Firehose Engineering

designing high-volume data collection systems

Josh Berkus
HiLoad++, Moscow
October 2011

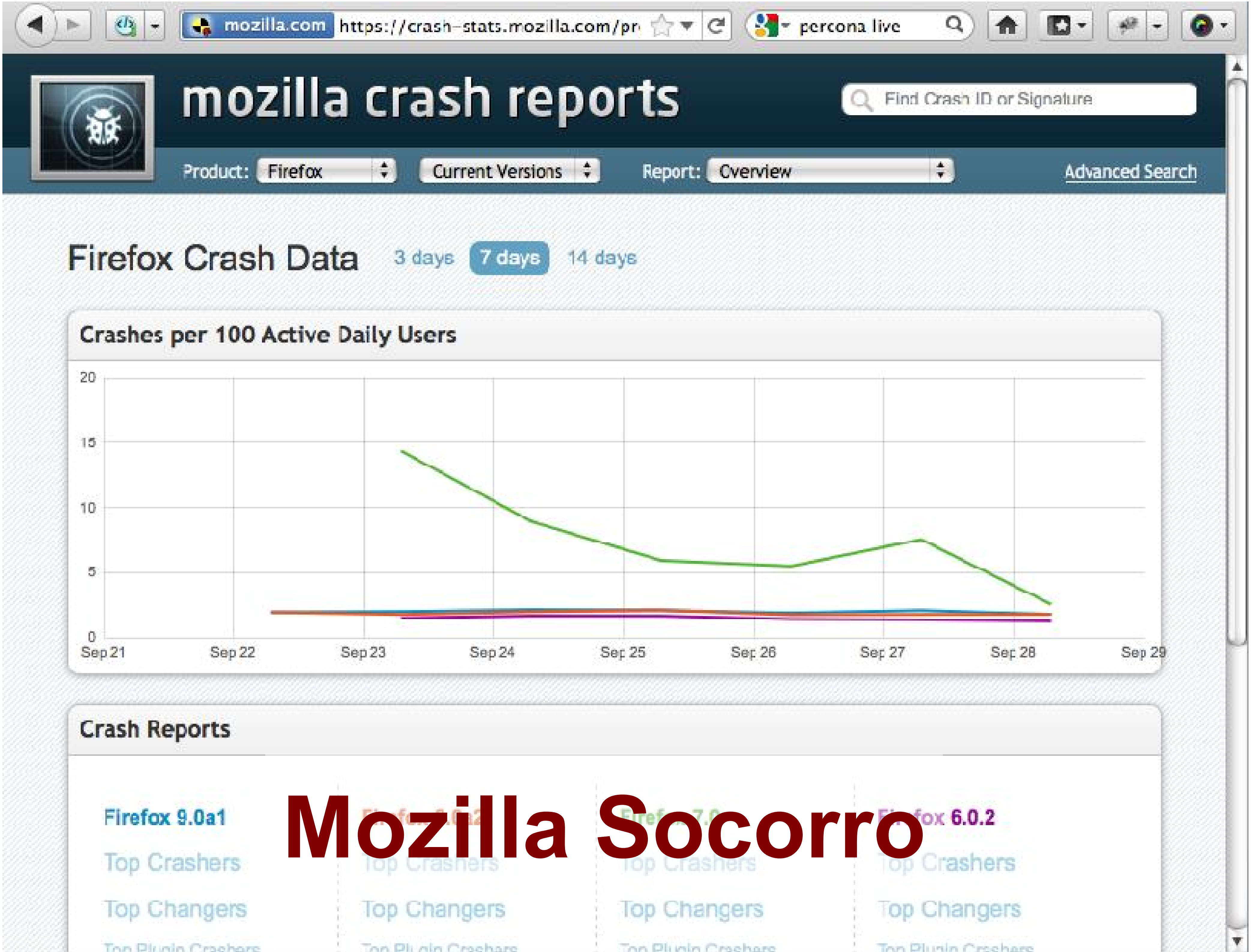
PGX
POSTGRESQL
EXPERTS, INC.

Firehose Database Applications (FDA)

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

(2) continuous processing of incoming data

Mozilla Socorro



Firefox Crash Data

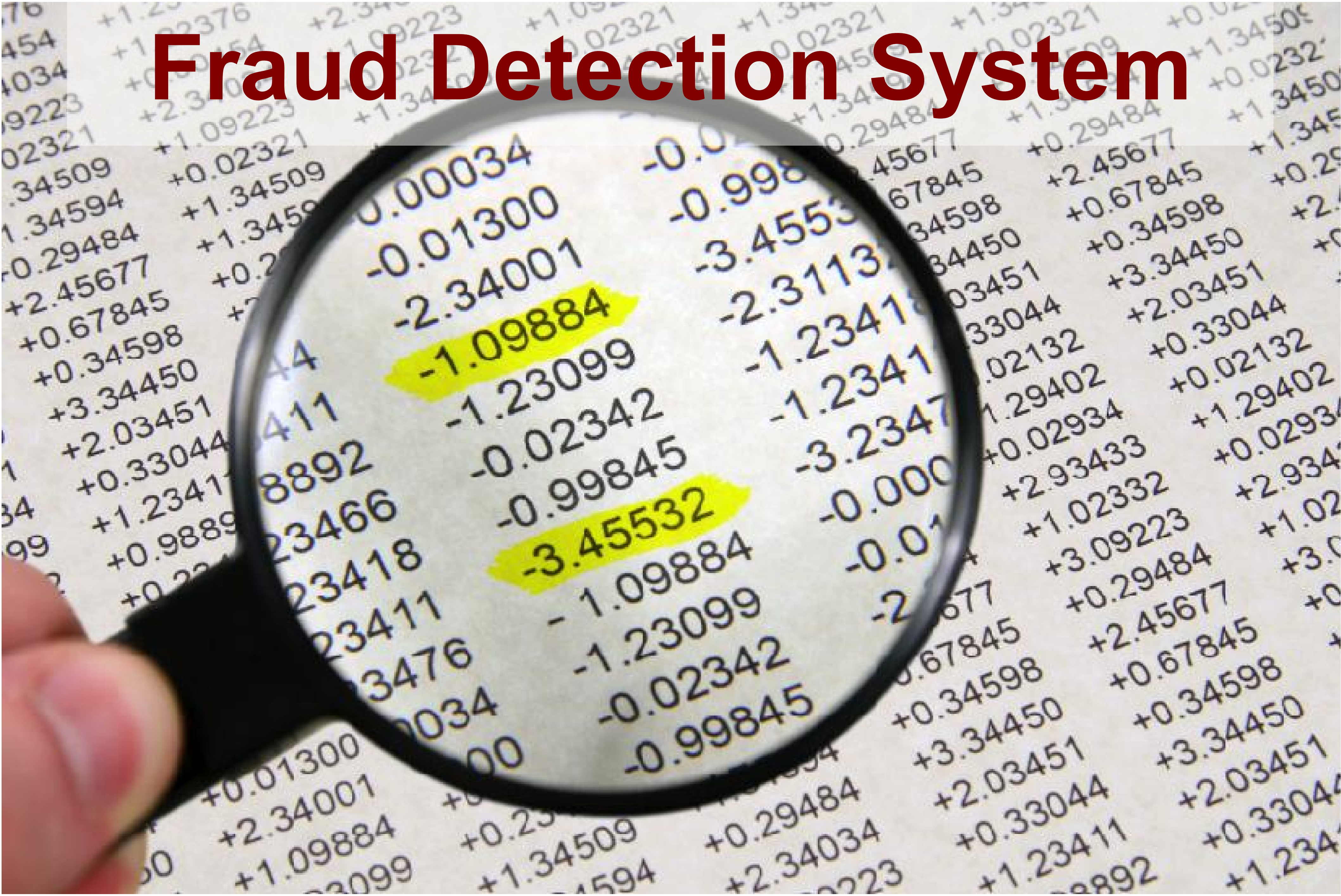
Crashes per 100 Active Daily Users

Crash Reports

Firefox 9.0a1	Firefox 8.0a2	Firefox 7.0	Firefox 6.0.2
Top Crashers	Top Crashers	Top Crashers	Top Crashers
Top Changers	Top Changers	Top Changers	Top Changers
Top Plugin Crashers	Top Plugin Crashers	Top Plugin Crashers	Top Plugin Crashers

Upwind

Fraud Detection System



Firehose Challenges

1. Volume

- 100's to 1000's facts/second
- GB/hour

1. Volume

- spikes in volume
- multiple uncoorindated 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

