RISng - Technical Overview

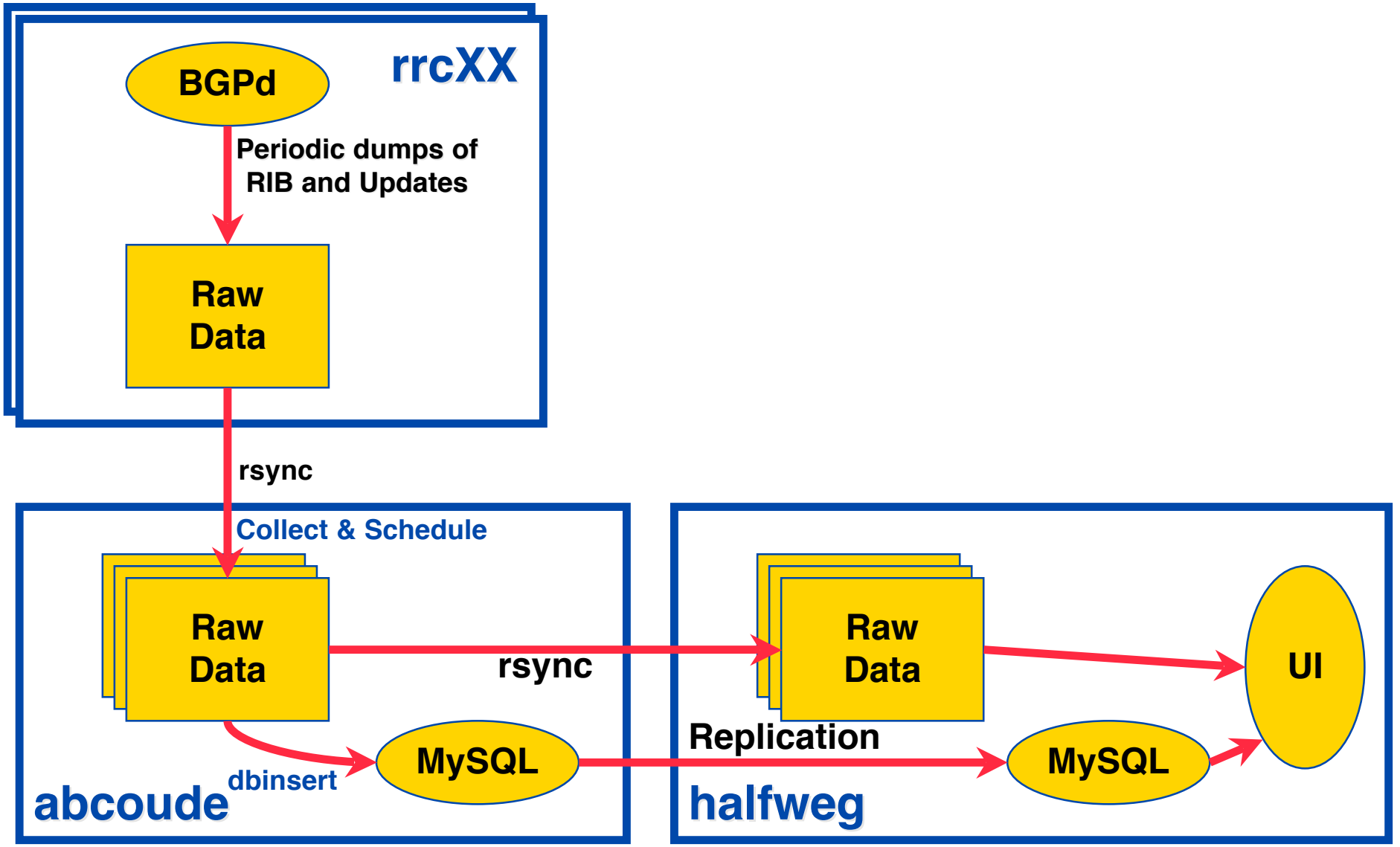
James Aldridge, Daniel Karrenberg, Henk Uijterwaal, Arife Vural, Matthew Williams

RIPE NCC New Projects Group

Contents

- “RIS Classic” - The original version
- Problems
- Potential solutions
- “RISng”
- Work in progress
- Summary
- Questions

