3.  :OutFile = 10240 // in MB
4.  :N = 50
5.  def main()
6.  {
7.    :InFile ::InFileRate infile[N]
8.    :OutFile outfile
9.    call data_gen(infile)
10.   Transfer infile to A
11.   forall g = 0:N
12.   {
13.     call reconstruct(infile, outfile)
14.    }
15.    Transfer outfile to B
16.    call analysis(outfile)
17. }
```

Each file is 100MB and it generates 50 files at 100 MB/s rate.

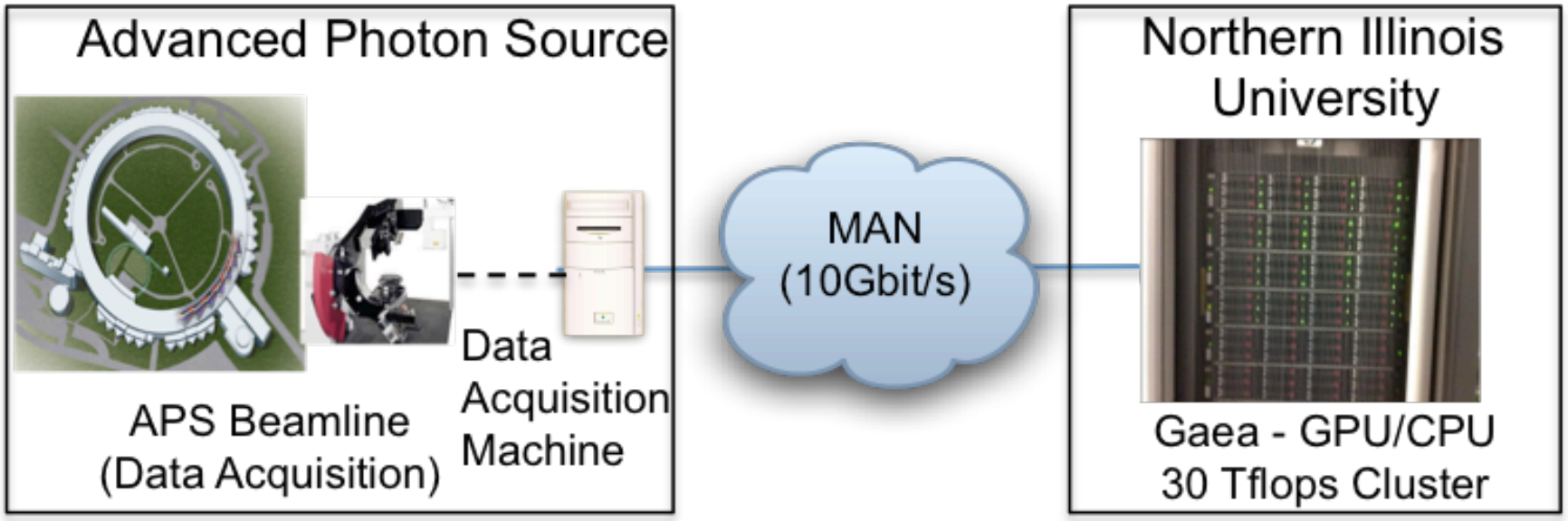
Reconstruction is done with 50x100MB files, and 10GB file is generated.

Analysis is done on the 10GB file.

