Cloud-hosted Data Transfer & Optimization:
Stork for the Cloud

Tevfik Kosar
University at Buffalo (SUNY)

May 2, 2012
Condor Week, Madison, WI
Big Data

- 1 PB is now considered “small” for many science applications today
- For most, their data is distributed across several sites

A survey among 106 organizations operating two or more data centers:
- 77% run replication among three or more sites
- 50% has more than 1 PB in their primary data center
- 1 PB is now considered “small” for many science applications today
- For most, their data is distributed across several sites

A survey among 106 organizations operating two or more data centers:
- 77% run replication among three or more sites
- 50% has more than 1 PB in their primary data center
Best Way to Move Big Data?
Best Way to Move Big Data?

• Sending 1 PB of data over 10 Gbps link would take nine days (assuming 100% efficiency) -- too optimistic!
Best Way to Move Big Data?

• Sending 1 PB of data over 10 Gbps link would take **nine days** (assuming 100% efficiency) -- too optimistic!

• Sending 1 TB Forensics dataset from Boston to Amazon S3 cost $100 and took **several weeks** [Garfinkel 2007]
Best Way to Move Big Data?

- Sending 1 PB of data over 10 Gbps link would take **nine days** (assuming 100% efficiency) -- too optimistic!
- Sending 1 TB Forensics dataset from Boston to Amazon S3 cost $100 and took **several weeks** [Garfinkel 2007]
- Visualization scientists at LANL dumping data to tapes and sending them to Sandia Lab via **Fedex** [Feng 2003]
Best Way to Move Big Data?

• Sending 1 PB of data over 10 Gbps link would take **nine days** (assuming 100% efficiency) -- too optimistic!
• Sending 1 TB Forensics dataset from Boston to Amazon S3 cost $100 and took **several weeks** [Garfinkel 2007]
• Visualization scientists at LANL dumping data to tapes and sending them to Sandia Lab via Fedex [Feng 2003]
• Collaborators have the option of moving their data into disks, and sending them as packages through UPS or FedEx [Cho et al 2011].
Best Way to Move Big Data?

• Sending 1 PB of data over 10 Gbps link would take nine days (assuming 100% efficiency) -- too optimistic!
• Sending 1 TB Forensics dataset from Boston to Amazon S3 cost $100 and took several weeks [Garfinkel 2007]
• Visualization scientists at LANL dumping data to tapes and sending them to Sandia Lab via Fedex [Feng 2003]
• Collaborators have the option of moving their data into disks, and sending them as packages through UPS or FedEx [Cho et al 2011].
• Will 100 Gbps networks change anything?
Stork Data Scheduler

- Implements state-of-the art models and algorithms for data scheduling & optimization
- Started as part of the Condor Project (was my PhD work)
- Currently developed at University at Buffalo and funded by NSF (CAREER, STCI, CiC)
- Based on the Condor code, uses Condor libraries (DaemonCore, ClassAds)
- Compatible with Condor products (i.e. DAGMan)

.....
Stork Data Scheduler

• ..... 
• Built & tested on Condor NMI (Metronome)
• Supports more than 20 platforms
• Futures include:
  • support for multiple transfer protocols
  • dynamic protocol tuning & optimization
  • end-to-end throughput prediction services
  • data aggregation & connection caching
  • early error detection and classification & recovery
End-to-end Problem

- Data flow
- Control flow

**Data flow**

**Control flow**

**Network Throughput**

**Memory-to-network Throughput on source**

**Disk-to-memory Throughput on source**

**Network-to-memory Throughput on Destination**

**Memory-to-disk Throughput on destination**
End-to-end Problem

protocol tuning
End-to-end Problem

**Data flow**

- **Control flow**

**protocol tuning**

**disk I/O optimization**

- Network -> Memory Throughput
- Mem->network -> Memory-to-network Throughput on source
- Disk->mem -> Disk-to-memory Throughput on source
- Mem->disk -> Memory-to-disk Throughput on destination

---

**Wednesday, May 2, 12**
End-to-end Problem

protocol tuning

disk I/O optimization

CPU optimization

Data flow
Control flow

Tnetwork :

