NNTSC: A Storage Backend for Network Measurements

Shane Alcock
Network Measurement Eco-System

My cool new measurement tool
Network Measurement Eco-System

My cool new measurement tool

Scheduling

Data Storage

Accessibility

Visualisation
Motivation

• Creating the whole eco-system takes time and skill

• Instead, prototypes get released so we end up with:
  ○ User-unfriendly tools
  ○ Lack of scalability
  ○ No maintainability
Motivation

• Stop rewriting the whole eco-system from scratch
  ○ Design systems that we can reuse and extend
  ○ Refined implementation
  ○ Standardised deployment
  ○ Spend more of our time on cool new tools!

• For this talk, I'm only going to focus on storage
The Idea

• Many network measurements are time series
  ○ Set of common defining parameters
    • Source, target, packet size, protocol, port
  ○ Regular frequency
  ○ Result is a series ID + timestamp + value

• Can we build a unified system to store and access any type of time series data?
Design

• Develop generic core to handle all common actions
  ○ Database inserts and queries, client management
  ○ Wrap around an existing database system

• Modules to define behaviour for specific collections
  ○ Table structure, result parsing
Design

• Historical data access
  ○ Static graphs, repeatable analysis, data download
  ○ Aggregated data vs full resolution data

• Live data access
  ○ Anomaly detection, streaming graphs
  ○ Often an afterthought
Databases are Hard

• Important to choose the database backend carefully!
  ○ We sank a lot of time in solving performance issues
  ○ Happy to share some war stories later :)

• Better options are available today (probably)
  ○ Research, don't just fall back to what you know
  ○ Beware the hype, run your own tests
“Disk Space is Cheap”

• Aim was to store full unaggregated data
  ○ Contrast with RRD – low resolution historic data

• Buying TBs of new disk space was not the problem
  ○ Adding more disk to a running system, however...

• Estimate your future disk usage
  ○ Small storage savings add up over months
Make Sure You Can Scale

- Test and evaluate at scale
  - Many systems look great when you prototype
  - Fall apart as soon as you scale up to production
  - Testing with 1000s of series, months of results
Keep Everything Moving

• Component independence
  ○ Delays in one component shouldn't block others
  ○ Think parallel from the start
    • Split tasks amongst CPU cores
  ○ Use message queues for buffering
    • RabbitMQ is great
    • Python multiprocessing Queues for small jobs
The Best Lesson

• NNTSC works as intended!
  ○ Easy to extend by adding new collections
  ○ A core part of our ongoing research projects

• AMP
  • Network anomaly detection
  • Passive layer-7 statistics
  • Cloud security monitoring