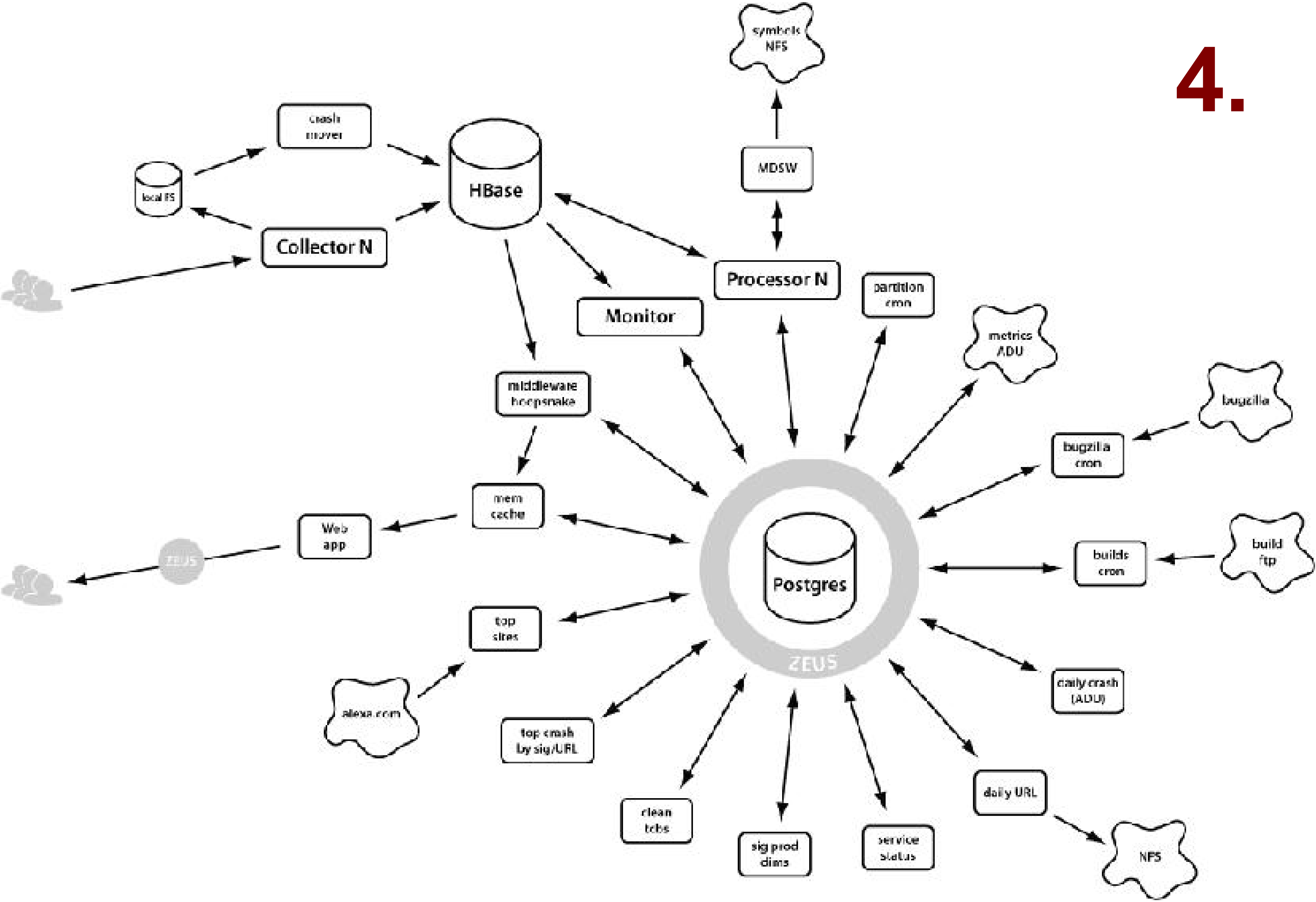
- database growth
  - size grows quickly
  - need to expand storage
  - estimate target data size
  - create data ageing policies

3. Database size

“We will decide on a data retention policy when we run out of disk space.”

– every business user everywhere

4.


symbols NFS
crash mover
HBase
MDSW
local FS
Collector N
Processor N
partition cron
Monitor
metrics ADU
middleware hoopsnake
bugzilla
bugzilla cron
mem cache
Web app
ZEUS
Postgres
build ftp
builds cron
top sites
ZEUS
daily crash (ADU)
alexa.com
top crash by sig/URL
daily URL
clean tcbs
service status
sig prod dims
NFS

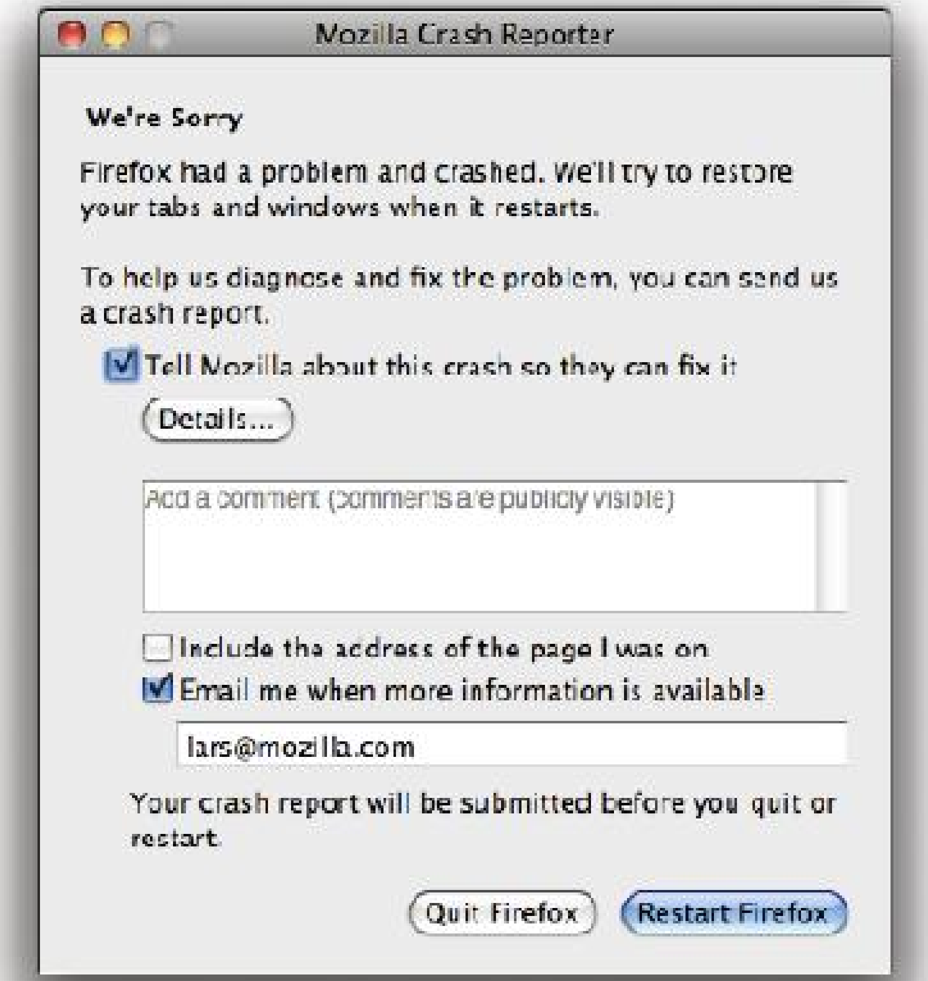
many components
= many failures

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



Mozilla Crash Reporter

We're Sorry

Firefox had a problem and crashed. We'll try to restore your tabs and windows when it restarts.

To help us diagnose and fix the problem, you can send us a crash report.

