

Distributed Analytics as a Service (Dublin Summary) - Edge Automation

Purpose:

Main purpose is to ensure that metric/log collection, correlation & analysis, and closed loop actions are performed closer to the data.

Analytics can be infrastructure analytics, VNF analytics or application analytics. But in Dublin, analytics-as-a-service is proved using infrastructure analytics.

- Big Data analytics to make sure that the analysis is accurate.
- Big data frameworks to allow usage of machine learning and deep learning.
- Avoid sending large amount of data to ONAP-Central for training, by letting training happen near the data source (Cloud-regions).
- ONAP scale-out performance, by distributing some functions out of ONAP-Central such as Analytics
- Letting inferencing happen closer to the edges/cloud-regions for future closed loop operations, thereby reducing the latency for closed loop.
- Opportunity to standardize Infra analytics events/alerts/alarms through output normalization across ONAP-based and 3rd party analytics application.

Owner : [Dileep Ranganathan](#) (Intel), TBD (from VMWare)

Participating Companies: Intel, VMware

Operator Support: China Mobile, Vodafone

Parent page: [Fine Grain Placement Service \(F-GPS\) Edge Automation \(Dublin\)](#)

Link to presentation documents : [Distributed Analytics-as-a-Service presentations](#)

Use Case Name : Distributed Analytics

Why we are terming this as use case instead of functional requirement?

Initially, distributed analytics is projected as 'functional requirement' as it was felt that all existing use-cases can leverage distributed analytics. Due to various reasons such as - not able to leverage DCAE/CLAMP due to resource issues, significant work in basic work to deploy analytics framework and analytics application such as deployment & configuration, it was felt to work on this basic work in R4. As part of basic work, existing use cases will not be integrated. This basic work only consists of generic deployment and configuration, which normally does not require enhancements to existing source code. But, it will require creation of new micro-services that would be deployed in cloud regions.

Showcase	Test Environment	Integration Team Liaison
Deploy Big Data analytics training framework on multiple cloud-regions	Intel/Windriver Lab, VMware Lab (TBD)	Srivahni Chivukula and Rajamohan Mohan Raj
Deploy inferencing framework on multiple cloud regions	Intel/Windriver lab	Srivahni Chivukula and Rajamohan Mohan Raj
Deploy collection services on multiple cloud regions	Intel/Windriver Lab, VMware Lab (TBD)	Srivahni Chivukula and Dileep Ranganathan
Deploy test analytics application as any other workload	Intel/Windriver Lab, VMware Lab (TBD)	Srivahni Chivukula , Dileep Ranganathan and Rajamohan Mohan Raj

Dublin focus

1. Support for following packages (Helm charts bundles)
 - a. Collection Service Package - Consisting of Prometheus, Node-exporter, cAdvisor and CollectD
 - b. Training framework package - Consisting of HDFS, M3DB, Spark with TF, Scikit-learn and other math libraries, HDFS writer, TF Model optimization tools and in future OpenVINO optimization tools
 - c. Messaging package - Kafka broker, Zookeeper
 - d. Model repository package - Consisting of Minio
 - e. Inferencing package - Consisting TF model serving and in future OpenVINO model serving
 - f. Visualization package - Consisting of Grafana to visualize data in M3DB, Hue for HDFS visualization.
 - g. Operators package - Consisting of kubernetes operators.
2. Sample/Example analytics application packages (Helm Chart bundles)
 - a. CPU & Memory threshold crossing application package.
 - b. MNIST application package for training
 - c. MNIST application package for inferencing
3. Helm Guidelines to ensure that helm charts can be deployed using OOM, SDC/SO/K8S and future EOM

4. Cloud infra Event/Alert/Alarm/Fault Normalization & Dispatching microservice deployment on K8S.
5. Spark Application management (to dispatch application image to various cloud regions)
6. Ensure that the Prometheus upon collection write into multiple M3DB instances and write into multiple Kafka brokers/topics
- 7.

