Monitoring end-to-end performance on TGrid

Shava Smallen (INCA project)

Margaret Murray (CAIDA)

Talk Outline

- Project goal
- Terms and Conditions
- Measuring available bandwidth: pathload
- The INCA architecture
- Demonstration
- Future Directions
- Discussion

Goal: Understand TGrid end-to-end performance

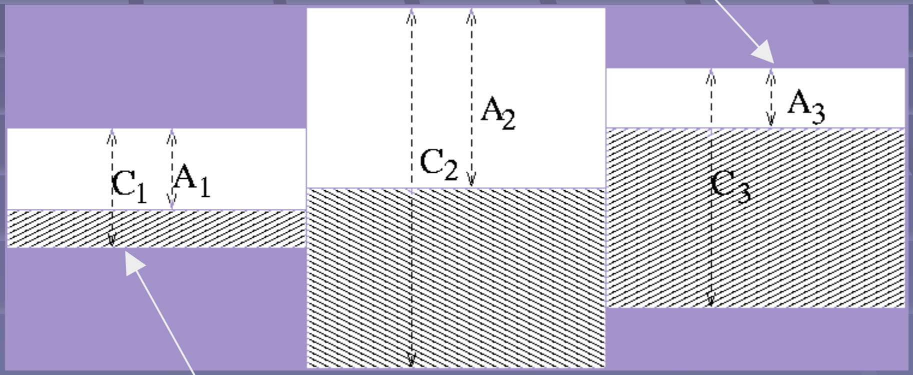
- User goals:
  - Optimize my application performance
  - Move my data… FAST
  - With whom am I sharing network bandwidth?
- Sysadmin goals:
  - Identify problems
  - Set realistic performance expectations
- Common denominator:
  - Maximize available bandwidth

Terms

“Bottleneck” is not a meaningful term

- e2e Capacity (C): min link capacity in the path
- e2e avail-bw (A): min unused bandwidth at time T
- BTC: max achievable TCP throughput

Tight link A3 (avail-bw)



Narrow link C1 (capacity)

…and Conditions

(factors in e2e network performance)

- Router buffer sizes and COS or QoS
- Host TCP settings
- Cross-traffic (load level, burstiness)
- Traffic type mix
  - TCP ==> guaranteed delivery + fair share
  - UDP ==> no guaranteed delivery
- Number of competing streams

Measuring end-to-end Available Bandwidth

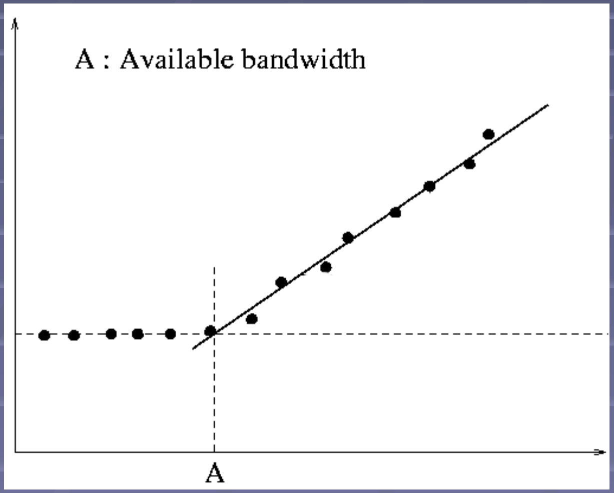
- It’s not easy, and tools haven’t been validated.
  - Even fewer tools developed and validated on high speed links.

CAIDA is performing first comprehensive tool evaluation on high speed links in CAIDA/SDSC lab.

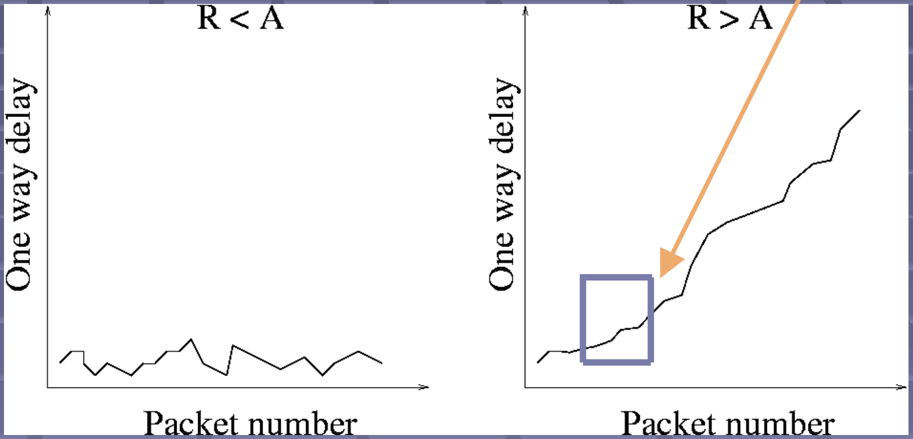
- Iperf (persistent TCP connection w/ large advertised window)
  - Can be intrusive: can saturate the path and increase path delays and jitter…depending on time scale
  - Measures “brute force” avail-bw
- Pathload (Self-Loading Periodic Streams)
  - Attempts to be non-intrusive over time (uses < 10% avail-bw)
  - Measures the dynamics of avail-bw over time