☑ Tell Mozilla about this crash so they can fix it

Add a comment (comments are publicly visible)

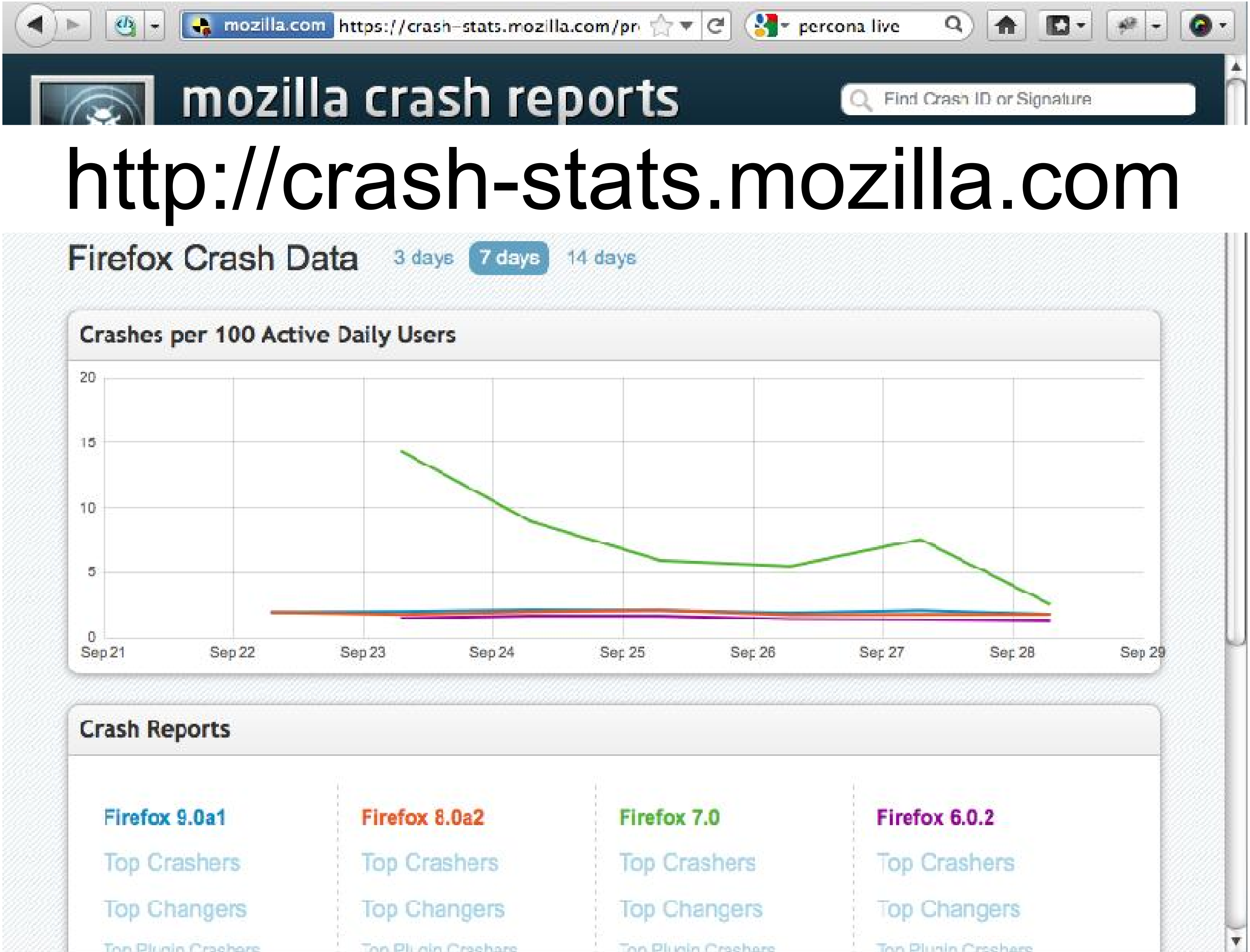
☐ Include the address of the page I was on

☑ Email me when more information is available

lars@mozilla.com

Your crash report will be submitted before you quit or restart.

http://crash-stats.mozilla.com



mozilla crash reports

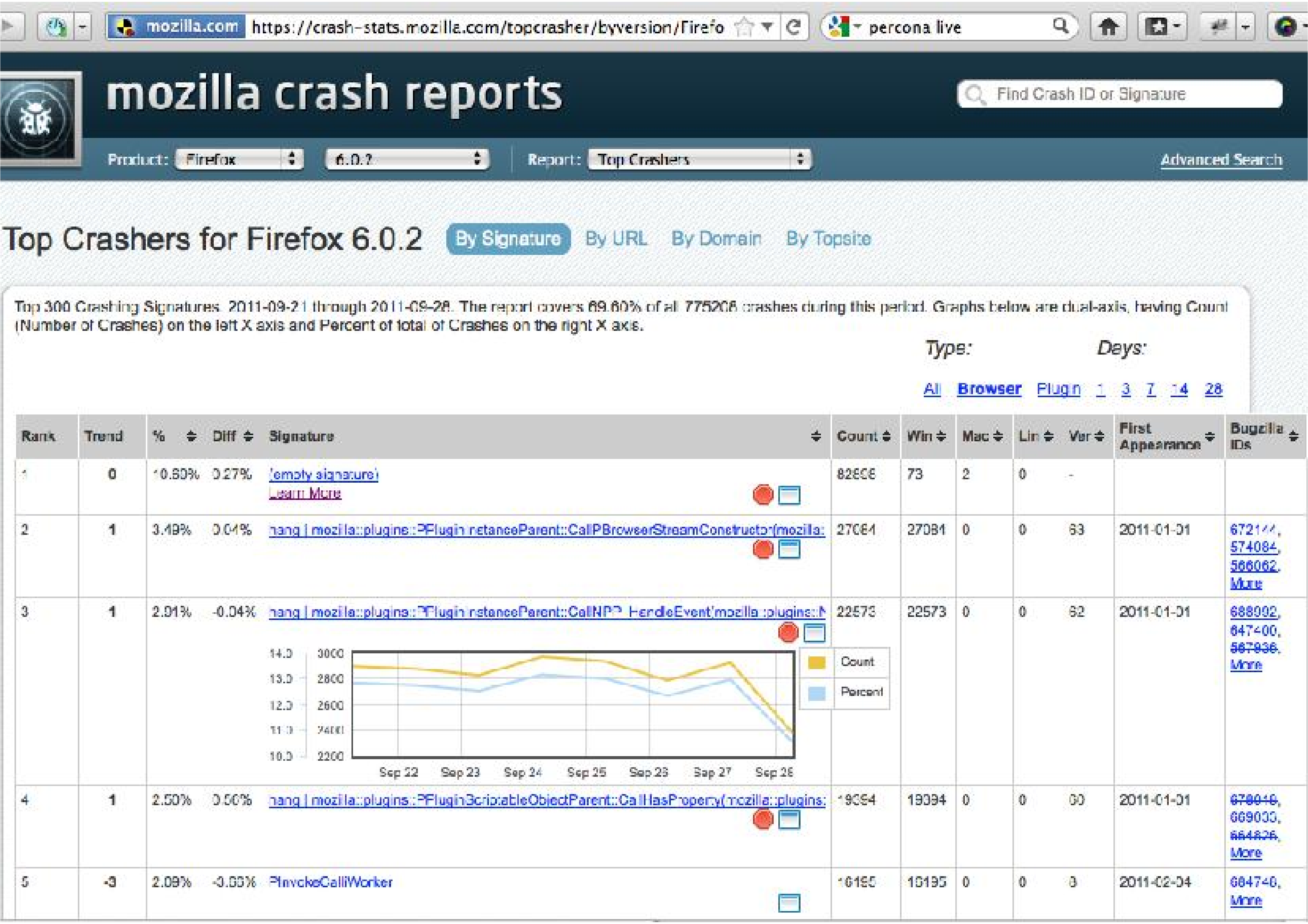
Firefox Crash Data

3 days 7 days 14 days

Crashes per 100 Active Daily Users

Crash Reports

Firefox 9.0a1	Firefox 8.0a2	Firefox 7.0	Firefox 6.0.2
Top Crashers	Top Crashers	Top Crashers	Top Crashers
Top Changers	Top Changers	Top Changers	Top Changers
Top Plugin Crashers	Top Plugin Crashers	Top Plugin Crashers	Top Plugin Crashers


mozilla.com
https://crash-stats.mozilla.com/topcrasher/byversion/Firefo
percona live
Find Crash ID or Signature
Product: Firefox
6.0.2
Report: Top Crashers
Advanced Search

