Icinga 2.11

CW18 2019 | NETWAYS | Nuremberg

Network Stack

Problem Analysis

https://github.com/Icinga/icinga2/issues/7041

01
Connections

The current thread pool implementation limits fast connection handling, has asynchronous TLS event polling (blocking), and custom fifo buffers – many problems under the surface.

02
Context Switches

2.9: Use 1 thread per connection, lots of resources, lock waits, switches
2.10: Thread pool for resource management, everything is slower but less resource consuming. Still blocks.
2.11: Find a solution for all the problems since 2.0.

03
HTTP Server

HTTP Server is written from scratch since 2.4
Many connections stall and overload the monitoring system
Long lasting queries are cut off (no detection of waiting clients)

Network Stack

Boost library – everything not in STD library

01
Boost

02
I/O Engine

03
HTTP API

- Raise Boost to 1.66+
- ASIO – Network I/O, TLS abstraction
- Beast – HTTP
- Coroutines – non-blocking threads

Network Stack

I/O Engine – Connections & CPU bound work

01
Boost

02
I/O Engine

03
HTTP API

- FAST async connection handling
- CPU bound work in the background
- IO bound work in the foreground
- Use coroutines to avoid locks
  - Stores function frame on stack
  - Continues when there are resources available
  - https://www.netways.de/blog/2019/04/04/modern-c-programming-coroutines-with-boost/

Network Stack

HTTP Server – REST API

01
Boost

02
I/O Engine

03
HTTP API

- Boost Beast
  - Async reads/writes (header, body)
  - Buffers
- HTTP verbs
  - *http::status::ok*
  - *http::field::accept*
  - *http::verb::options*

Network Stack

Rewrite core parts: The story.

https://github.com/Icinga/icinga2/issues/7041

01 Boost

Boost 1.66+ allows the usage of additional libraries for socket/network I/O, thread pools and HTTP server/clients.
Package Boost on platforms which don't have this in EPEL/Backports.
Status: Done

02 I/O Engine

Replace the current TLS socket I/O implementation with custom event handling (poll, epoll) with Boost ASIO.
Use IoBoundWork and CpuBoundWork thread pools.
Status: Done

03 HTTP API

Replace custom HTTP handling with Boost ASIO & Boost Beast.
Use Beast Buffers, HTTP verbs and more things for compile time errors, not runtime.
Replace HTTP Clients (InfluxDB, Elasticsearch, CLI commands, check_nscp_api) with Boost implementation.
Status: 80% done. TODO: console, check_nscp_api, logging, docs

Icinga 2.11

More goodness

01
HA & Failover

02
Configuration

03
Runtime Objects

- Feature HA

  https://github.com/Icinga/icinga2/issues/2941

  - Elasticsearch, Graphite, InfluxDB, etc.
- Failover in HA zones
  - Object authority update every 10s (was 30s)
  - DB IDO failover_timeout 30s (was 60s)
  - More logging
- Status: Done

Icinga 2

More goodness

01
HA & Failover

02
Cluster Config

03
Runtime Objects

- Story
  - https://github.com/Icinga/icinga2/issues/6716
  - Coming from #10000 😜 😜 😜 😜 😜
- Tackle existing problems
  - Staged sync, no broken config after restart
  - Don‘t include deleted zones on startup
  - Deal with race conditions on sync
- Status: PoC PR exists

Icinga 2.11

Runtime Objects in API config packages

01
HA & Failover

02
Cluster

03
Runtime Objects

- Story: https://github.com/Icinga/icinga2/issues/7119
  - Runtime objects (downtimes, etc.) are missing after restart (broken config package).
- Uses _api package internally
- Active-stage is read from disk every time
  - Race condition: can be empty
  - Incomplete object file path on disk
- Repair broken active stage (timer)
- Logs & troubleshooting docs
- Status: Done (since Friday)

Icinga 2.11

Fixes, crashes, and code quality – all done

Crashes

Bugs

Quality

- Permission filters API crashes #6874 (ref/NC)
- Logrotate timer crash #6737
- Replay log not cleared #6932
- Windows agent 100% cpu/logging #3029
- JSON library: YAJL -> Nlohmann #6684
- UTF8 sanitizing #4703
- Boost Filesystem for I/O #7102
- Boost Asio Thread Pool (checks, etc.) #6988

Icinga 2.11

Missing, what's next?

