

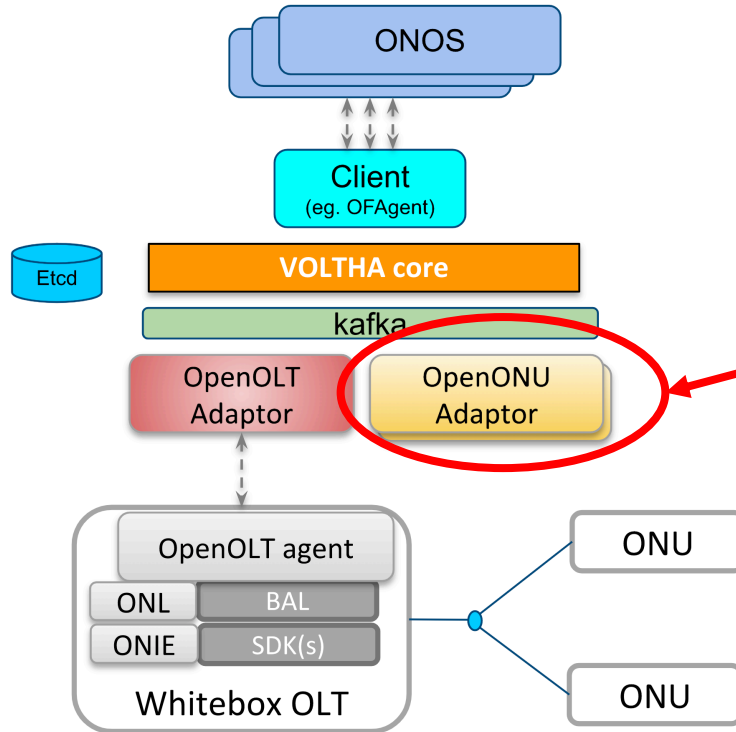


OpenONU-adapter-go

Michael Pagenkopf, Adtran
Holger Hildebrandt , Adtran
Andrea Campanella, ONF

ONF Spotlight 2020

Overview



Focus of this presentation

Motivation

- Main requirement to OpenONU adapter: One instance has to serve at least 1024 ONUs (32 PONs with 32 ONUs each)
- Existing Python OpenONU adapter showed scalability issues with this requirement
- At the time being 8 Python OpenONU adapters are necessary to support up to 1024 ONUs
- Deployment of multiple Python OpenONU adapters leads to additional complexity of the system and increased consumption of processing power

Solution

- Replace existing Python-based single-threaded OMCI library and OpenONU adapter by multi-threaded versions written in Golang
- Goal is to make use of multi-threading features of Golang to parallel communication with particular ONUs
- Existing functionality and capabilities of the Python-version have to be kept
- Meanwhile, porting of OMCI library was done already whereas porting of OpenONU adapter is still in progress

Current State of Implementation 1/2

- Adapter comes up in VOLTHA kubernetes framework
- BBSIM-ONUs reach state 'active' and reason 'initial-mib-downloaded'
- OMCI data are received by MIB-upload-process and stored in internal database
- ONU-devices/ports are visible in ONOS
- Adapter concurrency protection enhancements are deployed to ensure setup of multiple ONUs started in parallel
- One adapter is able to bring-up up to 1024 BBSIM-ONUs up to the state described above

Current State of Implementation 2/2

- OMCI MIB-download to provide ONU basic-config is implemented
- Adapter is able to use existing MIB-templates from etcd to skip MIB-upload-process
- Configuration of device/port admin-state is supported
- Adapter supports processing of ONU oper-status 'unreachable'/'down'
- Special Robot-testsuite exists to check intermediate implementation steps supporting variable numbers of ONUs between 1 and 1024
- All code is merged to Git