mozilla crash reports

Top Crashers for Firefox 6.0.2

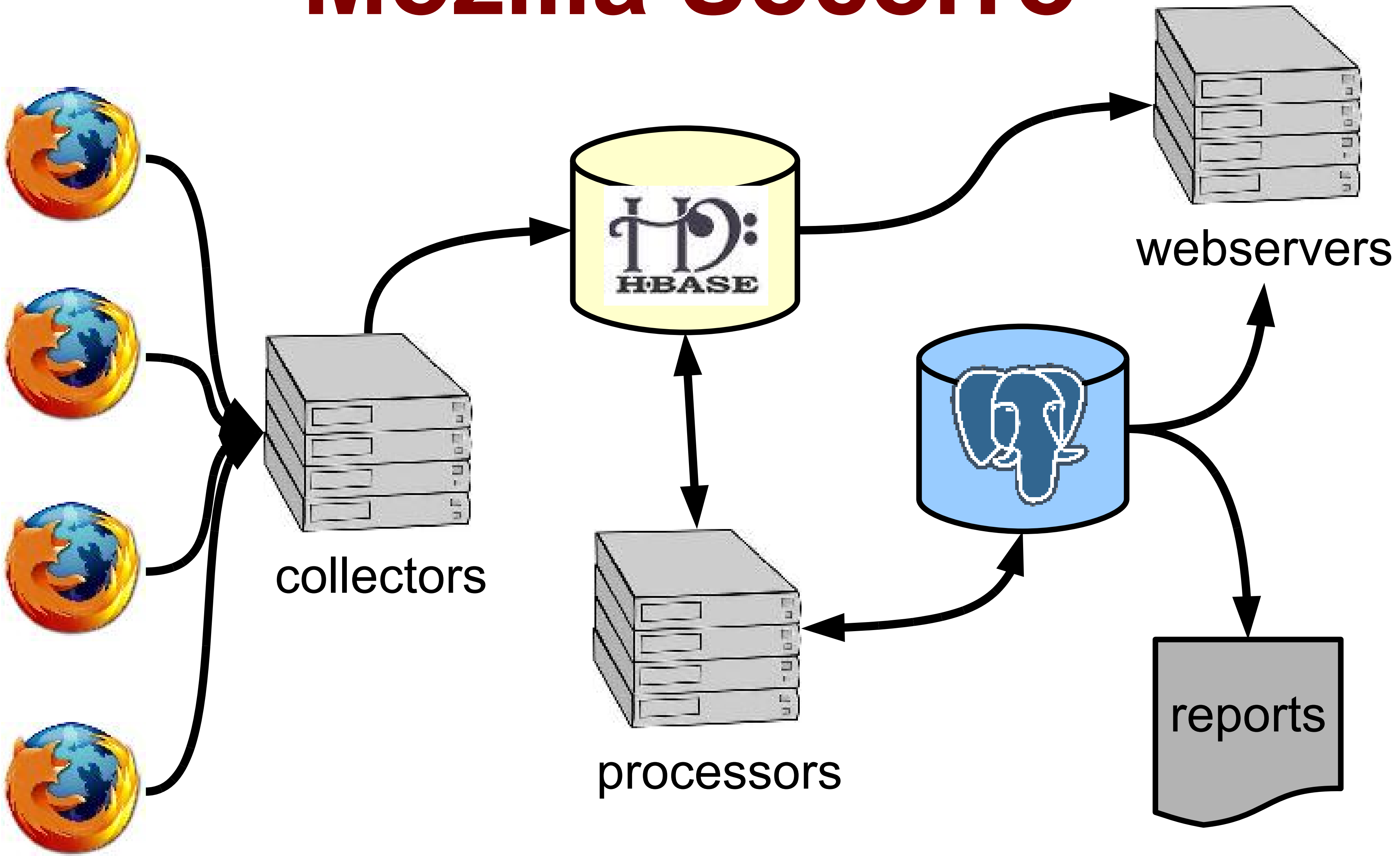
Top 300 Crashing Signatures. 2011-09-21 through 2011-09-28. The report covers 69.60% of all 775208 crashes during this period. Graphs below are dual-axis, having Count (Number of Crashes) on the left X axis and Percent of total of Crashes on the right X axis.

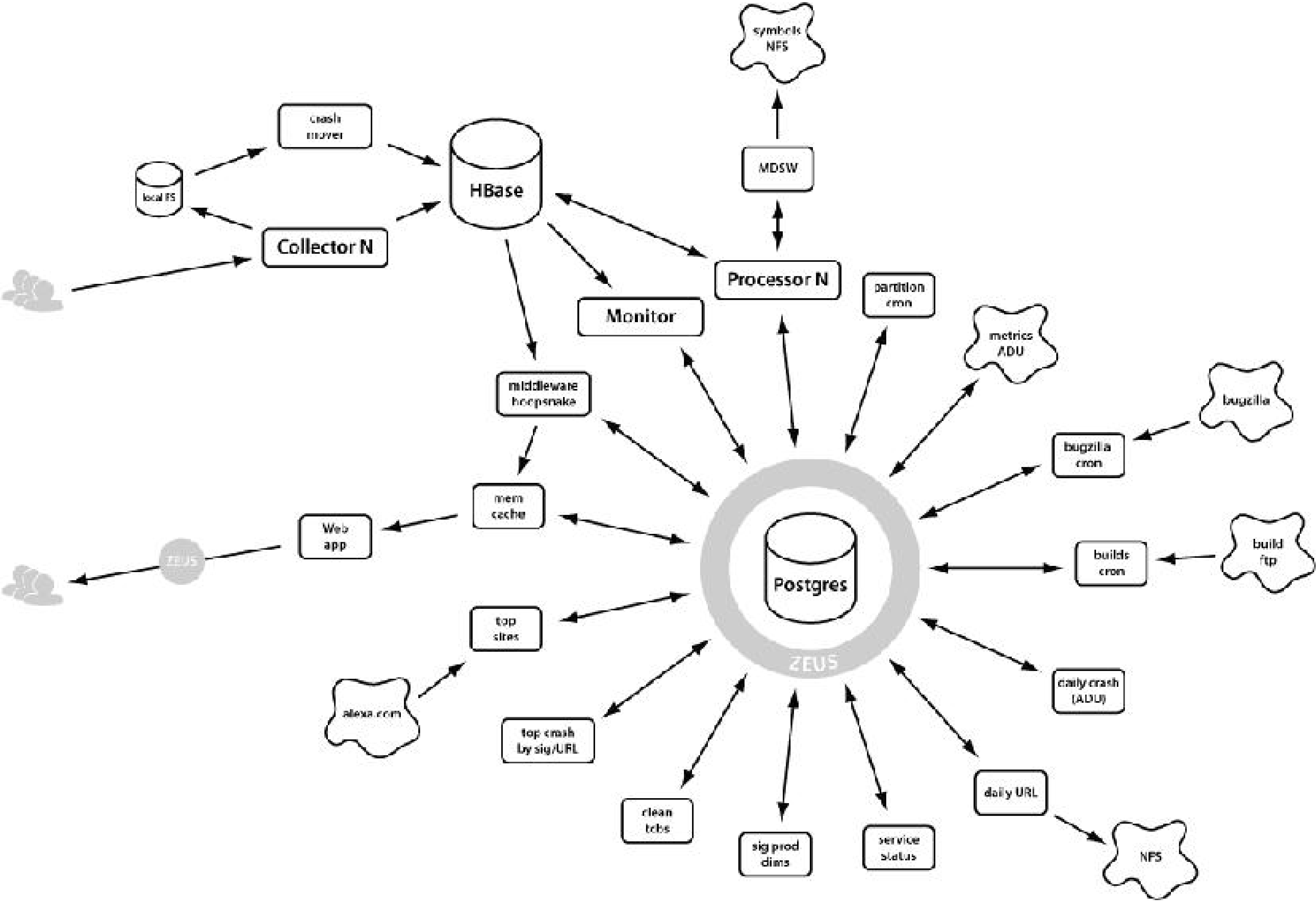
Type: All **Browser** Plugin

Days: 1 3 7 14 28

Rank	Trend	%	Diff	Signature	Count	Win	Mac	Lin	Ver	First Appearance	Bugzilla IDs
1	0	10.60%	0.27%	(empty signature) Learn More	82896	73	2	0	-		
2	1	3.49%	0.04%	hang | mozilla::plugins::PPluginInstanceParent::CallPBrowserStreamConstructor(mozilla:	27064	27064	0	0	63	2011-01-01	672144, 574084, 566062, More
3	1	2.91%	-0.04%	hang | mozilla::plugins::PPluginInstanceParent::CallNPP_HandleEvent(mozilla::plugins::N	22573	22573	0	0	62	2011-01-01	688002, 647400, ~~567936~~, More
4	1	2.50%	0.50%	hang | mozilla::plugins::PPluginScriptableObjectParent::CallHasProperty(mozilla::plugins:	19394	19394	0	0	60	2011-01-01	~~678046~~, 669000, ~~664826~~, More
5	-3	2.09%	-0.00%	PInvokeCalliWorker	16195	16195	0	0	3	2011-02-04	684740, More

