ONOS Update

Retrospective, Today, Future Direction

Thomas Vachuska - ONF

Overview

- Retrospective of where ONOS has been
- Update on where ONOS is now
- Preview of where ONOS is going

Retrospective

- In the last 12 months, ONOS had the following releases:
  - 1.12 (*Magpie*), 1.13 (*Nightingale*), 1.14 (*Owl*), 1.15 (*Peacock*)
- The primary focus of these releases:
  - support for deployments and operations
    - diagnostics, remote admin, new REST APIs, cluster helm charts, etc.
  - scalability optimizations
    - separate Atomix/ONOS cluster, distributed stores, etc.
  - reducing technical debt
    - bugs, incubator cleanup, build framework upgrade, etc.
- The above is just for the core platform... in the meantime...

Retrospective

- ONOS community continued to add apps, device drivers, etc.
    - 174 apps available
- New SB APIs for NG SDN & Stratum
    - gNMI, gNOI, P4Runtime
- GUI rewrite using Angular 7 and TypeScript
    - amazing job by Sean Condon
- ONOS/Trellis entered production deployments in Comcast
    - currently on 1.12.1, upgrading to 1.12.2
    - evaluating 1.14.0 for larger scale deployments

Current Release (*Peacock*)

- Java 8 LTS is about to reach obsolescence
  - Java 11 is the next LTS
- Apache Karaf 4.2.1 is the latest release of Karaf
  - also the first release that does not require Java 8
- Goal set upgrade Apache Karaf and Java versions
  - needed to ensure that ONOS remains a stable & viable platform
- But… update to Karaf 4.2.1 and associated OSGi framework involved changes that are not backwards compatible… so...

Peacock Release - Two ONOS Versions

- TST deliberately chose to move ONOS major version to 2.0
- Peacock work performed in two parallel streams:
  - code-base retrofit to Karaf 4.2.1 done on separate branch
  - normal development done on master
- The upgrade work took ~8 weeks
- Resulted in two ONOS versions

ONOS 1.15 (*Peacock*)

- ONOS 1.15 is the last release on Apache Karaf 3.0.8 and Java 8
- Captures all work done during the 8 weeks the code base was being upgraded on a separate dev branch
- Carries the original Peacock label
- Released last week
    - November 27th, 2018

ONOS 2.0 (*Quail*?)

- ONOS 2.0 is the first release on Apache Karaf 4.2.1 and Java 11
- Provides the necessary platform upgrade for long-term viability
    - backward incompatibilities are relatively minor
        - entire ONOS repo was retrofitted; other code should be easy to retrofit
    - core incubator eliminated
        - code either promoted to core, moved to apps or removed
    - code-base disaggregation postponed
- Yet-to-be named
    - Quail?, Quetzal?, ... naming poll will be announced soon
- Expected to release mid-December

Where we are now

- ONOS provides a stable platform with nice characteristics:
  - easy app development
    - SDK, distributed stores/primitives, app archetypes, etc.
  - easy deployment as a distributed cluster
    - Docker containers, Kubernetes, etc.
  - automatic service injection
  - super-fast
    - service calls are just method calls
  - lots of existing apps and extensions
    - protocol extensions, device drivers, utilities, etc.
    - support for both legacy protocols and next-gen SDN interfaces

Where we are now

- ONOS architecture also has some caveats and limitations:
  - limited isolation mechanism
    - core & apps share same resources
  - unable to have tenant-specific apps
    - only tenant-aware ones
  - apps limited to Java or JVM-based languages
    - e.g. Scala, Jython, Groovy
  - horizontal app/service scaling is difficult
    - enforced cluster symmetry
  - difficult to migrate components off-platform
    - e.g. control-plane modules embedded on switch

Looking Ahead

- With ONOS 2.0 being a stable platform for some time to come, now is the time to consider next generation architecture
- With UPAN reference design starting to materialize with Stratum being its DP, now is the time to consider its CP
- Goal is to establish the next generation SDN controller architecture
  - completely in the open and with the help of the ONOS community
  - kick of at start of 2019
- Continue to curate ONOS 1.x & 2.x maintenance and releases
  - core team to focus solely on bug fixes, code reviews and release engineering
  - ONOS community to continue new feature development

NG ONOS Architectural Tenets

- Use gRPC-centric interfaces
  - gNMI, gNOI, P4Runtime, OpenConfig, etc.
- Follow micro-services principles
  - horizontal scaling of services, support for tenant apps, etc.
- Rely on existing orchestration platforms
  - e.g. Kubernetes, Helm charts
- Reuse code as appropriate
  - e.g. Atomix, GUI, protocol libraries
- Focus on features required for production deployments
  - live update, diagnostics, monitoring, integrations with orchestrators, etc.
- Allow components written in different languages
  - Java, Go, Python, etc.

Engage!

(Make it so!)