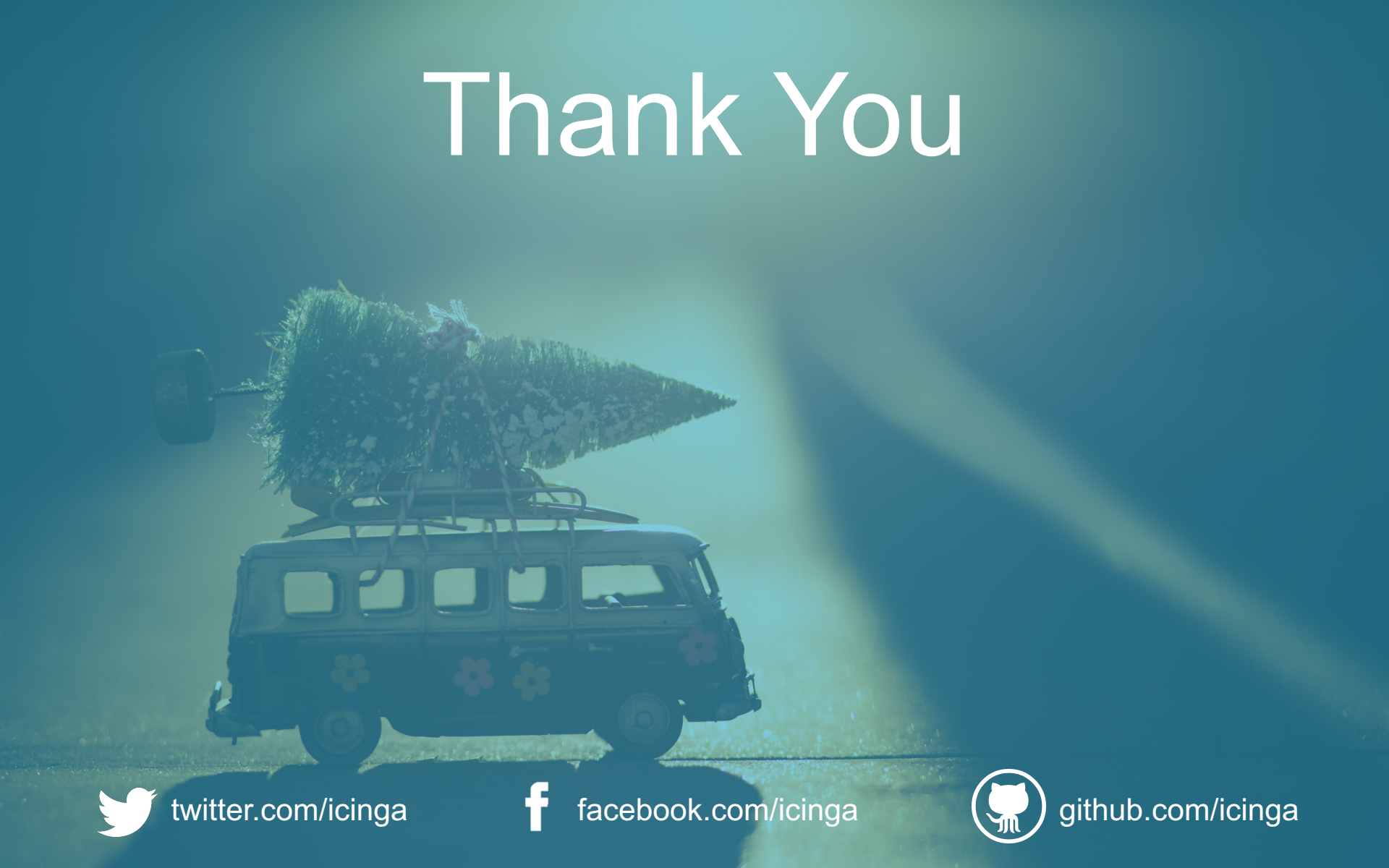
Test

Fix

Profit

- Finish Network Stack
- Review PoC for Cluster Config Sync
- Customer problems, anything else
- Draufhaun - https://github.com/Icinga/icinga2/milestone/72
- Extensive tests from customers, partners, users required
- Overall status: 60%

Thank You



twitter.com/icinga

facebook.com/icinga

github.com/icinga