Mozilla Socorro

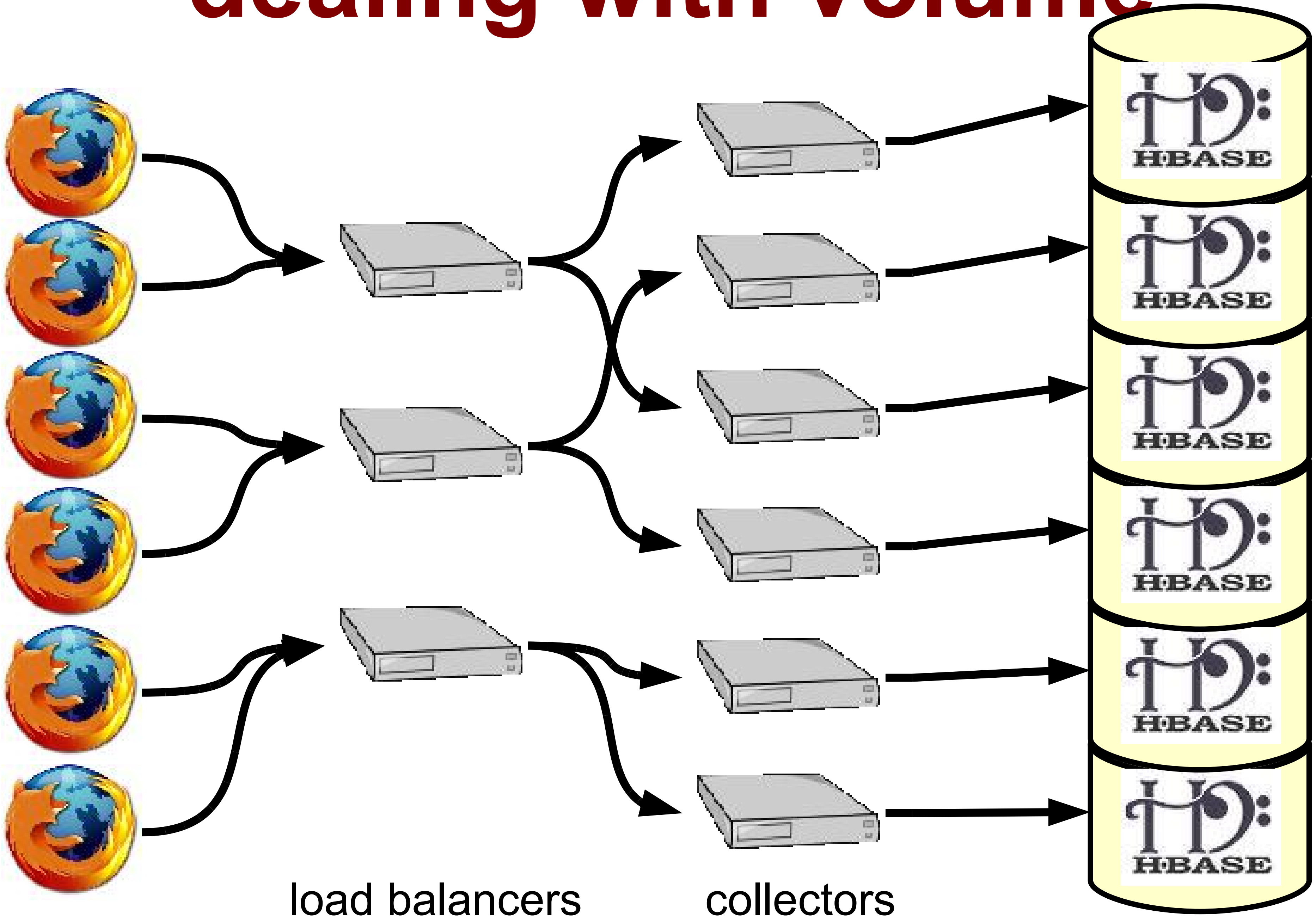



symbols NFS
crash mover
HBase
local FS
MDSW
Collector N
Processor N
partition cron
Monitor
metrics ADU
middleware hoopsnake
bugzilla
bugzilla cron
mem cache
Web app
ZEUS
build ftp
builds cron
Postgres
top sites
ZEUS
daily crash (ADU)
alexa.com
top crash by sig/URL
daily URL
clean tabs
sig prod dims
service status
NFS

