DataTurbine for science

Paul Hubbard
hubbard@sdsc.edu
June 5 2008

Cyberinfrastructure Lab for Environmental Observing Systems
Science R&D
SDSC/UCSD
Outline

• A bit about streaming data
• Introduction to DataTurbine
• Science motivations
• Where to use DataTurbine
• Where not to use DataTurbine
• Example applications and related topics
• Can I use it?
• Please ask questions at any time!
A bit about streaming data

- TCP/IP makes sending data from point A to point B easy. What’s the big deal?
  - Add more listeners
  - Add ability to have listeners subscribe to different streams
  - What happens to the data if and when the network fails?
  - What happens if one listener is slower than the others?
- The canonical answer to the general problem is a ‘ring buffer’.
- Multicast solves some but not all of this.
- Mainly useful for distributed systems...
Science motivations for DataTurbine

• You have...
  • Collaborators spread out, or remote sites

• You need...
  • To see data as soon as possible
  • To integrate equipment from different vendors

• You want...
  • To integrate video or photographs with your numeric data
  • The ability to collaboratively annotate the experiment as it happens
  • To help us debug our audio support
  • To play around with live data + Google Earth
Why do you want a ring buffer?

- Ring buffers (a.k.a circular buffers) keep a finite history for each stream
  - Accommodate slow clients, post-processing, replay
  - The buffers can be stored on disk and therefore be arbitrarily large
  - Programs can subscribe to streams; very efficient
- There’s no infinite buffer, so circular is the best you can do.
- Make each buffer as long as you want; disk is cheap!
When to use something else

• **Low-availability network connections**
  • E.g. MBARI ocean buoys that dial up over satellite networks. We have no existing mechanism for this sort of batch transfer; JMS would probably be better.

• **Simple topology, single vendor**
  • Point A -> Point B -> files -> MATLAB is often a complete solution

• **Signal-limited data**
  • If you absolutely need every data point, a transactional system (database, JMS) provides stronger guarantees.
What is DataTurbine?

- DataTurbine is an open source, Java based network ring buffer for all sorts of data. You can use memory + disk for the ring and it runs on almost any JVM.
- Started life as a NASA telemetry project
- The basic division of work looks like this:
‘A series of tubes’

- The web is very similar: Browser, webserver and back-end data from a variety of sources.
A complex example
Sounds complex. Is it fast?

Yes! This is a Macbook pro to T2000 over gigabit - ~30MB/sec from MATLAB.

* We burst 65MB/sec on a single gigabit link
More about DataTurbine

- Sources can have multiple channels with varied types - numeric (e.g. sensors), video, audio, text, binary blobs.
- We have a variety of sources and sinks: In-house, from the original author Creare and also community contributed.
- Can also use plugins for tightly-coupled computations such as image processing.
- Runs on J2ME, J2EE and 64-bit JVM as well. Extremely scalable.
Deployments of DataTurbine

- I’ve included some screenshots of DataTurbine in use to give a flavor of current utilization.
- I’m hoping to give you some ideas how you might use it yourself, so please ask or interrupt.
CLEEOS/HPWREN deployment at Santa Margarita Ecological Reserve
Santa Margarita Ecological Reserve
**NCHC Kenting (Taiwan)**

- Kenting National Park and Yuan-Yang Lake, pictures from Fang-Pang Lin and Ebbe Strandell
More Kenting
SUNY Buffalo - Earthquake engineering
Insight Racing - video

- DARPA autonomous vehicle competition
- Insight is using DataTurbine for their vehicle video in their Lotus
- North Carolina State University, using multiple Axis 206 cameras, 30fps each
- http://www.insightracing.org/
Lehigh - 3D viewer
NASA Dryden Flight Center

- Intelligent Network Data Server (INDS)
- Fusion of DataTurbine, Google Earth and live telemetry
- Instruments flown on ER-2 (U2) and DC-8
One more NASA
You can also view data via the web

Data, converted to graphs by the PNG plugin

- /neespop/DAQ/BaseAccelerationY
- /neespop/DAQ/BaseAccelerationX
- /neespop/FakeDAQ/0
- /neespop/FakeDAQ/1
System monitoring

• “A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.” --Lamport

• Baseline: Web-accessible system for monitoring hardware, software and network

• Better: Keep a history of up/down/metrics

• Best: Infer state from events and automatically act. E.g., “All services and ICMPs are down from X. I infer from this that the network has failed.”

