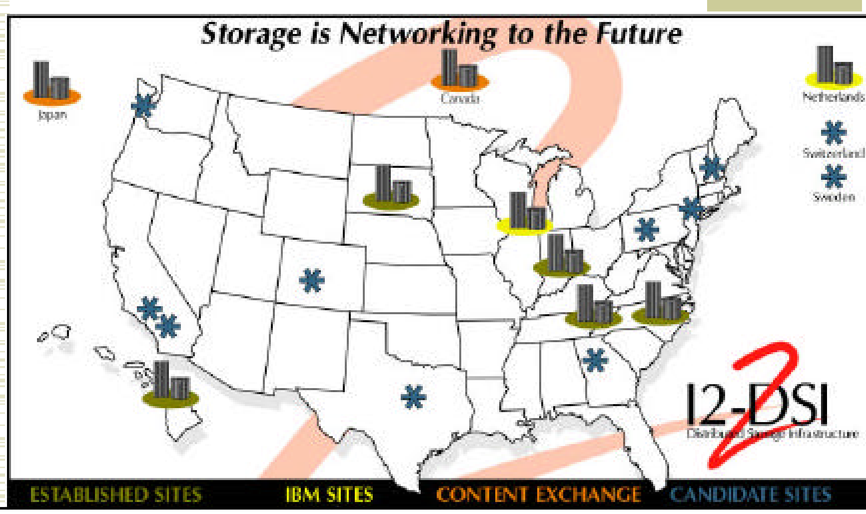
Content Replication in I2-DSI using Rsync+

Bert J Dempsey
Debra Weiss

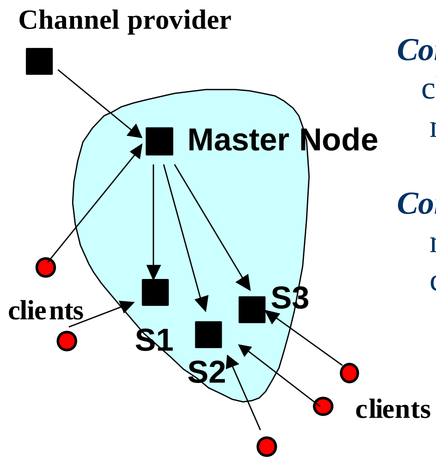
University of North Carolina at Chapel Hill
dempsey@ils.unc.edu

Multiple-site replication in I2-DSI

http://dsi.internet2.edu/



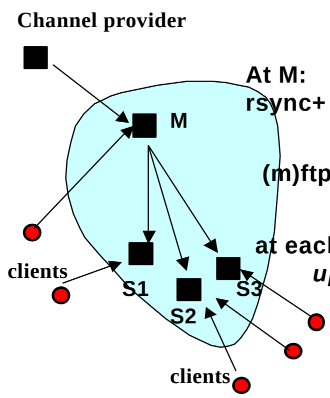
Replicating Channels



Content import from channel provider to master I2-DSI node

Content Replication to all replication sites that carry the channel

Rsync+ for Content Replication



At M: **Master-side**

rsync+ -F srcLatest/ src/ → *updates*

(m)ftp *updates*

at each of S1, S2, S3: **Slave-side**

***updates* → rsync+ -f src/**

Rsync is popular filesystem sync tool

Rsync+ is our mods to enable local capture of update info for store-and-forward communication