Dublin Assumptions:

- Kubernetes support in Cloud regions (support others in future. What to be supported is TBD)
- PNDA as a base (Alignment with DCAE – DCAE already decided to use PNDA framework)
- Spark framework even for both training and inference (future - make the inference as a Micro Service for easier deployment and make the inference as set of executable to be deployed even within application/NF workload or in the compute node)
- Instantiates in new name space (not on existing namespace) in remote cloud regions
- Dynamic configuration updates to analytics applications via operators Other mechanisms for further study.
- Closed loop actions are performed at the ONAP-Central.

DCAE/CLAMP integration

It was felt that DCAE integration can't happen in R4 due to lack of understanding and resources. Hence the intention is to develop common items in R4 and integrate with DCAE/CLAMP in future releases. But, during Dublin time frame, like to do following though.

- Understand on how DCAE/CLAMP can play role in analytics-as-a-service.
- Work done elsewhere that is happening in R4. These are dependencies for DCAE/CLAMP integration with analytics-as-a-service
 - PNDA is integration in DCAE
 - Understand how cloudify plugin works in SO
 - Helm charts based analytics app description support in Cloudify
 - Cloudify HA support
 - Dynamic configuration support
 - Dedicated analytics app for VNFs.
- Identify work items
- Create E2E sequence flows.

Why we have chosen SDC/SO/OOF/MC approach to deploy analytics framework and analytics applications

Analytics applications in cloud-regions are being treated as any other workloads for following reasons

- Bring up analytics applications along with the VNF - Yet times, analytics applications are dedicated to the VNF. Analytics applications are expected to be brought up when VNF is being brought up and terminate when VNF is terminated. Hence, it is felt that analytics application is also described in the same service as VNF.
- Need for bringing up analytics application in the same place as VNF - It can be achieved using VNF and analytics-app affinity rules and hence need to be part of the service.
- Need for bringing up analytics applications in compute nodes having accelerators (e.g ML/DL) - ONAP/OOF can provide this functionality.
- Need for bringing up analytics applications in the right cloud regions based on cost and distance from the Edge locations - ONAP/OOF can provide this functionality.
- Consistent configuration orchestration across components of analytics applications - Leverage MC provided configuration service even for analytics applications
- Configuration of dependent services or VNFs/NFVI/existing-services - Yet times, configuration of dependent services is required as part of bringing up analytics application. Also, when analytics-app is terminated, added configuration needs to be removed. Since, this requirement is same for VNFs, same facilities can be leveraged here too.

JIRA Stories:

Key	Summary	T	Assignee	P	Status	Resolution
ONAPARC-508	Helmize sample promql_query_app		Unassigned		CLOSED	Done
ONAPARC-506	Prometheus missing RBAC		Unassigned		CLOSED	Done
ONAPARC-492	Modify Helm Charts for Minio to include gloo Ingress controller		Unassigned		CLOSED	Done