• We have ‘better,’ but not ‘best’: Inca, Monit, M/Monit
**Monit in action - basic state**

**Monit Service Manager**

Monit is running on iguassu with `uptime 13m` and monitoring:

<table>
<thead>
<tr>
<th>System</th>
<th>Status</th>
<th>Load</th>
<th>CPU</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>iguassu.sdsc.edu</td>
<td>running</td>
<td>[0.00] [0.00] [0.00]</td>
<td>0.0%us, 0.0%sy, 0.0%wa</td>
<td>3.2% [198808 kB]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Status</th>
<th>Space usage</th>
<th>Inodes usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>rootlvm</td>
<td>accessible</td>
<td>77.9% [194.6 MB]</td>
<td>21.2% [15079 objects]</td>
</tr>
<tr>
<td>tmp</td>
<td>accessible</td>
<td>9.0% [14.8 MB]</td>
<td>0.3% [315 objects]</td>
</tr>
<tr>
<td>usr</td>
<td>accessible</td>
<td>44.5% [1898.4 MB]</td>
<td>12.0% [73053 objects]</td>
</tr>
<tr>
<td>var</td>
<td>accessible</td>
<td>18.3% [382.1 MB]</td>
<td>1.6% [912 objects]</td>
</tr>
<tr>
<td>home</td>
<td>accessible</td>
<td>5.1% [457.7 MB]</td>
<td>0.0% [19871 objects]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Host</th>
<th>Status</th>
<th>Protocol(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>niagara.sdsc.edu</td>
<td>online with all services</td>
<td>[ICMP Echo Request]</td>
</tr>
<tr>
<td>niagara-stage.sdsc.edu</td>
<td>online with all services</td>
<td>[ICMP Echo Request]</td>
</tr>
<tr>
<td>niagara-dev.sdsc.edu</td>
<td>online with all services</td>
<td>[ICMP Echo Request]</td>
</tr>
</tbody>
</table>
m/monit: time-series monitoring

Latest status

Hosts Status

Events in last 24-hours
INCA monitoring GLEON

Summary Status for Common, Inca, PoP, Rocks

The map below uses the Google Maps API to display a summary status for Inca test suites.

Click on resource markers to view test errors for individual resources (any cross-resource tests will have toggle buttons to display them under the map image).
### Snippet of detailed Inca view

<table>
<thead>
<tr>
<th>java version</th>
<th>1.6.0_23</th>
<th>1.5.0_07</th>
<th>1.6.0_25</th>
<th>1.6.0_25</th>
</tr>
</thead>
<tbody>
<tr>
<td>openssh version</td>
<td>4.7p1</td>
<td>3.9p1</td>
<td>4.7p1</td>
<td>4.7p1</td>
</tr>
<tr>
<td>os version</td>
<td>1.5.24(0.156/4/2)</td>
<td>2.6.9-42.0.2.ELsmp</td>
<td>1.5.24(0.156/4/2)</td>
<td>1.5.25(0.156/4/2)</td>
</tr>
</tbody>
</table>

**Inca**

- all2all:ping to lakeErken
- all2all:ping to lakeSunapee
- all2all:ping to wisconsin
- apache http page test
- httpd version
- inca consumer test
- postgres connect test
- postgres version

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Inca Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>all2all:ping to lakeErken</td>
<td>pass</td>
</tr>
<tr>
<td>all2all:ping to lakeSunapee</td>
<td>error</td>
</tr>
<tr>
<td>all2all:ping to wisconsin</td>
<td>pass</td>
</tr>
<tr>
<td>apache http page test</td>
<td>pass</td>
</tr>
<tr>
<td>httpd version</td>
<td>2.0.52</td>
</tr>
<tr>
<td>inca consumer test</td>
<td>pass</td>
</tr>
<tr>
<td>postgres connect test</td>
<td>pass</td>
</tr>
<tr>
<td>postgres version</td>
<td>7.4.13</td>
</tr>
</tbody>
</table>