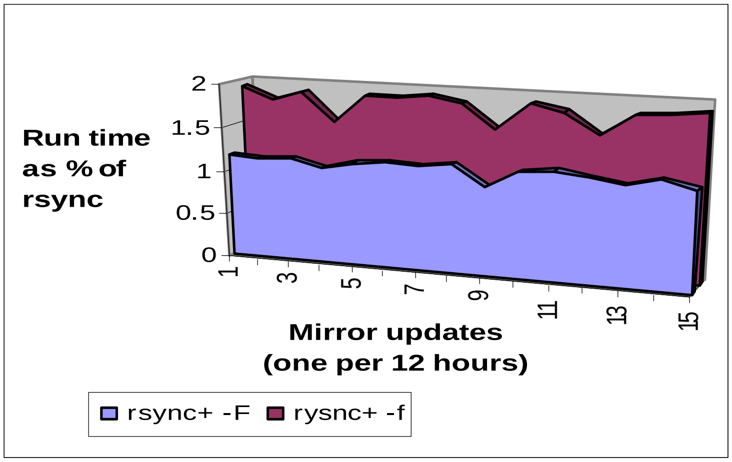
Server Experiment

- Instrumented Mirror
  - Active Linux repository (8 GB, 25,000 files)
  - Twice daily synchronization
- On dsi.ncni.net:
  - **rsync+ -F**: Perform **master-side** rsync+ processing between two local directories to create *updates* file
  - **rsync+ -f**: Use *updates* to perform **slave-side** rsync+ processing

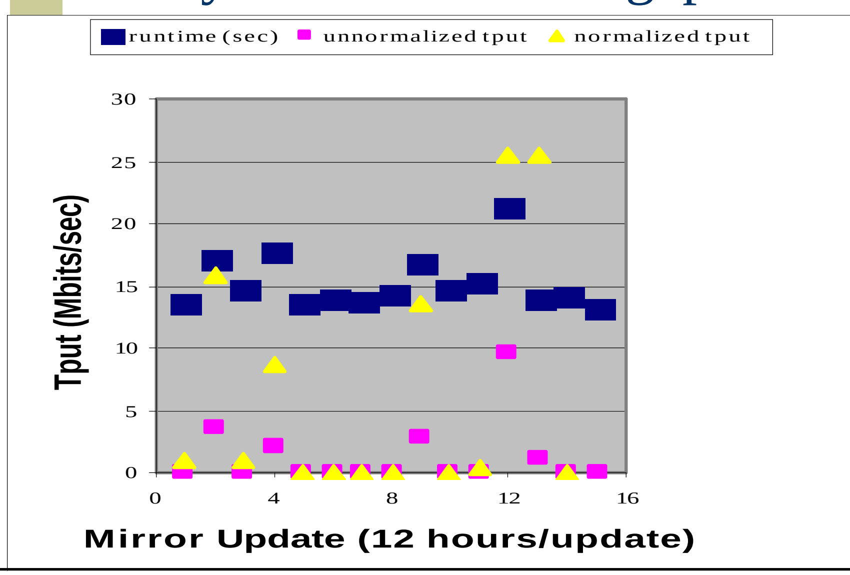
Content Change Patterns

- Data here from 1-month Linux mirror
  - Update per 12-hour period
  - No files to change 13 of 60 periods (21%)
- Average size of updated data (all periods)
  - 0.144% of aggregate archive
  - 0.104% under rsync+
- Maximum size of updated data
  - 2.42% of mirror

