High Volume VES Collector

General information

- Repository: https://gerrit.onap.org/r/#/admin/projects/dcaegen2/collectors/hv-ves
- Changes: https://gerrit.onap.org/r/#/q/project:dcaegen2/collectors/hv-ves

Background

HV-VES collector has been proposed, based on a need to process high-volumes of data generated frequently by a large number of NFs. The driving use-case is the 5G RAN, where it is expected that up to 10k NF instances report the data, per DCAE platform deployment. The network traffic generated in simulations - based on 4G BTS Real-Time PM data has shown, that GPB serialization is 2-3 times more effective, than JSON serialization utilized in VES collector.

Results have been published within ONAP presentation in Casablanca Release Developer Forum: Google Protocol Buffers versus JSON - 5G RAN use-case - comparison

The goal of the collector is to support high volume data. It uses plain TCP connections tunneled in SSL/TLS. Connections are stream-based (as opposed to request-based) and long running. Payload is binary-encoded (currently we are using Google Protocol Buffers). HV-VES uses direct connection to DMaaP's Kafka. All these decisions were made in order to support high-volume data with minimal latency.

For more details on the rationale, please read a high-level feature description.

Description

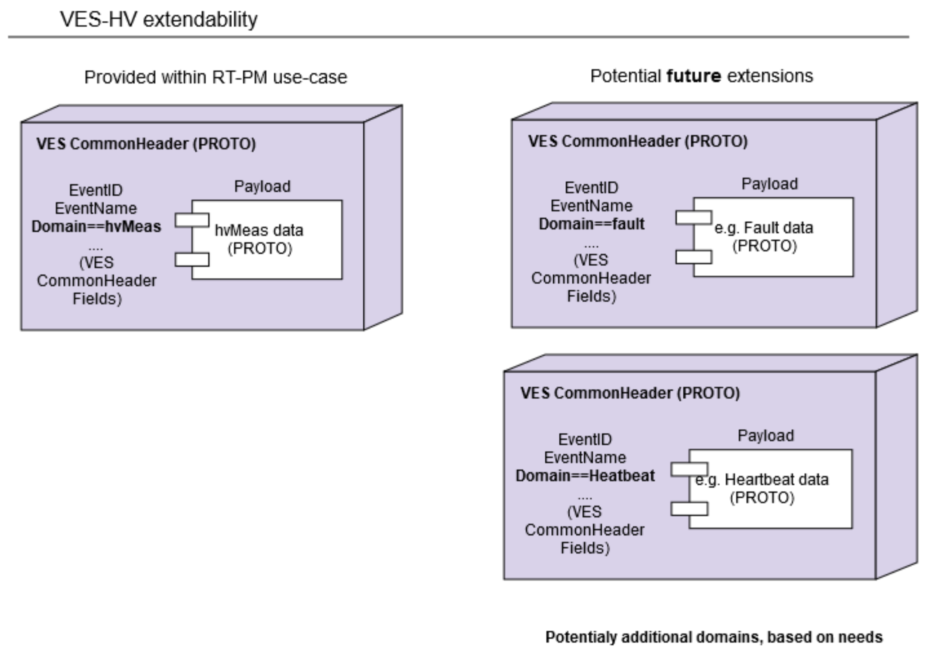
Compatibility aspects (VES-JSON)

- VES-HV has been designed as a high-volume variant of the existing VES(JSON) collector, and not a completely new collector
- VES-HV follows the VES-JSON schema - as much as possible
  - It uses a PROTO representation of the VES Common Header
  - The PROTO files tend to use most encoding effective types defined by GPB to cover Common Header fields.
  - It makes routing decisions based mostly on the content of the "Domain" parameter
  - It allows to embed Payload of different types (by default hvMeas domain is included)
- VES-HV publishes events on DMaaP-Kafka bus, using native Kafka Interfaces
- Analytics applications impacts
  - An analytics application operating on high-volume data needs to be prepared to read directly from Kafka
  - An analytics application need to operate on GPB encoded data in order to benefit from GPB encoding efficiencies
  - It is assumed, that due to the nature of high volume data, there would have to be dedicated applications provided, able to operate on such volumes of data.

Extendability

VES-HV was designed to allow for extendability - by adding new domain-specific PROTO files.