“RIS Classic” - Overview



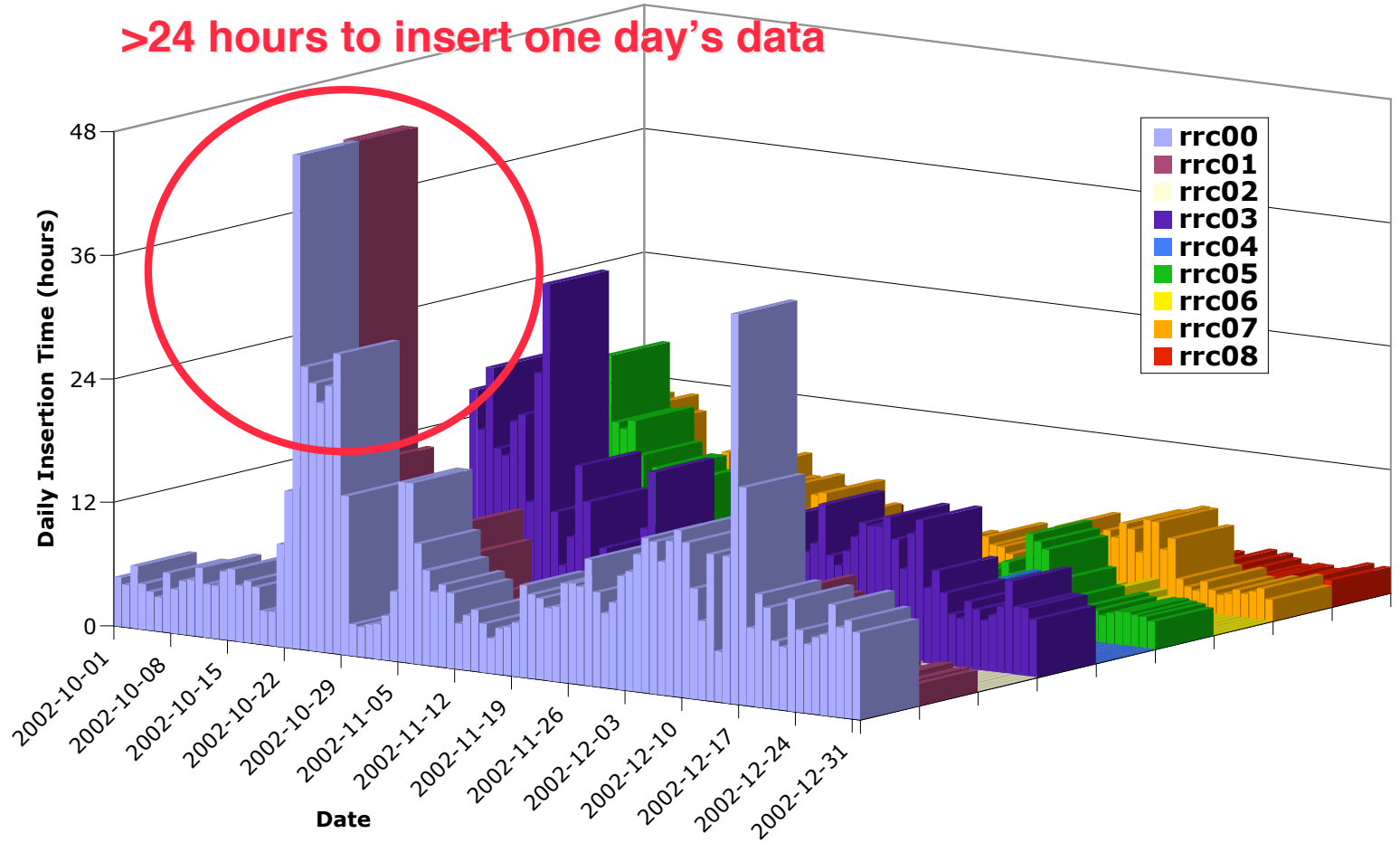
“RIS Classic”

- RRCs only run Zebra bgpd
  - All other processing done centrally
- All database insertion done centrally
  - abcoude.ripe.net
  - Solaris / Sun SPARC Ultra Enterprise 420R
  - MySQL insertion script written in Perl
    - Text output from “`route_btoa -m`” (Update dumps)
    - Binary processing (RIB dumps)
- Front-end server handles user interface, etc
  - halfweg.ripe.net
  - Linux
  - Replicated copy of database

Problems

- Database insertion of data from 9 route collectors on a single central machine is slow
  - Little headroom to allow for abnormal cases
  - Can sometimes take more than 24 hours to insert a single day's data
  - Little capacity to add more RRCs or full BGP feeds
- Limited attributes are stored in the database:
  - Only first 255 characters of AS Path stored
  - Other BGP attributes (communities, MEDs, etc.) ignored