TSmem->network

Tnetwork -> Network Throughput

TDnetwork->mem

Tnetwork -> Network-to-memory Throughput on destination

TSdisk->mem

TSdisk->mem -> Disk-to-memory Throughput on source

Tmem->disk

Tmem->disk -> Memory-to-disk Throughput on destination

Smem->network

Sdisk->mem

Smem->network -> Memory-to-network Throughput on source

Sdisk->mem -> Disk-to-memory Throughput on source

Smem->network -> Memory-to-network

Tmem->disk

Dnetwork->mem

Dnetwork->mem -> Network-to-memory Throughput on destination

Dmem->disk

Dmem->disk -> Memory-to-disk Throughput on destination
End-to-end Problem

Parameters to be optimized:
- # of streams
- # of disk stripes
- # of CPUs/nodes

protocol tuning

disk I/O optimization

CPU optimization
End-to-end Optimization

- CPU nodes are considered as nodes of a maximum flow problem
- Memory-to-memory transfers are simulated with dummy source and sink nodes
- The capacities of disk and network is found by applying parallel stream model by taking into consideration of resource capacities (NIC & CPU)
Challenging Problem

Optimize:
- concurrency
- parallelism
- pipelining
- conn. caching
- buffer size
- block size
- disk striping
- threading
- ....
Challenging Problem

Optimize:

- concurrency
- parallelism
- pipelining
- conn. caching
- buffer size
- block size
- disk striping
- threading
- ....

512 x 8 MB files
Challenging Problem

Optimize:

- concurrency
- parallelism
- pipelining
- conn. caching
- buffer size
- block size
- disk striping
- threading
- ....

512 x 8 MB files
Challenging Problem

Optimize:

- concurrency
- parallelism
- pipelining
- conn. caching
- buffer size
- block size
- disk striping
- threading
- ....

512 x 8 MB files
Challenging Problem

Optimize:
- concurrency
- parallelism
- pipelining
- conn. caching
- buffer size
- block size
- disk striping
- threading
- ....

512 x 1 MB files
Challenging Problem

Optimize:

- concurrency
- parallelism
- pipelining
- conn. caching
- buffer size
- block size
- disk striping
- threading
- ....

512 x 1 MB files
Challenging Problem

Optimize:

- concurrency
- parallelism
- pipelining
- conn. caching
- buffer size
- block size
- disk striping
- threading
- ....

512 x 32 MB files
Challenging Problem

Optimize:

- concurrency
- parallelism
- pipelining
- conn. caching
- buffer size
- block size
- disk striping
- threading
- ....

512 x 32 MB files
Kosar et al Models

Exponential Packet Loss

Break Function Modeling

Modeling Based on Newton’s Iteration

Modeling Based on Full Second Order

\[ p_n' = a'n^c + b' \]

\[ p_n = p_n \frac{RTT_n^2}{c^2MSS^2} = a'n^2 + b'n + c' \]
Kosar et al Models

- Details in 2 TPDS 2011 papers
- Implemented in the latest version of Stork (v.2.0.1)
- Provides throughput optimization as well as estimation

Modeling Based on Newton’s Iteration

\[ p'_n = a' n^c' + b' \]

Modeling Based on Full Second Order

\[ p'_n = p_n \frac{RTT_n^2}{c^2 MSS^2} = a' n^2 + b' n + c' \]
Stork for the Cloud