The PROTO file, which contains the VES CommonHeader, comes with a binary-type Payload parameter, where domain-specific data shall be placed.
Domain-specific data are encoded as well with GPB, and they do require a domain-specific PROTO file to decode the data.
This domain-specific PROTO needs to be shared with analytics applications - VES-HV is not analyzing domain-specific data.



in order to support the RT-PM use-case, VES-HV includes a "hvMeas" domain PROTO file, as within this domain, the high volume data is expected to be reported to VES-HV collector.
Still, there are no limitations to define additional domains, based on existing VES domains (like Fault, Heartbeat) or completely new domains. New domains can be added "when needed".

In case of new domains, it is necessary to extend the Common Header PROTO "Domain" enumeration with new values covering this new domain(s).
GPB PROTO files are backwards compatible, and such a new domain could be added without affecting existing systems.

Analytics applications will have to be as well equipped with this new domain-specific PROTO file.
Currently, these additional, domain specific proto files could be simply added to respective repos of VES-HV collector.

Implementation details

Technology stack

- Project Reactor is used as a backbone of the internal architecture.
- Netty is used by means of reactor-netty library.
- We are using Kotlin so we can write very concise code with great interoperability with existing Java libraries.
- Types defined in rrow library are also used when it improves readability or general cleanness of the code.

Rules

- Do not block. Use non-blocking libraries. Do not use block* Reactor calls inside the core of the application.
- Pay attention to memory usage.
- Do not decode the payload - it can be of a considerable size. The goal is to direct the event into a proper Kafka topic. The routing logic should be based only on VES Common Header parameters.
- All application logic should be defined in hv-collector-core module and tested on a component level by tests defined in hv-collector-ct. The core module should have a clean interface (defined in boundary package: api and adapters).
- Use Either functional data type when designing fail-cases inside the main Flux. Using exceptions is a bit like using goto + it adds some performance penalty: collecting stack trace might be costly but we do not usually need it in such cases. RuntimeExceptions should be treated as application bugs and fixed.

Stories

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
DCAEGEN2-712	HV VES Seed code	🟩	Aug 17, 2018	Sep 17, 2018		Unassigned	None	^	CLOSED	Done

DCAEGEN2-689	CI/CD repository setup	🟩	Aug 07, 2018	Dec 30, 2018		Unassigned	None	=	CLOSED	Done
DCAEGEN2-713	HV VES Component Tests	🟩	Aug 17, 2018	Sep 17, 2018		Unassigned	None	=	CLOSED	Done
DCAEGEN2-714	HV VES DCAE Platform Integration	🟩	Aug 17, 2018	May 12, 2020		Unassigned	None	=	CLOSED	Done
DCAEGEN2-828	HV VES Documentation	🟩	Sep 24, 2018	May 15, 2020		Unassigned	None	=	CLOSED	Not Done
DCAEGEN2-984	Update proto file to v14	🟩	Nov 20, 2018	Dec 14, 2018		Unassigned	None	=	CLOSED	Done
DCAEGEN2-1028	HV-VES Performance verification	🟩	Dec 12, 2018	Jul 21, 2020		Unassigned	None	=	CLOSED	Done
DCAEGEN2-1065	HV-VES Manual scale-in	🟩	Jan 08, 2019	Apr 08, 2019		Unassigned	None	=	CLOSED	Done
DCAEGEN2-1069	SDK: HV-VES Producer client	🟩	Jan 09, 2019	Apr 08, 2019		Unassigned	None	=	CLOSED	Done
DCAEGEN2-1132	Assure HV-VES compiles and works on arm64 architecture	🟩	Jan 29, 2019	Dec 09, 2021		Unassigned	None	=	CLOSED	Done
DCAEGEN2-1148	Fix CLM scan vulnerabilities	🟩	Feb 01, 2019	Apr 05, 2019		Unassigned	None	=	CLOSED	Done
DCAEGEN2-688	HV-VES Logging enhancement	🟩	Aug 07, 2018	May 15, 2020		Unassigned	None	⌄	CLOSED	Not Done
DCAEGEN2-710	HV VES Improvements	🟩	Aug 17, 2018	May 15, 2020		Unassigned	None	⌄	CLOSED	Not Done
DCAEGEN2-711	HV VES Analysis	🟩	Aug 17, 2018	Nov 06, 2018		Unassigned	None	⌄	CLOSED	Won't Do

14 issues