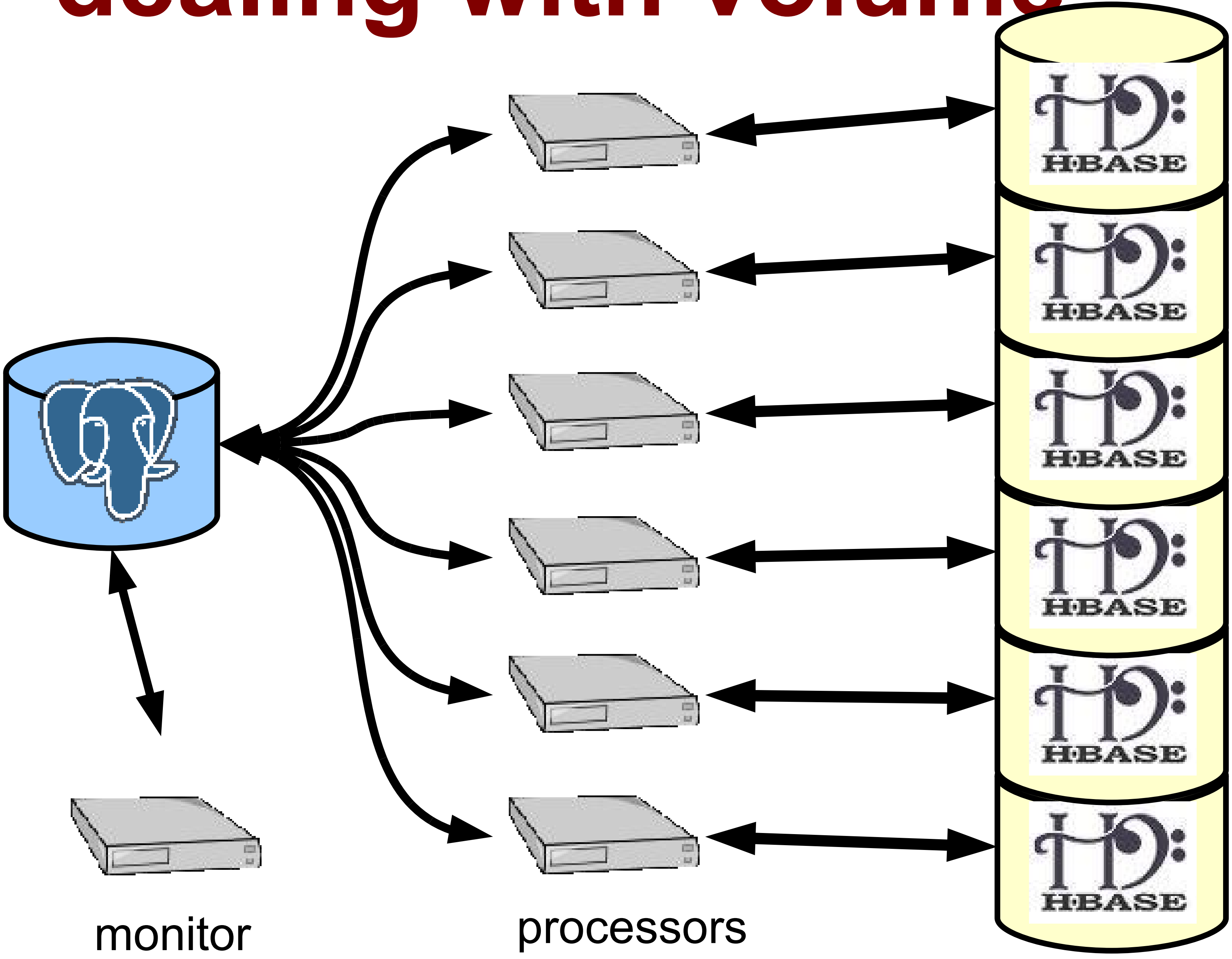
socorro data volume

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

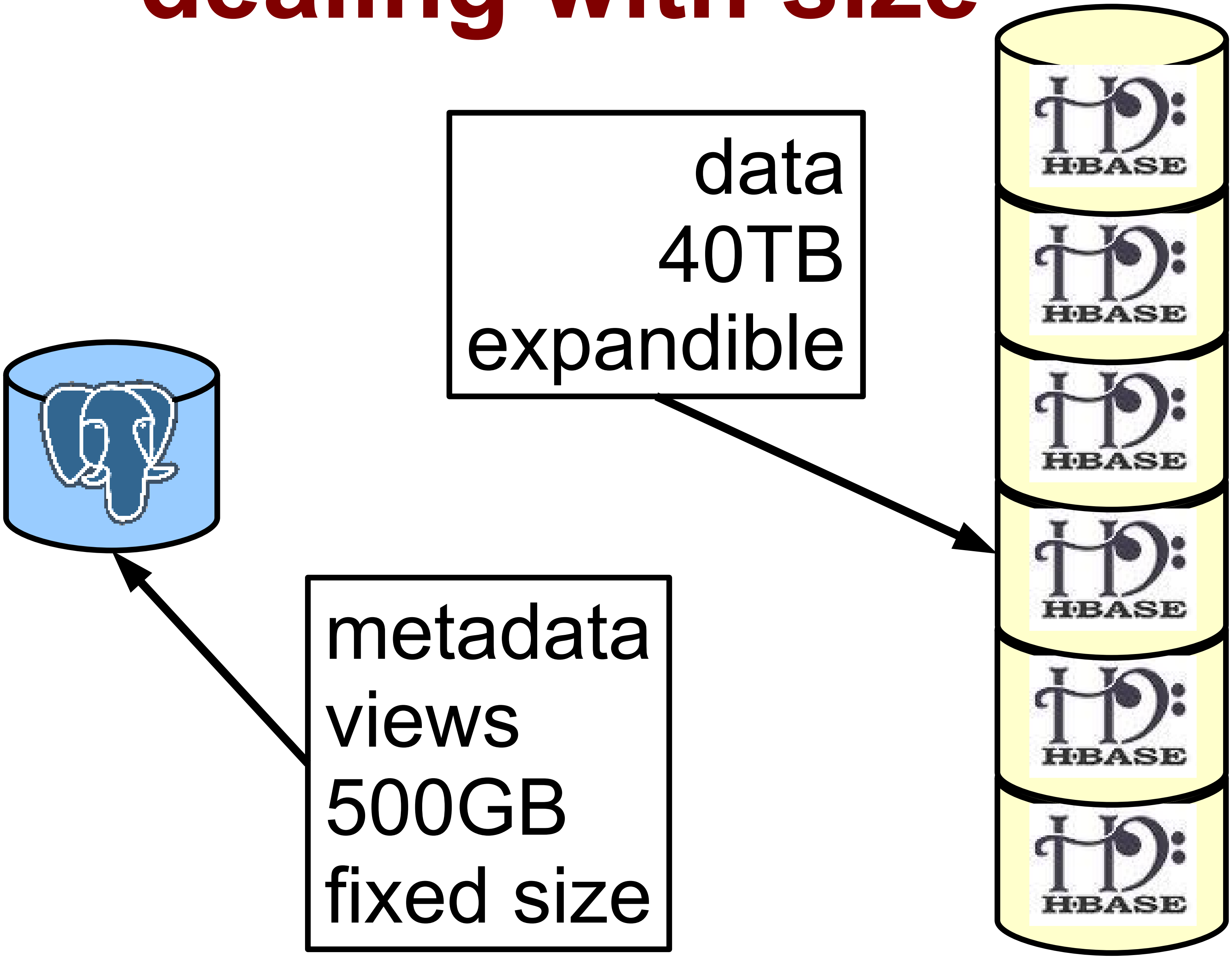
dealing with volume



dealing with volume



dealing with size



data
40TB
expandible

metadata
views
500GB
fixed size

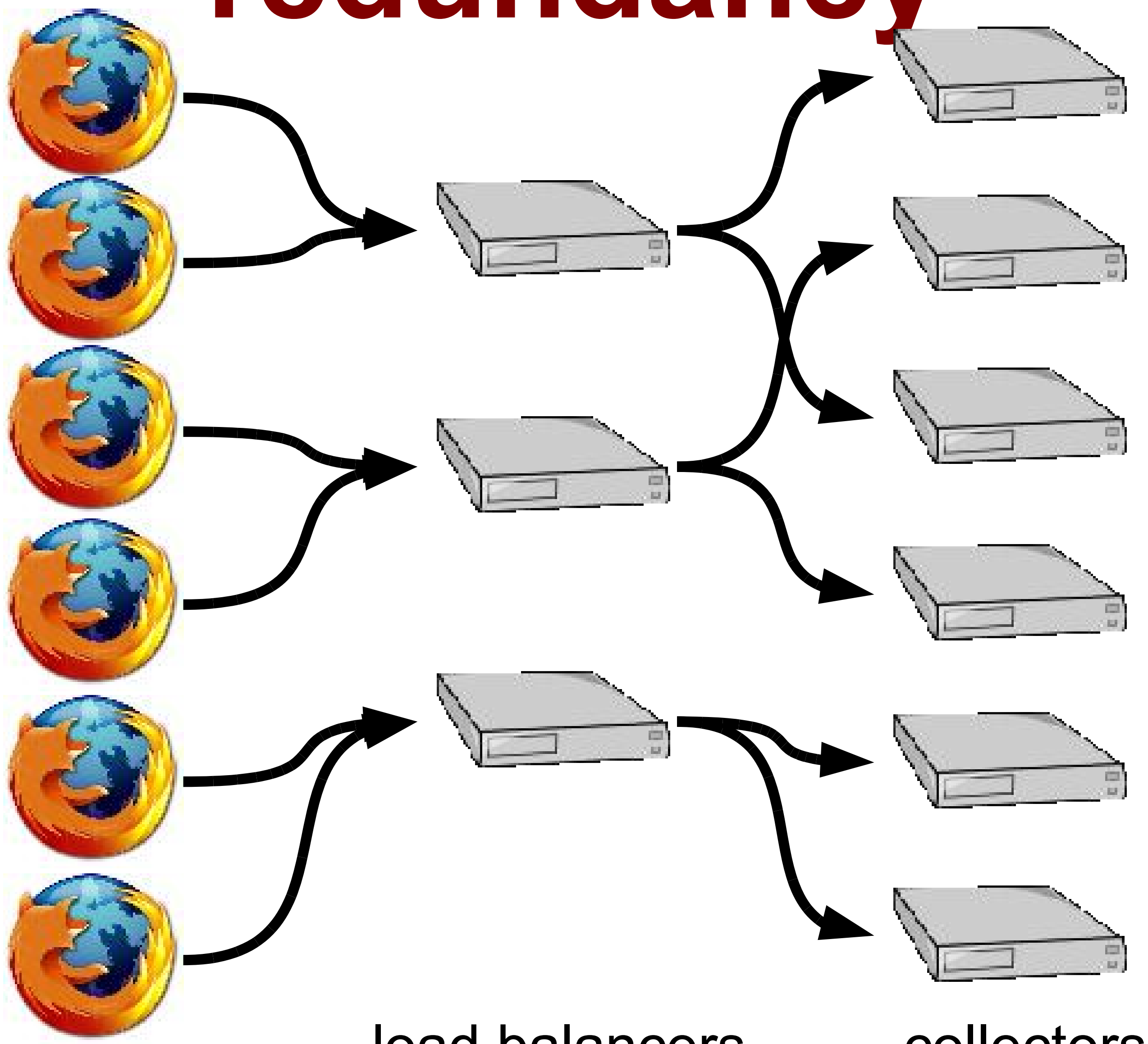
dealing with component failure

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

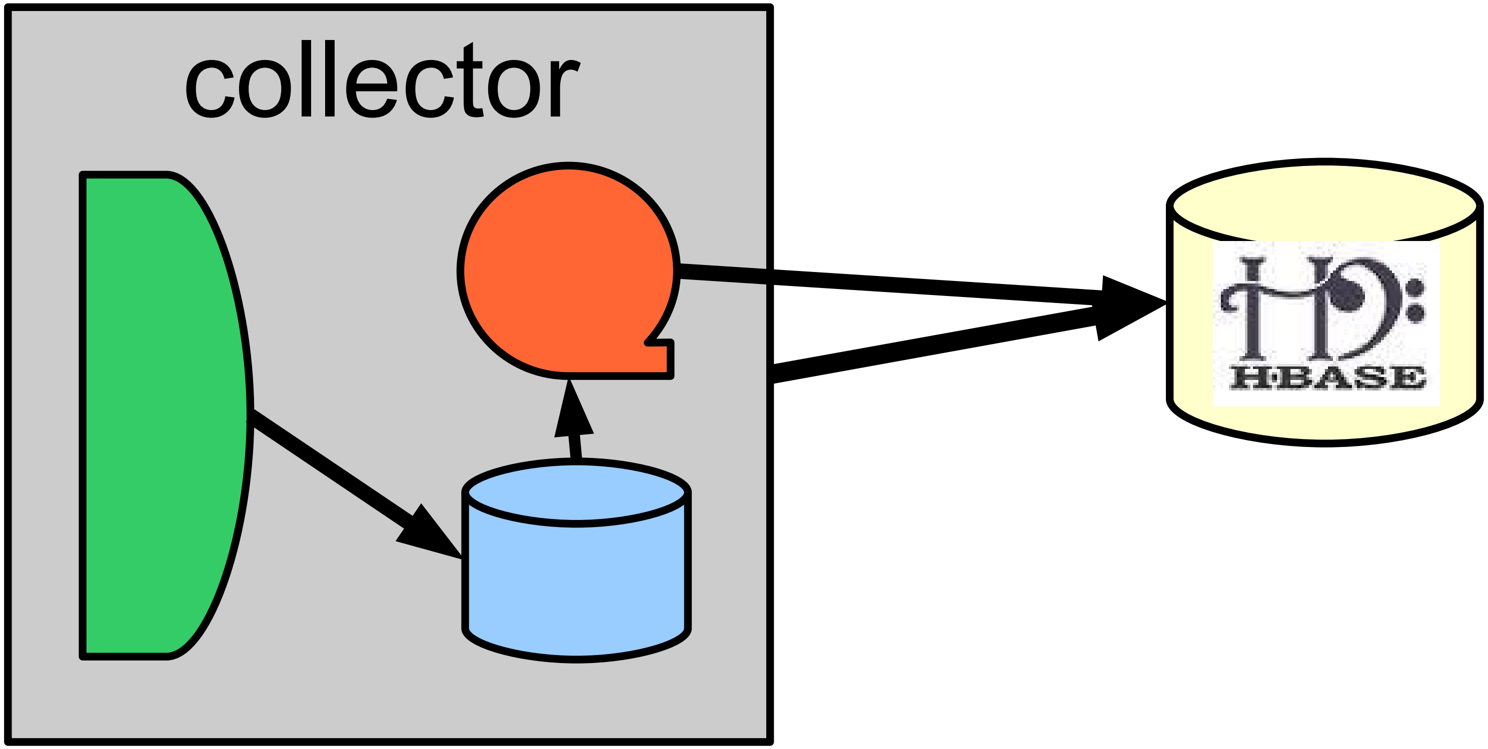


elastic connections