ONAPARC-491	Create Helm charts for Gloo Ingress		Unassigned	=	CLOSED	Done
ONAPARC-490	Modify cleanup-crds.yaml		Unassigned	=	OPEN	Unresolved
ONAPARC-488	Separate collectd plugin config from configmap		Unassigned	=	CLOSED	Done
ONAPARC-487	Create Model repo package		Unassigned	=	CLOSED	Done
ONAPARC-486	Enable rolling update feature of daemonset		Unassigned	=	CLOSED	Done
ONAPARC-485	Visualization Helm charts package		Unassigned	^	CLOSED	Done
ONAPARC-474	Make collectd and cadvisor values.yaml flat		Unassigned	≡	CLOSED	Done
ONAPARC-469	Understand prometheus-config-reloader		Unassigned	=	CLOSED	Done
ONAPARC-468	Add "common" directory in DAaaS project		Unassigned	≡	CLOSED	Done
ONAPARC-467	Push custom spark-image Dockerhub		Unassigned	∨	CLOSED	Done
ONAPARC-466	Setup Istio Ingress and Metal LB for SLB		Unassigned	^	IN PROGRESS	Unresolved
ONAPARC-465	Create Day-0 config profile for inference package		Unassigned	=	OPEN	Unresolved
ONAPARC-464	Create Day-0 config profile for M3DB		Unassigned	=	OPEN	Unresolved
ONAPARC-463	Create Day-0 config profile for Kafka		Unassigned	=	OPEN	Unresolved
ONAPARC-462	Create Day2 config for M3DB		Unassigned	=	OPEN	Unresolved
ONAPARC-461	Create collectd operator		Unassigned	=	CLOSED	Done
ONAPARC-460	Explore minio api to save model from Analytics application		Unassigned	=	CLOSED	Done
ONAPARC-459	Create Day-2 config for inference package		Unassigned	=	OPEN	Unresolved
ONAPARC-458	Create Day2 config for Kafka		Unassigned	=	OPEN	Unresolved
ONAPARC-457	Create Day2 config for collection service		Unassigned	=	CLOSED	Done
ONAPARC-456	Create Day2 config for Spark and HDFS		Unassigned	=	OPEN	Unresolved
ONAPARC-455	Create Day-0 profiles for Spark, HDFS.		Unassigned	=	OPEN	Unresolved
ONAPARC-454	Helm Charts for Inference Package		Unassigned	^	OPEN	Unresolved
ONAPARC-453	Analyze requirement and scope for HDFS operator		Unassigned	∨	IN PROGRESS	Unresolved
ONAPARC-452	Create a sample Spark application reading from M3DB.		Unassigned	∨	CLOSED	Done
ONAPARC-451	Explore and create Kafka to HDFS writer using Kafka Connect or other mechanisms		Unassigned	^	CLOSED	Done
ONAPARC-450	Create sample Tensorflow MNIST application using horovod		Unassigned	^^	CLOSED	Done

ONAPARC-449	Create a sample Spark application with Minio integration		Unassigned	=	OPEN	Unresolved
ONAPARC-448	Add Anaconda with tensorflow, keras, horovod support to Spark image		Unassigned	^	CLOSED	Done
ONAPARC-447	Create Spark application helm chart as part of application package		Unassigned	^	CLOSED	Done
ONAPARC-446	Create helm charts for HDFS		Unassigned	^	CLOSED	Done
ONAPARC-445	Create helm chart for Spark on K8S operator		Unassigned	^	CLOSED	Done
ONAPARC-444	Move the prometheus operator from collection package to operator package		Unassigned	=	CLOSED	Done
ONAPARC-443	Test Prometheus day-2 incremental config.		Unassigned	^	CLOSED	Done
ONAPARC-442	Remove hardcoded values for release and namespace from prometheus values.yaml		Unassigned	^	CLOSED	Done
ONAPARC-441	Delete deprecated charts		Unassigned	v	CLOSED	Done
ONAPARC-440	Fix node exporter integration with Prometheus.		Unassigned	=	CLOSED	Done
ONAPARC-439	Fix collectd prometheus integration issues		Unassigned	=	CLOSED	Done
ONAPARC-438	Create cAdvisor helm charts		Unassigned	=	CLOSED	Done
ONAPARC-437	Helm Charts for M3DB		Unassigned	^	DELIVERED	Unresolved
ONAPARC-393	Prometheus based Collection		Unassigned	^	DELIVERED	Unresolved
ONAPARC-392	Kafka Broker via Helm Charts and Operators		Unassigned	^	DELIVERED	Unresolved
ONAPARC-391	Helm Charts for HDFS&HBASE		Unassigned	^	OPEN	Unresolved
ONAPARC-366	Docker images & Helm charts required to instantiate big data analytics framework		Unassigned	^^	OPEN	Unresolved

47 issues

Impacted Projects

Project	PTL	JIRA Epic / User Story*	Requirements
Demo repository			1. A repository to keep reference ML/DL Model Management and spark application image management service 2. Reference collection Services to create e2e demo: Prometheus based collection information, Collectd-to-Kafka/Avro, