Firehose Engineering

designing high-volume data collection systems
Firehose Database Applications (FDA)

(1) very high volume of data input from many automated producers

(2) continuous processing of incoming data
Mozilla Socorro
Upwind
Fraud Detection System
Firehose Challenges
1. Volume

- 100's to 1000's facts/second
- GB/hour
1. Volume

- spikes in volume
- multiple uncoordinated sources
1. Volume

volume always grows over time
2. Constant flow

since data arrives 24/7 …

while the user interface can be down, data collection can never be down
2. Constant flow

- can't stop receiving to process
- data can arrive out of order
3. Database size

- terabytes to petabytes
  - lots of hardware
  - single-node DBMSes aren't enough
  - difficult backups, redundancy, migration
  - analytics are resource-consumptive
3. Database size

• database growth
  • size grows quickly
  • need to expand storage
  • estimate target data size
  • create data ageing policies
3. Database size

“We will decide on a data retention policy when we run out of disk space.”
– every business user everywhere
many components = many failures
4. Component failure

- all components fail
  - or need scheduled downtime
  - including the network
- collection must continue
- collection & processing must recover
solving firehose problems
socorro project
http://crash-stats.mozilla.com
Top Crashers for Firefox 6.0.2

Top 300 Crashing Signatures, 2011-09-21 through 2011-09-28. The report covers 69.60% of all 775208 crashes during this period. Graphs below are dual-axis, having Count (Number of Crashes) on the left X axis and Percent of total or Crashes on the right X axis.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Trend</th>
<th>%</th>
<th>Diff</th>
<th>Signature</th>
<th>Count</th>
<th>Win</th>
<th>Mac</th>
<th>Lin</th>
<th>Ver</th>
<th>First Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>10.63%</td>
<td>3.27%</td>
<td>domly signature/learn More</td>
<td>82656</td>
<td>73</td>
<td>2</td>
<td>0</td>
<td></td>
<td>2011-01-31</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3.43%</td>
<td>0.04%</td>
<td>plugin::PluginRuntime::PluginClientProcess:mozilla::</td>
<td>27084</td>
<td>27084</td>
<td>0</td>
<td>0</td>
<td>83</td>
<td>2011-01-31</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2.01%</td>
<td>-0.04%</td>
<td>plugin::PluginRuntime::PluginClientProcess:mozilla::</td>
<td>22573</td>
<td>22573</td>
<td>0</td>
<td>0</td>
<td>82</td>
<td>2011-01-31</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2.50%</td>
<td>0.06%</td>
<td>plugin::PluginRuntime::PluginClientProcess:mozilla::</td>
<td>19334</td>
<td>19334</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2011-01-31</td>
</tr>
<tr>
<td>5</td>
<td>-3</td>
<td>2.08%</td>
<td>-0.03%</td>
<td>Plugin::PluginRuntime::PluginClientProcess:mozilla::</td>
<td>6135</td>
<td>6135</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2011-02-04</td>
</tr>
</tbody>
</table>
Mozilla Socorro

collectors

processors

webservers

reports
socorro data volume

- 3000 crashes/minute
  - avg. size 150K
- 40TB accumulated raw data
  - 500GB accumulated metadata / reports
dealing with volume

load balancers  collectors
dealing with volume

monitor

processors
dealing with size

- Data: 40TB expandible
- Metadata views
- 500GB fixed size
dealing with component failure

Lots of hardware ...

- 30 Hbase nodes
- 2 PostgreSQL servers
- 6 load balancers
- 3 ES servers
- 6 collectors
- 12 processors
- 8 middleware & web servers

... lots of failures
load balancing & redundancy

load balancers

collectors
elastic connections

- components queue their data
- retain it if other nodes are down
- components resume work automatically
- when other nodes come back up
elastic connections

collector

receiver

local file queue

crash mover
server management

- puppet
  - controls configuration of all servers
  - makes sure servers recover
  - allows rapid deployment of replacement nodes
Upwind
Upwind

- speed
- wind speed
- vibration
- noise
- direction
Upwind

1. maximize power generation
2. make sure turbine isn't damaged
dealing with volume

each turbine: 90 to 700 facts/second
windmills per farm: up to 100
number of farms: 40+
est. total: 300,000 facts/second
(will grow)
dealing with volume

local storage -> historian -> analytic database

local storage -> historian -> analytic database

reports

reports
dealing with volume

local storage ➔ historian ➔ analytic database ➔ master database

local storage ➔ historian ➔ analytic database ➔ master database
multi-tenant partitioning

- partition the whole application
  - each customer gets their own toolchain
- allows scaling with the number of customers
  - lowers efficiency
  - more efficient with virtualization
dealing with:
constant flow and size
time-based rollups

- continuously accumulate levels of rollup
  - each is based on the level below it
  - data is always appended, never updated
  - small windows == small resources
time-based rollups

- allows:
  - very rapid summary reports for different windows
  - retaining different summaries for different levels of time
  - batch/out-of-order processing
  - summarization in parallel
firehose tips
data collection must be:

- continuous
- parallel
- fault-tolerant
data processing must be:

- continuous
- parallel
- fault-tolerant
every component must be able to fail

- including the network
- without too much data loss
- other components must continue
5 tools to use

1. queueing software
2. buffering techniques
3. materialized views
4. configuration management
5. comprehensive monitoring
4 don'ts

1. use cutting-edge technology
2. use untested hardware
3. run components to capacity
4. do hot patching
firehose mastered?
Contact

- Josh Berkus: josh@pgexperts.com
- blog: blogs.ittoolbox.com/database/soup
- PostgreSQL: www.postgresql.org
- pgexperts: www.pgexperts.com

Upcoming Events

- PostgreSQL Europe: http://2011.pgconf.eu/
- PostgreSQL Italy: http://2011.pgday.it/