Near real-time data movement between APS and PNNL



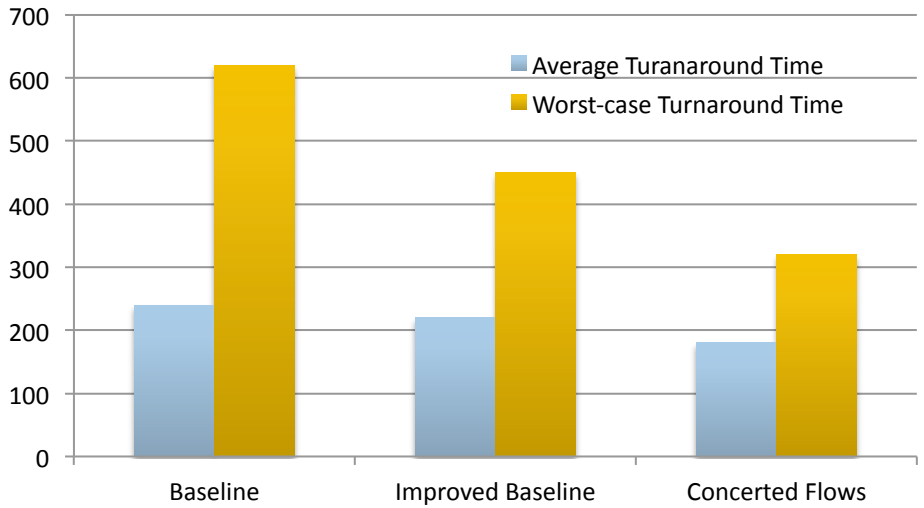
Near real-time data movement between APS and NIU

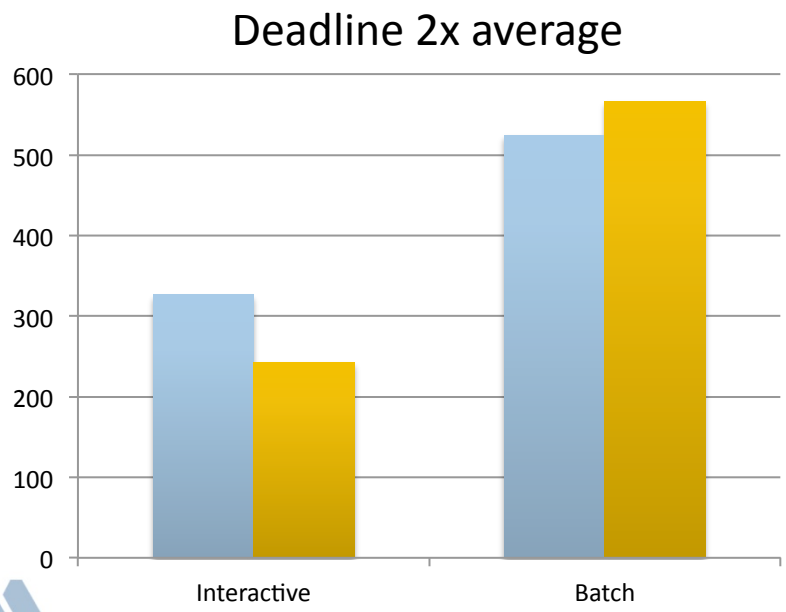


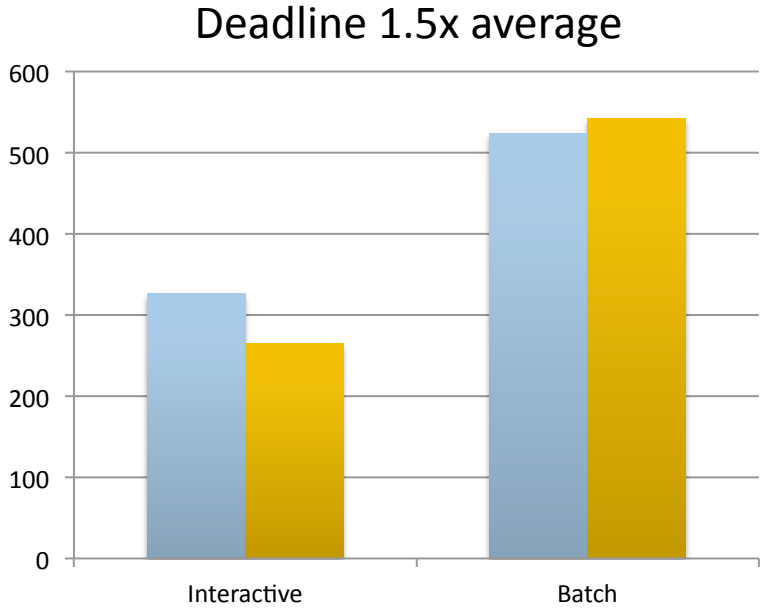
Scheduling transfers

- Maximize resource utilization and reduce slowdown
  - Adaptively queue and adjust concurrency
  - Use both models and recent observed behavior
- Transfers have different requirements and constraints
  - Time constraints - near real time to highly flexible
  - Loss tolerance, rate requirements
- Objective – account requirements to improve overall user experience