**PoP**

- rbnb ping
- rbnb version

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>rbnb ping</td>
<td>error</td>
</tr>
<tr>
<td>rbnb version</td>
<td>trunk build 2862</td>
</tr>
</tbody>
</table>

**Open Source Data Turbine Initiative**

Empowering the Scientific Community with Streaming Data Middleware

SAN DIEGO SUPERCOMPUTER CENTER, UCSD
Can I use it? Current device support

- Data acquisition (DAQ)
  - National Instruments (NI-DAQ, DAQmx, Compact RIO) via Java proxy
  - Campbell Scientific: File-based, via LoggerNet, up to 1Hz tested.
  - Dataq Instruments (serial connect via C & DaqToRbnb)
  - PUCK, (Programmable Underwater Connector with Knowledge)
  - Seabird/Seacat
  - Vaisala weather station

- Video and still cameras
  - Anything with motion JPEG via URL: Axis, Panasonic, etc
  - Still images via WebDAV, HttpMonitor

- Accelerometers
  - ADXL202 and Apple laptop
Can I use it? Software support

- Primary interface is via the Java-based API
- Other avenues
  - If you’re on Windows, there’s ActiveX
  - TCP/UDP proxy (some code required)
  - WebDAV/HTTP (Not as fast, but cross-platform and very flexible)
  - Java-based proxy on TCP/IP - we use this a lot
  - MATLAB API provided, including performance metrics and test suite.
- Geotagging and Google Earth...works now!
Is my data captive?

• Definitely not:
  • For permanent storage, we have tools to stream data to files on disk or to an SQL database.
  • You can also load data from disk (numeric or video) back into DataTurbine

• You can also do real-time analysis from within MATLAB, reading and writing data from the DataTurbine
Caveats, gotchas and limitations

• All data is timestamped, therefore clocks have to be synchronized, usually via NTP
• As previously explained, a modular distributed system is not appropriate for all research
• Basic host-based security; no encryption or per-user access control
  • No need so far, can be tunneled over VPN, VLAN or SSH
• Java is easiest, but not always possible.
• Flexible, modular => learning curve.
Upcoming developments

• Working on a VMWare virtual machine based on Debian Linux that has DataTurbine, RDV, source programs, INCA, Tomcat, plugins, etc.
  • Download and run on Linux, Mac, Windows

• Evaluation using Tomcat/HTML as a GUI for command-line source programs

• Google Earth integration
  • Add lat/long/elevation metadata to sources
  • Write server-side pages for graphs on demand
Tomcat as a GUI

Portable Network Graphic Plugin

This Plugin converts time series data to a PNG format plot viewable by most browsers.

RBNB server address: localhost:3333
PlugIn client name: PNGPlugIn
Width of plot (pixels): 320
Height of plot (pixels): 200

Start PlugIn

Help - Home
Motivations, or ‘Who pays the bills?’

- In summer 2007, the CLEOS group at SDSC won a two-year NSF award under the SDCI (Software Development for Cyberinfrastructure Improvement) to work on DataTurbine.
- Also funded other grants (Moore foundation, GLEON, MoveBank).
- Our agenda is ‘software for science.’
- ... and we’re always interested in new uses of streaming data that move science forward.
Where to learn more?

- Main site is http://dataturbine.org/
  - Mailing list, docs, FAQ, links to Subversion, code, etc.
- rbnb-dev@sdsc.edu mailing list is the best place to ask questions
- My email is hubbard@sdsc.edu
Quick demo if time (and interest)

- On laptop, run
  - DataTurbine
    - `java -jar $RBNB_HOME/bin/rbnb.jar`
  - SMS-RBNB to measure and stream 3-axis accelerometer
    - `cd $SMS; ant run`
  - ISightToRbnb to stream onboard video camera
    - `java ISightToRbnb`
  - RDV to display and explore the data
    - `java -jar rdv.jar`
- SMS-RBNB is Mac-only, rest are for any platform
In case the demo barfs...slides on a plane!
Brainstorming uses