How pathload works…

Concept:



…find the range of the knee



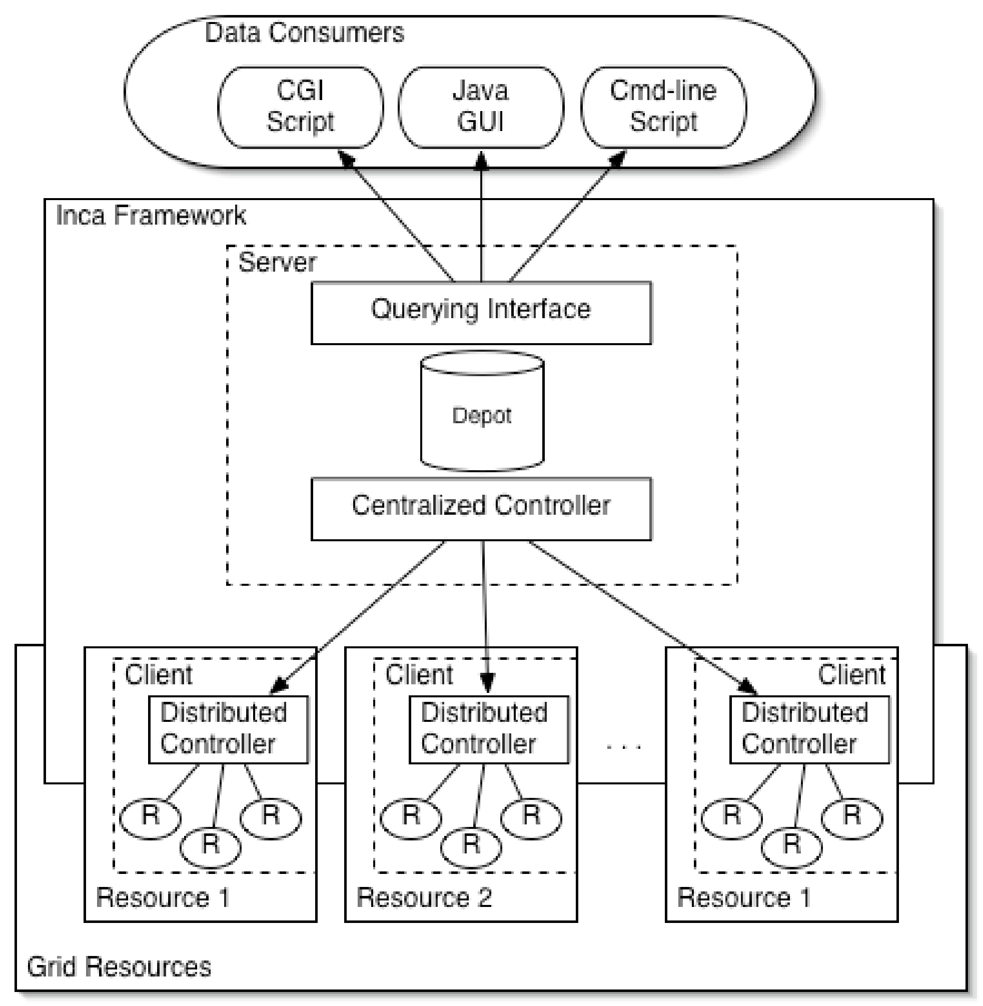
- Send ≈100 probes of equal-sized packets at rate R and measure one-way delays; iterate while modifying R (and limit probing rate to < 10%)
- One-way delays only increase when the stream rate R is *larger* than the avail-bw A

Why use INCA?

- Infrastructure exists and it works!
- Take advantage of INCA’s:
  - Full mesh deployment
  - Data repository/archive
  - Web interface
  - Schedule options
- To collect network performance data:
  - Add Network Reporter
    - Reporter-Pair - a new variation
    - Same wrapper can work with multiple avail-bw tools

Inca Architecture

- Data consumer - user-friendly web interface, application, etc.
- Framework - daemons
  - Planning and execution of reporters
  - Centralized data collection
  - Publishing
- Reporter - a script or executable