“RIS Classic” Database insertion times



Considered Solutions

- New, faster hardware to replace or supplement abcoude.ripe.net
- Make better use of existing hardware
- Database redesign
- Use other database than MySQL

- But we don't want to spend more money than necessary

- RISng is the result…

“RISng”

- Aims
  - Improve scalability
  - Easier software maintenance
  - Store more complete route attributes
- New database structure
  - Remove arbitrary limit on AS Path
  - Store additional attributes
- Perform database insertion locally on (otherwise mostly idle) route collectors
- New database insertion process
  - written in C instead of Perl

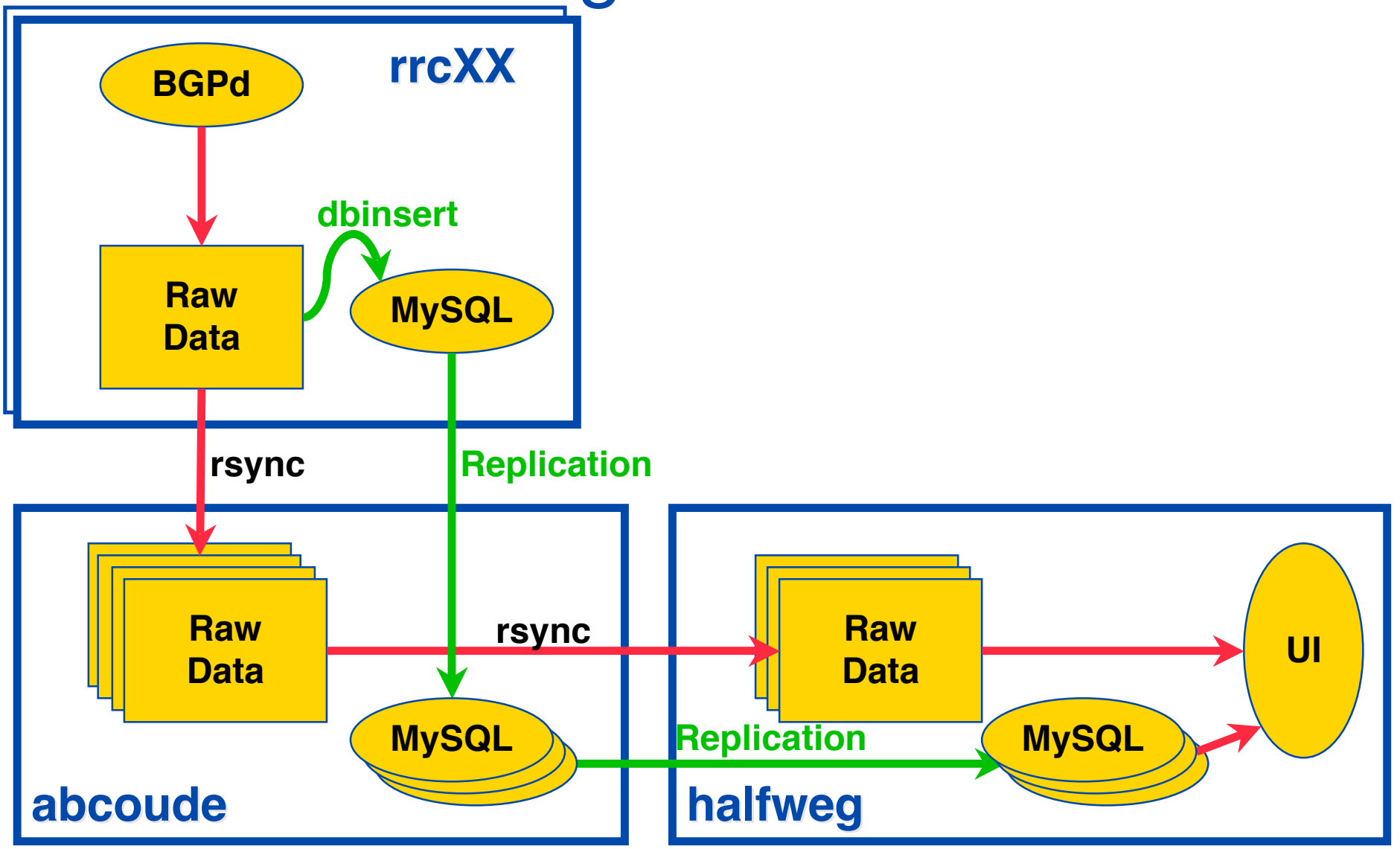
Single software version

- Up to now, software maintenance has been more difficult than necessary
  - 6 different versions of FreeBSD (FreeBSD 3.5 onwards)
  - Almost every individual RRC needs its own software build
- Aim to bring all RRCs to the same OS version (FreeBSD 4.6.2)
  - Bootable CDROM created and shipped to hosts
    - Contains a snapshot of complete RRC system
    - Allows remote (SSH) access for manual configuration and subsequent maintenance if necessary
  - Upgrades are in progress (3 out of 9 RRCs upgraded so far)
    - But some boxes have problems:
      - RRC01 (@ LINX) fails to boot the new FreeBSD kernel
      - RRC04 (@CERN) has a faulty CDROM drive
- Subsequent software updates can use rsync