Storage Servers → iRODS server → SRM server → Thin Clients

Data Movement

Control/query Data/metadata

Stork for the Cloud

iRODS GridFTP SRM
Pluggable Protocol Interface
TMS Scheduler DLS
REST API

Status info

Request

Tablets Smartphones Web Interface
Stork Android Client
Stork Android Client

Stork Client

Connect to a server

Connect to a server
Stork Android Client

Stork Client

Connect to a server

Connect to a server
Stork Android Client

tg-login.spur.tacc.teragrid.org

earslan

Login
Stork Android Client

tg-login.spur.tacc.teragrid.org

earslan

gsiftp

Login
Stork Android Client

tg-login.spur.tacc.teragrid.org

Connect to a server
Stork Android Client

tg-login.spur.tacc.teragrid.org

Connect to a server
Stork Android Client

login1.ls4.tacc.utexas.edu

earslan

sftp

Login
Stork Android Client

tg-login.spur.tacc.teragrid.org

login1.ls4.tacc.utexas.edu
Stork Android Client

tg-login.spur.tacc.teragrid.org

login1.ls4.tacc.utexas.edu
Stork Android Client

tg-login.spur.tacc.teragrid.org
root
.globus
.lmod.d
.ssh
.copy

login1.ls4.tacc.utexas.edu
root
.globus
.lmod.d
.ssh
.externals
<table>
<thead>
<tr>
<th>Directory Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tg-login.spur.tacc.teragrid.org</td>
<td>Stork Client</td>
</tr>
<tr>
<td>.globus</td>
<td></td>
</tr>
<tr>
<td>.lmod.d</td>
<td></td>
</tr>
<tr>
<td>.ssh</td>
<td></td>
</tr>
<tr>
<td>copy</td>
<td></td>
</tr>
<tr>
<td>login1.ls4.tacc.utexas.edu</td>
<td></td>
</tr>
<tr>
<td>.globus</td>
<td></td>
</tr>
<tr>
<td>.lmod.d</td>
<td></td>
</tr>
<tr>
<td>.ssh</td>
<td></td>
</tr>
<tr>
<td>externals</td>
<td></td>
</tr>
</tbody>
</table>
Stork Android Client

tg-login.spur.tacc.teragrid.org
   globus
   gt5.0.4-all-source-installer
   stork-2.0.1
   .Xauthority
   .cshrc

login1.ls4.tacc.utexas.edu
   root
   .globus
   .lmod.d
   .ssh
   externals
Stork Android Client

tg-login.spur.tacc.teragrid.org
  stork-2.0.1
  CVS
  condorlib
  config
  externals

login1.ls4.tacc.utexas.edu
  createDummyFiles.py
  createFolders.pl
  externals.tar.gz
  stork-2.0.1.tar.gz
  x1
Stork Android Client

Stork Client

`tg-login.spur.tacc.teragrid.org`
- stork-2.0.1
- CVS
- condorlib
- config
- externals

`login1.ls4.tacc.utexas.edu`
- createDummyFiles.py
- createFolders.pl
- externals.tar.gz
- stork-2.0.1.tar.gz
- x1
Stork Android Client

tg-login.spur.tacc.teragrid.org

login1.ls4.tacc.utexas.edu

- copy
- externals
- globus

Disconnect All
<table>
<thead>
<tr>
<th>Job</th>
<th>Progress</th>
<th>URL 1</th>
<th>URL 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Finished</td>
<td>ftp://didc-ws8.buffalo.edu</td>
<td>irods://didc-ws7.buffalo.edu</td>
</tr>
<tr>
<td>3</td>
<td>98%</td>
<td>gsiftp://tg-login.spur.tacc.teragrid.org/etc/1.dat</td>
<td>gsiftp://nbirn.org</td>
</tr>
<tr>
<td>5</td>
<td>Queued</td>
<td>gsiftp://loni.org</td>
<td>gsiftp://dest.dsl-stork.org/home/sivahpc/test/dest</td>
</tr>
<tr>
<td>6</td>
<td>95%</td>
<td>gsiftp://dest.dsl-stork.org/home/sivahpc/test/dest</td>
<td>gsiftp://loni.org</td>
</tr>
</tbody>
</table>
Stork Android Client

Job Progress

1
Job ID: 1
Job Details
Cancel Job
Remove From List
Stork Android Client

tg-login.spur.tacc.teragrid.org
- stork-2.0.1
- CVS
- condorlib
- config

login1.ls4.tacc.utexas.edu
- createDummyFiles.py
- createFolders.pl
- externals.tar.gz
- stork-2.0.1.tar.gz

Disconnect All
Stork for the Cloud

- Prototype implementation complete, testing stage
- Will be deployed as hosted service
- Allow deployment on private clouds as well
- Available on Amazon EC2 and Windows Azure
- More optimizations coming
100 Gbit Performance

[Graph showing throughput over time with peak performance around 100 Gbit/s and a steady decline thereafter.]
Summary

• Scientific and commercial applications are getting more and more data intensive

• Data sharing and bulk data transfers are still a major bottleneck in front of multi-institutional and inter-disciplinary collaborative science

• Stork for the Cloud provides end-to-end throughput optimization in hosted environment accessible through ultra-thin clients
This work has been sponsored by:

**NSF, DOE, ONR, NOAA**

For more information:

**Stork web page:** [http://www.storkproject.org](http://www.storkproject.org)
This work has been sponsored by:
NSF, DOE, ONR, NOAA

For more information:
Stork web page: http://www.storkproject.org