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

elastic connections

crash mover



collector

reciever

local file queue

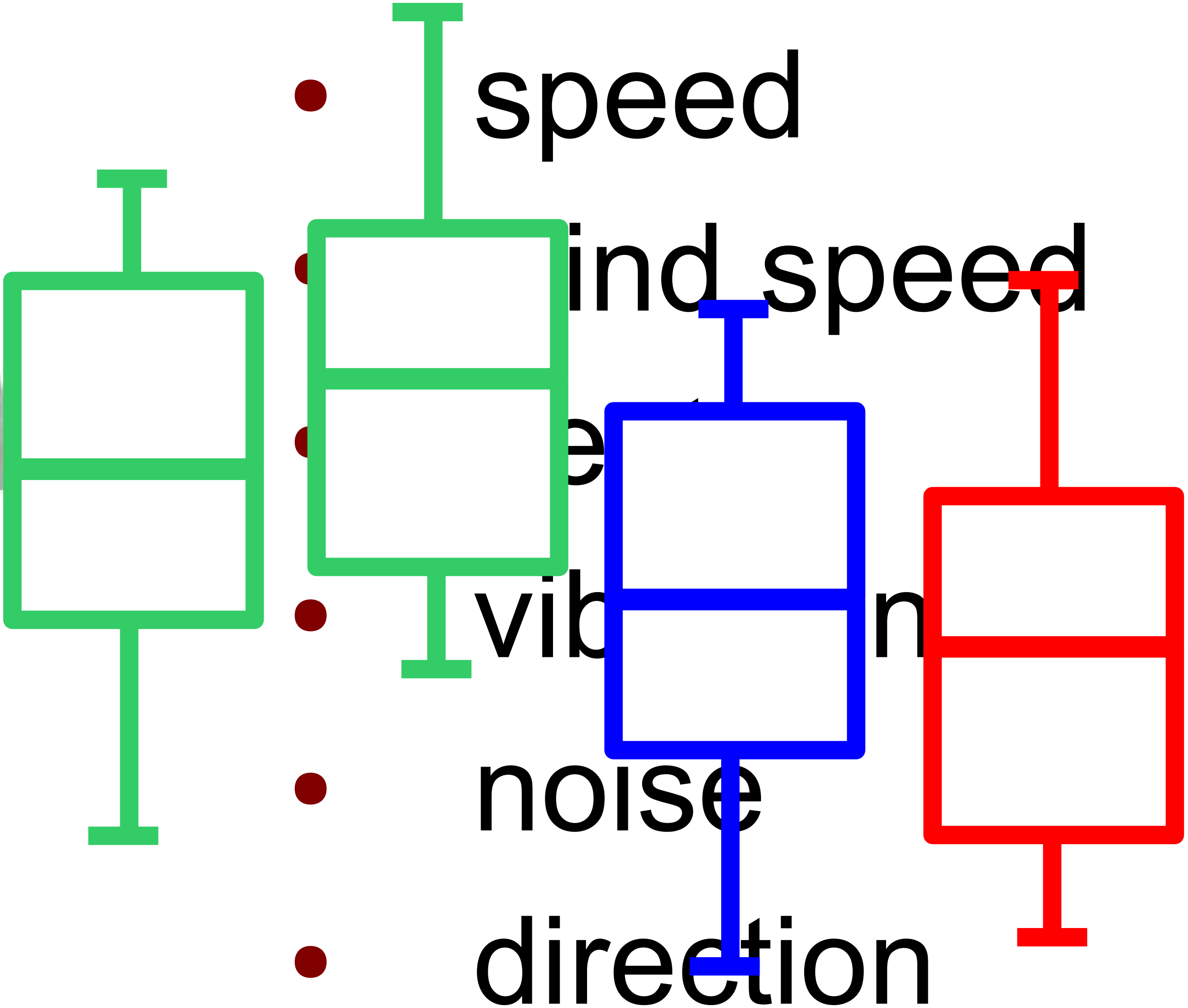
server management

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

Upwind

Upwind

- speed
- wind speed
- e [illegible]
- 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

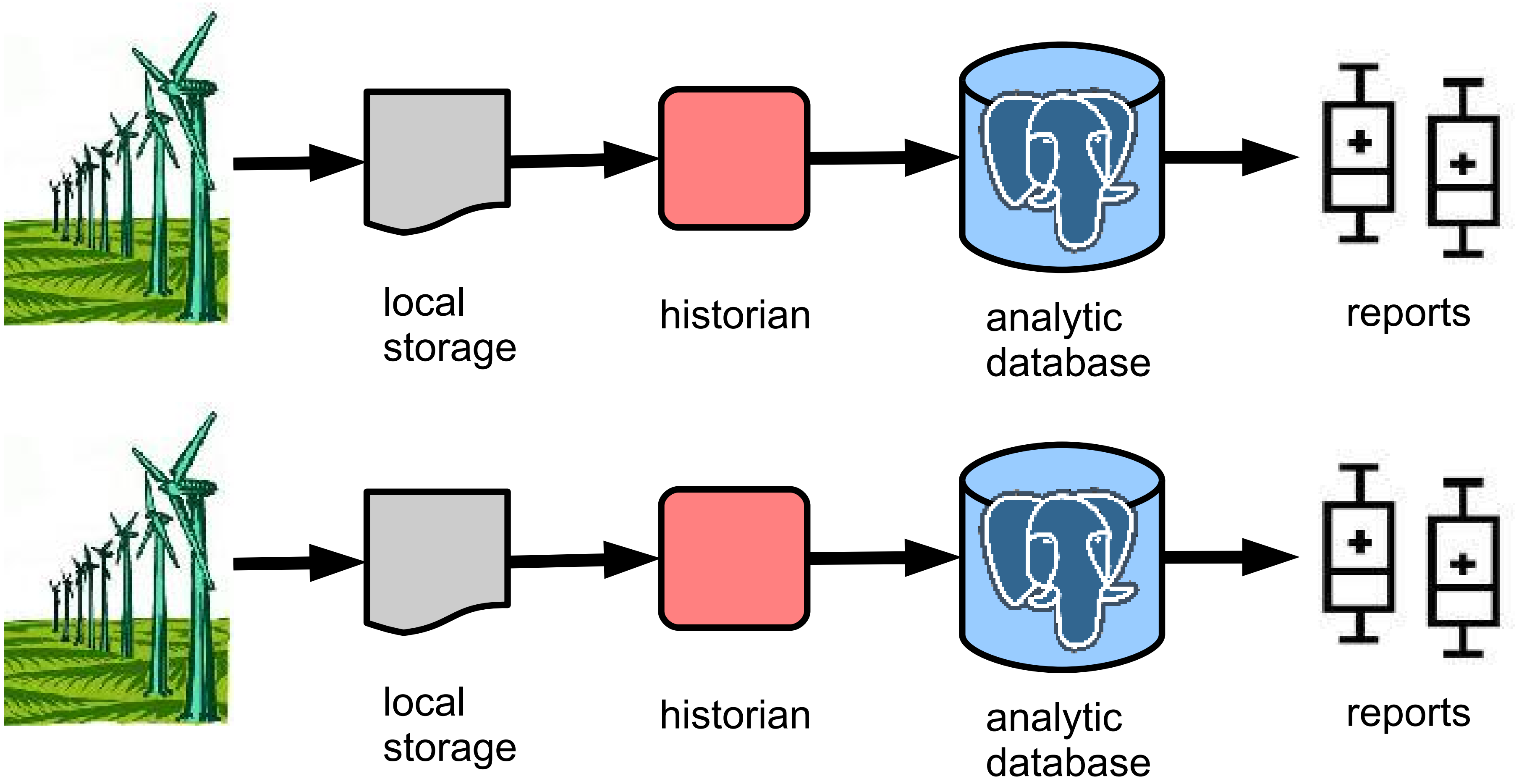
windmills per farm: up to 100

number of farms: 40+

est. total: 300,000 facts/second

(will grow)