RISng Overview

- We take advantage of unused processor resources in each route collector:
  - Data is now inserted into a local database on each RRC and replicated to central servers
- Software changes
  - Zebra BGPd
  - Insertion process

RISng - Overview



RISng - Changes to Zebra BGPd

- Normally Zebra dumps RIB and Updates at times relative to when the process started
- We would like to have dumps at fixed times of the day to make it easier to compare data between RRCs
  - RIB dumped at 00:00, 08:00, 16:00 hours
  - Updates dumped at xx:00, xx:15, xx:30, xx:45
- Small change made to Zebra's `bgpd/bgp_dump.c` to achieve this
  - First dump starts immediately after start-up as before
  - Subsequent dumps follow the above pattern

RISng - New Database Insertion Process

- Rewritten in C
- Runs locally on each RRC
  - Processing time not influenced by what load other RRC's data insertion may be generating
  - Local database insertion removes delay caused by once hourly rsync of dump files to central machine
- Database is replicated to central servers
  - Simplifies porting of front-end applications to use new database

New Database Structure

- Arbitrary length restrictions removed
- Many new attributes stored
  - More information allows better diagnosis of routing problems
  - RIS users can now see whether what appeared to be duplicate announcements really are identical or whether some other attribute (MED, Community, etc.) has changed.
  - Prevents time being wasted tracking down the wrong problem.

New Database - Prefix Table

	RIS Classic	RISng
Prefix	Prefix string	
Start	First address in prefix range	
End	Last address in prefix range	
First time seen	Timestamp	
Last time seen	Timestamp	
IP version		New
Status	Never used	Removed
RRC	RRC Number	
Origin AS		New

New Database - AS Path Table

	RIS Classic	RISng
AS Path	255 characters	Unlimited
First time seen	Timestamp	
Last time seen	Timestamp	
Status	Never used	Removed
Length		New: Length of "raw" BGP AS Path attribute
RRC	RRC Number	

New Database - Peer Table

	RIS Classic	RISng
IP Address	Peer IP Address	
AS Number	Peer AS Number	
Status	Peer Status (up or down)	
IP version		New
First time seen	Timestamp	
Last time seen	Timestamp	

New Database - Attributes Table

All New!

	RISng
Origin Type	‘IGP’, ‘EGP’, ‘Unknown’
Next Hop	IP address
MED	Multi-Exit Discriminator
Community	Unlimited length string
IP Version	4 or 6
“Rest”	Any other attributes we choose to store
RRC	RRC Number
First time seen	Timestamp
Last time seen	Timestamp

New Database
RIB & Updates Tables

	RIS Classic	RISng
UTC	Timestamp	
Prefix ID	Reference to entry in prefix table	
AS Path ID	Reference to entry in AS path table	
Attribute ID	Reference to entry in Attributes table	
Type	'U' (Advertisement) 'W' (Withdrawal) 'S' (BGP state change)	
RRC ID	RRC Number	
Origin AS	Origin AS ('U' entries only)	
IP version	IPv4 or IPv6	
Peer ID	Reference to entry in Peer table	
Pstate	Not Used	
Nstate	Not Used	

RISng - Ongoing Work

- Improve handling of RIB dump
  - RISng insertion is currently slower than before…

    … but we don't use any of the Perl tricks to bypass MySQL which the "Classic" insertion process uses
  - Better indexing of database?
  - Intermediate processing of RIB dump followed by single pass to update MySQL tables?
  - Is MySQL the best way of handling this?
- Store even more attributes in the database
- Store IPv6 entries in the database

Credits

- Dan Ardelean for providing the `libbgpdump` building block for reading Zebra's BGP dump files.

- RRC Hosts for hands-on support during the sometimes problematic software upgrade process

Summary

- Scalability and software maintenance issues improved
- More attributes stored in database
- More work still to do…
- Test drive the new database:
  - http://www.ris.ripe.net/risng/

Questions?