Rsync+ processing cost



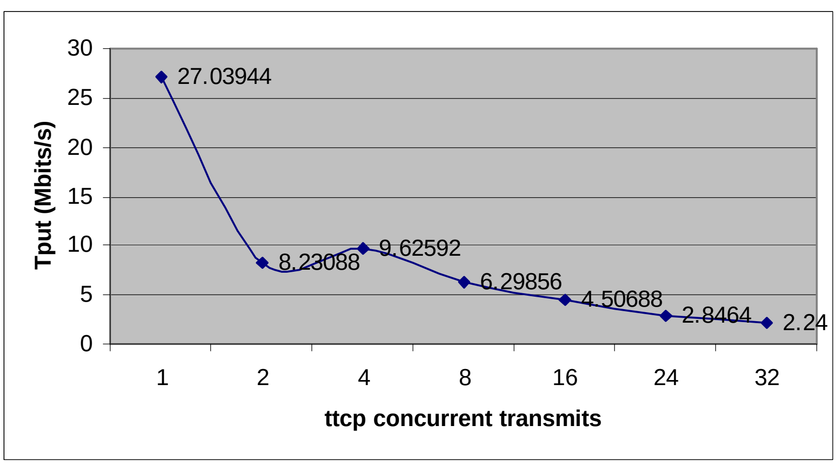
Rsync+ Local Throughput



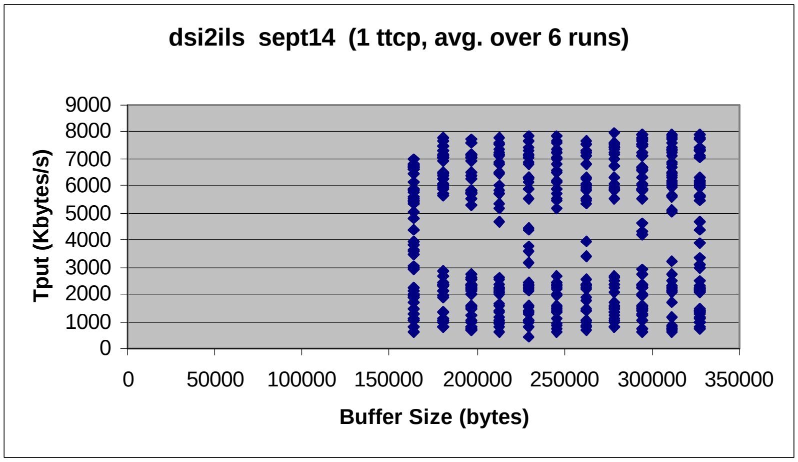
Network Throughput: ttcp experiment parameters

Parameter	*Values*
File Size	5.45 MB
Network Path (100 Mbit/s min)	dsi.ncni.net → ils.unc.edu
Concurrent *ttcp* connections	1,2,4,8,16,24,32
Receiver socket buffer size (KB)	240 KB
Buffer Policy	240KB / 240KB shared

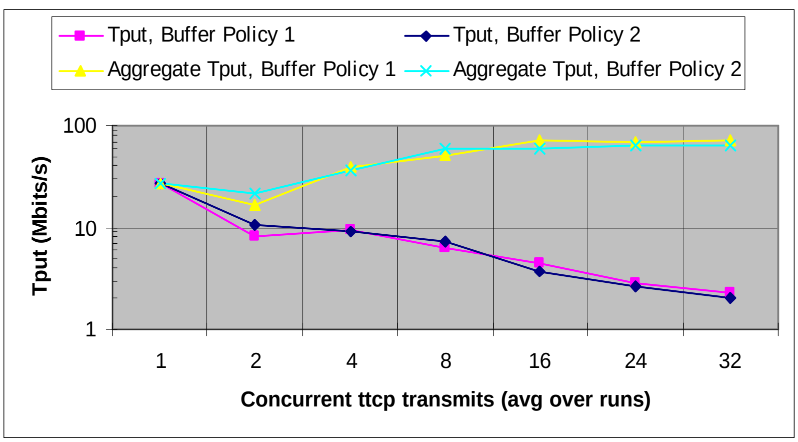
Network Throughput: concurrent ttcp transmits



Network experiments: setting socket buffer sizes



Network Throughput: concurrent ttcp tputs



Baseline Scalability Analysis using empirical inputs

- Update of content
  - 0.1 % avg, 2.4 % maximum
- Network tput
  - 8 Servers, thus 6.2 Mbits/sec to each
- Server tput (local rsync actions)
  - Master: 11.4 Mbits/sec
  - Slave: 8.18 Mbits/sec

Baseline Scalability Analysis: end-to-end update latency

Content Channel Size	Updates Avg Max	Master processing latency	Network latency	Slave processing latency	End-to-end update latency
10 GB	10 MB 240 MB	7 secs 168 sec	12.9 secs 309 sec	9.7 secs 233 s	29.6 sec 710 s
100 GB	100 MB 2.4 GB	70 sec 28 min	129 sec 51.5 min	97 sec 38.8 min	296 sec 118.3 min
1 TB	1 GB 24 GB	11.7 min 280.8 min	21.5 min 516 min	16.1 min 386.4 min	49.3 min 19.72 hrs

Conclusions

- Our work creates scalable design for filesystem-level tool for data synchronization
- Current systems without tuning suggest O(100 GB) content can be handled for initial server set
- For TB content, system advances will need to provide speed-ups
  - Tuning
  - Hardware
  - Distributed processing