dealing with volume



local storage

historian

analytic database

reports

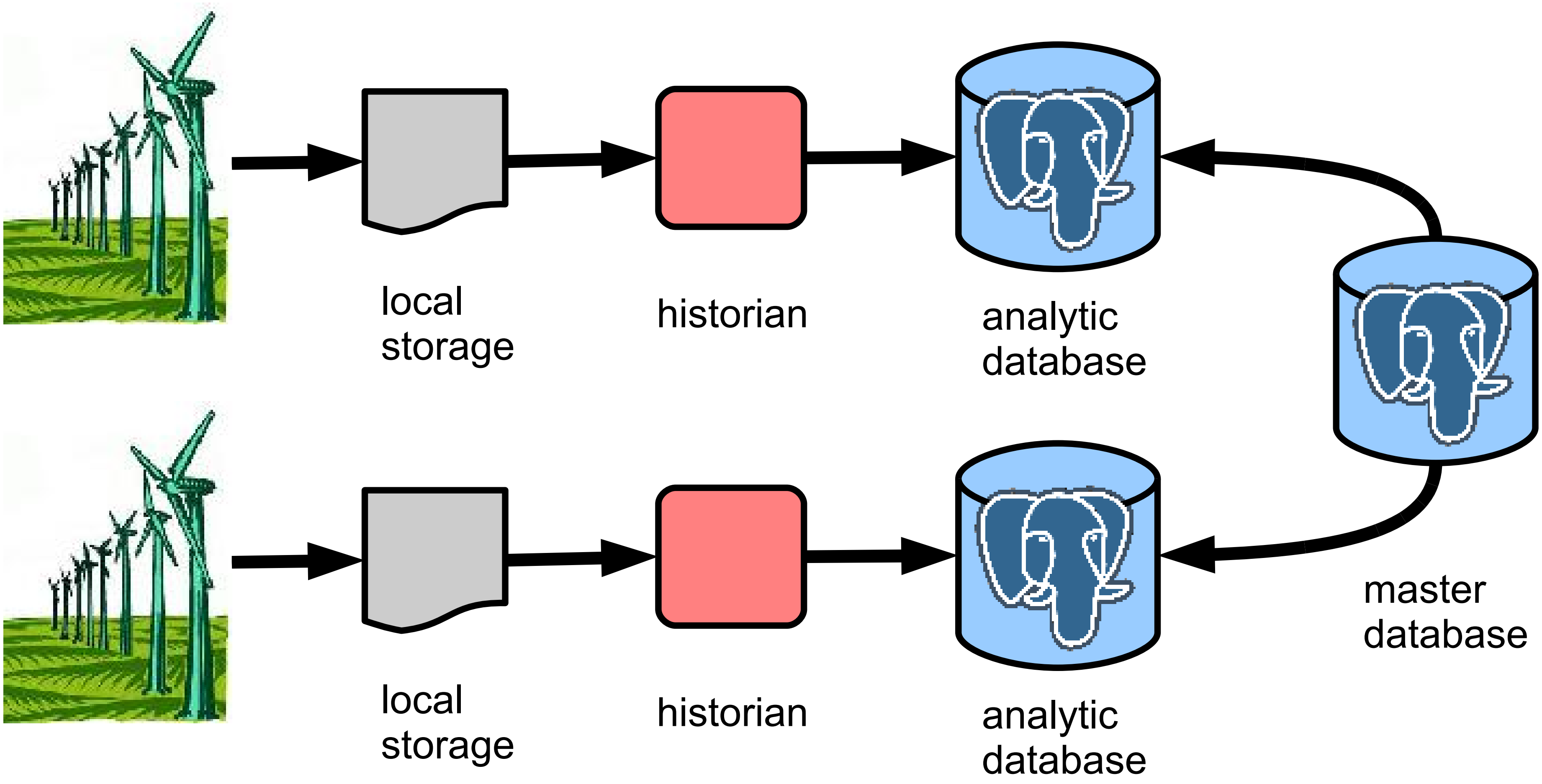
local storage

historian

analytic database

reports

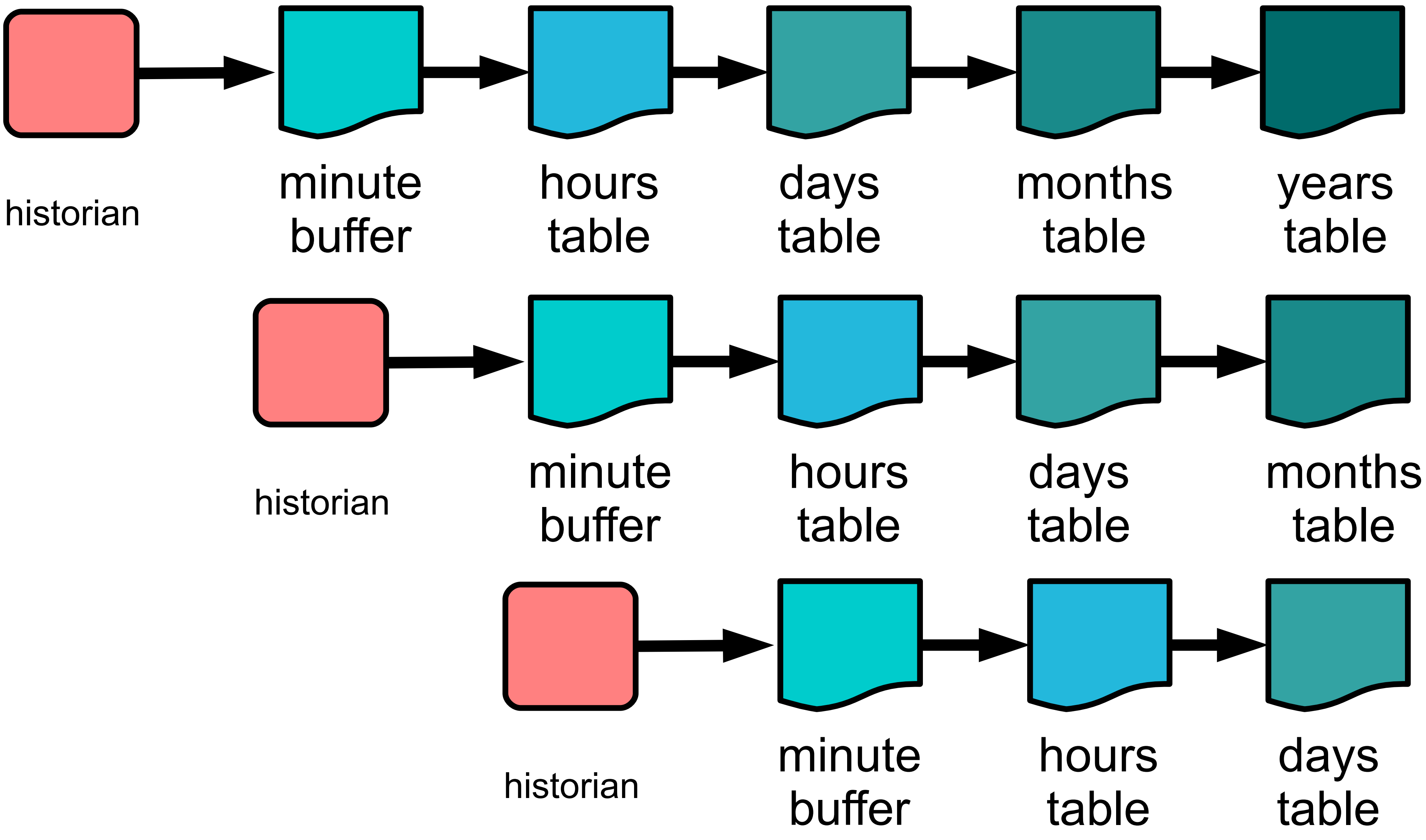
dealing with volume



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/