- Consider 2 job types – batch and interactive
  - Exploit relaxed deadlines of batch jobs

Results


700
600
500
400
300
200
100
0
Average Turanaround Time
Worst-case Turnaround Time
Baseline
Improved Baseline
Concerted Flows


Deadline 2x average
600
500
400
300
200
100
0
Interactive
Batch


Deadline 1.5x average
600
500
400
300
200
100
0
Interactive
Batch

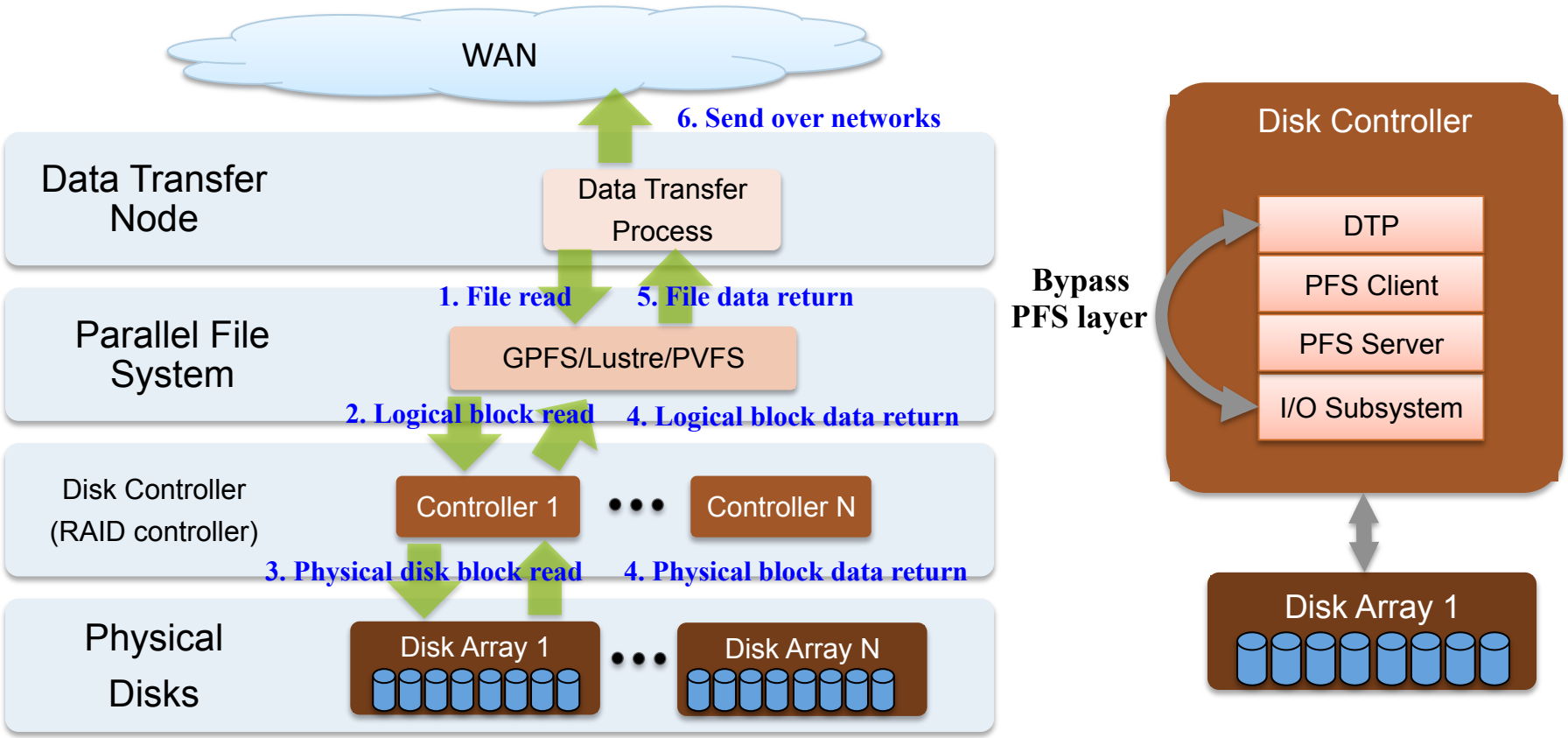
Publications

- E. Jung, R. Kettimuthu, and V. Vishwanath, “Cluster-wise Disk-to-Disk Transfer with Data Compression over Wide-Area Networks,” Special Issue of JPDC, 2014.