Git-location

<https://github.com/opencord/voltha-openonu-adapter-go>

opencord / voltha-openonu-adapter-go

<> Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights

OpenONU Adapter written in GO for VOLTHA

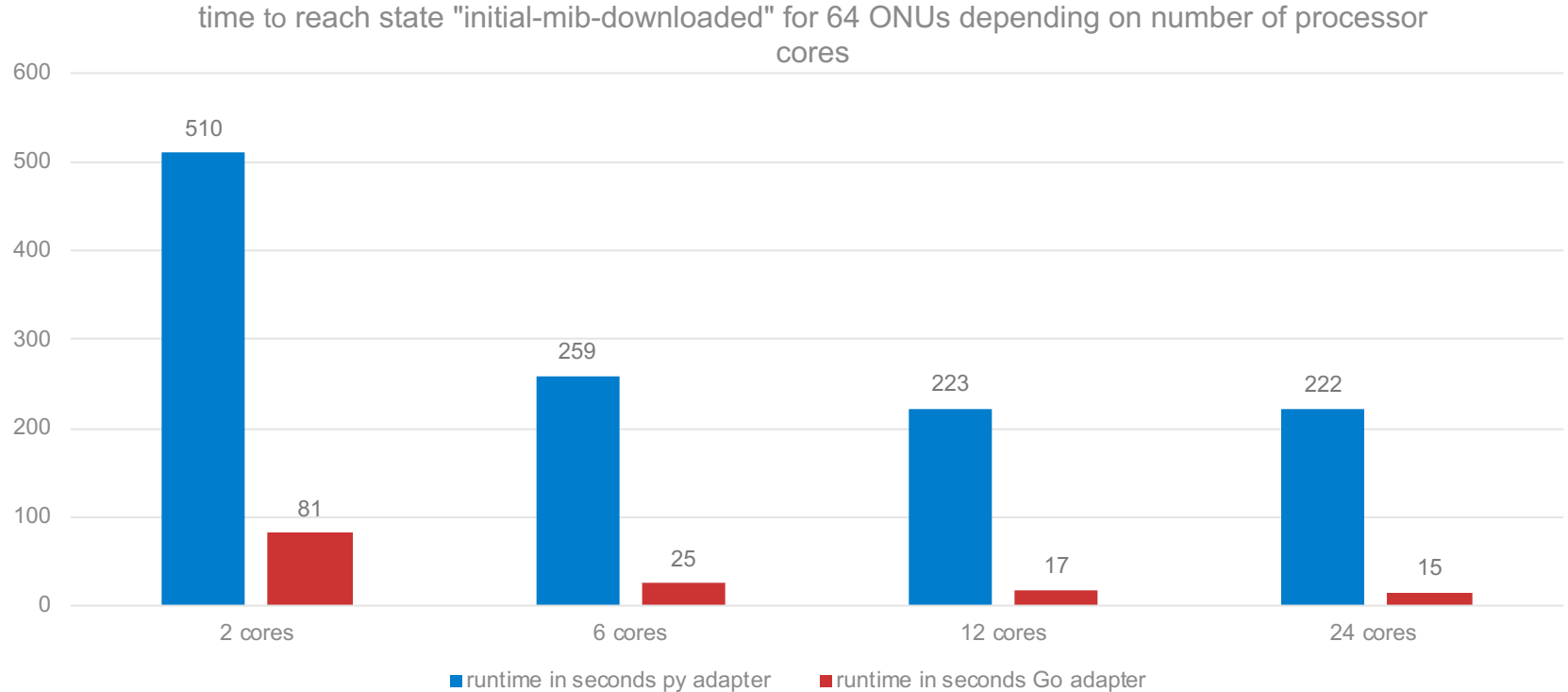
16 commits 1 branch 0 packages 0 releases 4 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

File/Folder	Description	Last Commit
cmd/openonu-adapter	Support per package log level	2 months ago
docker	Using the same dockerfile as openolt-adapter	2 months ago
internal/pkg	[VOL-3036] Read MIB Templates from ETCD	21 hours ago
pkg/mocks	WIP [VOL-2811] - Incorporate preliminary onu-adapter-go code into ope...	2 months ago
vendor	Support per package log level	2 months ago
.gitignore	Docker build back to working state	3 months ago
.gitreview	[VOL-2763] - Create Make- and Dockerfile for Golang OpenONU Adapter	3 months ago
Makefile	Support per package log level	2 months ago
README.md	[VOL-2763] - Create Make- and Dockerfile for Golang OpenONU Adapter	3 months ago
VERSION	[VOL-3036] Read MIB Templates from ETCD	21 hours ago
go.mod	Support per package log level	2 months ago
go.sum	Support per package log level	2 months ago

Runtime comparison - Python vs Golang version

BBSIM-environment, 8 PONs with 8 ONUs each, only one adapter instance and no MIB-template was used



Additional runtime tests

- The following test results are available for the Golang version only since a single instance of Python adapter run into timeouts during the tests
- Time to bring-up ONUs to state „initial-mib-downloaded“ on a server with 24 cores. BBSIM-environment, only one adapter instance and no MIB-template was used:
 - 512 ONUs: 80.29 seconds (32 PONs with 16 ONUs each)
 - 1024 ONUs: 146.74 seconds (64 PONs with 16 ONUs each)*

(*) to be repeated with 32 PONs/32 ONUs

Next Steps

- Configuration of traffic profiles (T-Cont/GemPort/TrafficScheduler etc.)
- Change port status according to physical events
- Full-blown error/timeout treatment (reporting and reaction)
- ONU soft and hard reboot
- OMCI alarm/test/performance supervision and notification
- ONU SW Update
- Support creation of MIB-templates
- Dynamic config of ports/devices based on external config requests or ONU state changes
- Supervision and response to inbuilt liveness/probe checks
- Complete storage of OMCI data within etcd (to enable automatic reflow)



Thank You

<https://github.com/opencord/voltha-openonu-adapter-go>