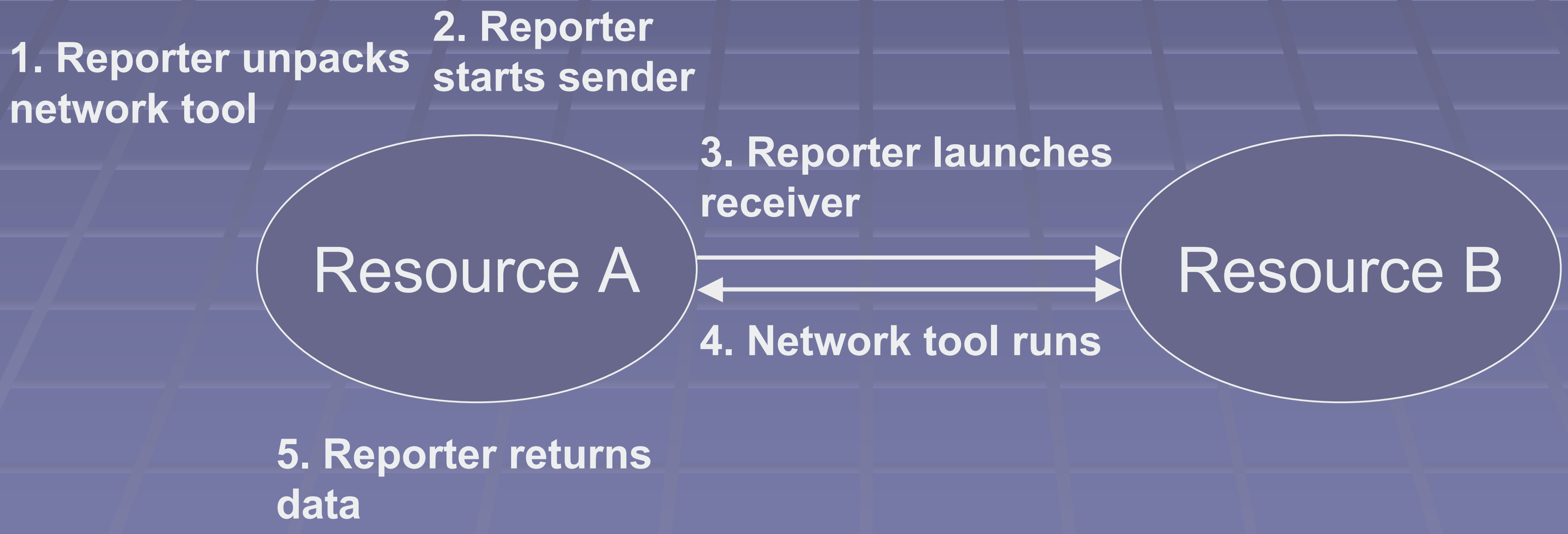


Gathering performance data

1. Write reporter to wrap benchmark and print XML output according to Inca reporter specification
2. Write configuration file to express:
   a) Inputs
   b) Frequency of execution
   c) Data to archive
3. Write web page to display data

Writing performance reporter

- Perl API to enable running of network probes across sites (uses globusrun)



1. Reporter unpacks network tool

2. Reporter starts sender

3. Reporter launches receiver

Resource A

Resource B

4. Network tool runs

5. Reporter returns data

Executing reporter

- Cron scheduling
  - Schedule far enough apart so they don't collide
  - Not foolproof
- Move to token-passing protocol (NWS)?

Graphing data

- Calls rrdtool commands to generate graphs
- CGI script currently uses SOAP call to get graph from Inca archive

https://repo.teragrid.org/inca/html/pathload.html

Future Directions…

- Scheduling frequency
  - Now: once/hr
  - Check result distributions
  - Refine scheduling: Move to token-passing protocol (NWS)?
- Compare results of other tools
  - pathload, pathchirp, Spruce, iperf
  - Consider error and overhead
- Refine graphs and web interface
- Run network probes across different Oses
- Consider more e2e paths than just between login nodes
- *Coming soon: SRB reporter in the works…*

Discussion

- Will my application perform better if I don’t use TCP?

- *Claim: TCP is not suitable for Grid apps in high-performance networks.*

- *Claim: I can get better performance with UDP*
  - *…careful what you wish for! App must control everything*
  - *“Doesn’t play well with others.”*

Try SOBAS instead!

- Socket Buffer Auto-Sizing (SOBAS) [Prasad, Jain & Dovrolis, GaTech]
  - Apps use a SOBAS enabled socket library.
  - Concept: Limit the send window after reaching avail-bw to avoid “self-induced” packet loss.
  - Experimental results show 20-80% increase in throughput compared to TCP transfers using max possible socket buffer size.

R. Prasad, M. Jain and C. Dovrolis, “Socket Buffer Auto-Sizing for High-Performance Data Transfers” Journal of Grid Computing June 2004. http://www.cc.gatech.edu/~ravi/tools/sobas.tar.gz

Summary

- The INCA architecture now supports available bandwidth measurements.
- Pathload reports a range variation of available bandwidth on an e2e path.
- INCA/pathload measures available bandwidth on TGrid e2e paths (login node to login node).