The Future of DSC

OARC/DSC Workshop

Denver

2010-10-15

Schedule

- Part I
  - Background/History
  - Design review
  - Usage case studies
- Part II
  - DSC's flaws
  - Design/feature wishlist
- Part III
  - Putting a plan in place?

Background/History

- Born from a conversation between Duane And P. Vixie in 2004
- Briefly named “dnsstatd” 😐
- Designed for OARC and for Root servers
- Mostly written by Duane
- Some significant enhancements from Ken Keys
- Numerous patches from others

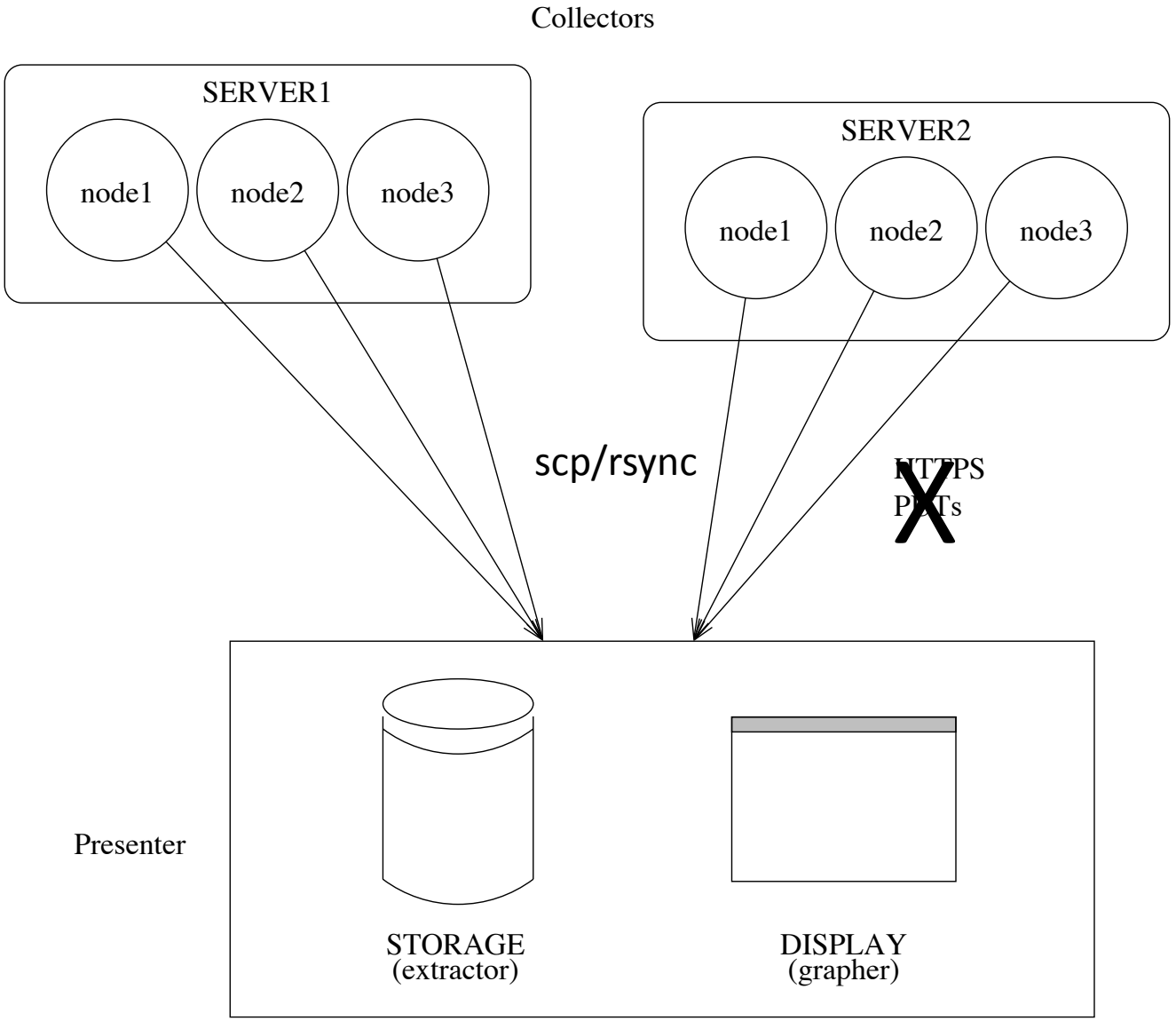
Property Rights & License

- BSD license
- Joint ownership between The Measurement Factory (TMF) and Internet Systems Consortium (ISC).
- TMF offered to assign its rights to DNS-OARC
  - Hoping that its development and support would continue

Terminology

- Collector : The part that captures DNS packets on/near a DNS server
- Presenter : The part where data is permantely stored and can be viewed.
- Extractor : A subset of the Presenter that processes incoming data
- Server : A collection of Nodes
- Node : A component of a Server

Architecture



Implementation

- Collector in C/C++
- Presenter mostly Perl and /bin/sh

Part II

Discussion Topics

Keep Collector Simple?

- keep lightweight
- like to reconfigure on the fly
- maybe use spare cycles for pre-processing
- Need to be able to use a single process to collect lots of different “nodes” (by IP address or query name) and share different nodes with different presenters.

Datasets

- improve interface to choosing datasets (GUI)
- Amount of data sent between C/S not generally an issue
- improve support for list of authoritative domains (ie, for the non-root, non-TLD cases)

Ditch XML?

- JSON?
- YAML?
- maybe okay to keep XML if processing were faster (not Perl)

Data Storage

- Should have choices of backends and an API
- Cassandra?
- SQLIte?
- Need to benchmark
- presenter may scale better with non-FS backend

Presentation Technologies

- AJAX?
- Javascirpt?
- see Nfsen
- make it easier to select different time scales

Views/ACLs

- Do not put authentication in DSC
- but pull username from HTTP auth
- remove some interactivity for some users

Long Term Archiving

- Current archiving is acceptable
- does not require unreasonable amounts of disk space

Dynamic Configuration

- push configs from a presenter (or other central entity) to collectors
- maybe using XML
- GUI to select choices (datasets)
- Define new datasets by cloning + add filter
- loadable modules for filters, and datasets
- Default/base configs for common environs (TLD, root, resolver, etc)

Attack Resilience

- DSC may not be able to “keep up” when a nameserver is under attack
- Should it prioritize its processing
- ie, drop some datasets when out of CPU/ memory?

Anomaly Detection?

- API into datastore so third party tools can do detection
- Nagios triggers?
- Alert when presenter not receiving data from collector

SIE Integration?

- SIE integration is not seen as a priority

60 Sec Window

- No reasons to change this

Node/Server -> Tree

- A tree structure allows for better representation of complex systems such as TLD, site, node, ???
- “View” feature should be able to restrict level of detail exposed

Presenter -> Presenter

- useful with ability to filter/anonymize
- aggregators
- if not filtering then maybe easier to send directly from collector to different presenter

make it faster

- The presenter, especially, needs to be faster.
- Less code in Perl
- Faster backend storage

improved documentation

- DSC’s documentation should be improved