- Eun-Sung Jung, and Rajkumar Kettimuthu, "Data-intensive Computing on the Cloud: State-of-the-art, Challenges, and Opportunities", accepted to IEEE Computer (SCI), 2014.
- K. Maheshwari, E. Jung, J. Meng, V. Vishwanath, and R. Kettimuthu, “Improving Multisite Workflow Performance Using Model-based Scheduling,” ICPP’14, Sep‘14.
- R. Kettimuthu, G. Vardoyan, G. Agrawal and P. Sadayappan, “Modeling and Optimizing Large-Scale Wide-Area Data Transfers,” CCGrid2014, May 2014.
- E. Jung, R. Kettimuthu and V. Vishwanath, “Toward optimizing disk-to-disk transfer on 100G networks,” IEEE ANTS 2013, Dec. 2013.
- E. Jung, K. Maheshwari and R. Kettimuthu, “Pipelining/Overlapping Data Transfer for Distributed Data-Intensive Job Execution,” 2013 ICPP Workshops Oct. 2013.
- K. Maheshwari, E. Jung, J. Meng, V. Vishwanath and R. Kettimuthu, “Model-Driven Multisite Workflow Scheduling Based on Task-Resource Adaptation, IEEE Cluster 2013, Sep. 2013.
- D. Gunter, R. Kettimuthu, E. Kissel, M. Swany, J. Yi, J. Zurawski, “Exploiting Network Parallelism for Improving Data Transfer Performance,” IEEE/ACM Annual SuperComputing Conference (SC12) Companion Volume, Nov. 2012.

Publications

- J. Yi, R. Kettimuthu, V. Vishwanath, “Accelerating Data Movement Leveraging Endsystem and Network Parallelism,” IEEE/ACM SC12 Network-Aware Data Management Workshop, Nov. 2012.
- J. Yi, R. Kettimuthu, and V. Vishwanath, “Toward Characterization of Data Movement in Large-Scale Scientific Applications,” 8th IEEE eScience, Oct. 2012.
- E. Jung and R. Kettimuthu, "High-Performance Serverless Data Transfer over Wide-Area Networks", submitted to NDM workshop to be held in conjunction with SC'14 (under review).
- E. Jung, R. Kettimuthu, and V. Vishwanath, "Distributed Multipath Routing Algorithms for Data Center Networks", submitted to DISCS workshop to be held in conjunction with SC'14 (under review).
- E. Jung, and R. Kettimuthu, "An overview of Parallelism Exploitation and Cross-layer Optimization for Big Data Transfer", in preparation for submission to journal.
- K. Maheshwari, E. Jung, J. Meng, V. Vishwanath, and R. Kettimuthu, "Improving Multisite Workflow Performance using Model-based Scheduling", in preparation for submission to FUTURE GENERATION COMPUTER SYSTEMS (SCIE).
- R. Kettimuthu, G. Vardoyan, G. Agrawal, P. Sadayappan, and I. Foster, “Adaptive Scheduling of Large-Scale Wide-Area Data Transfers”, in preparation to IEEE International Parallel and Distributed Processing Symposium (IPDPS), May 2015.

Serverless Data Transfer



How do we allocate data transfer bandwidth to users?

- User specify requirements for transfers
  - Might always want maximum available resources
- Network backbone may not be the bottleneck
  - End-to-end bandwidth is limited
- What measure do we use?
- How do we allocate?
- How do we enforce?
- Distributed resources – network – multiple domains, multiple end systems

Questions?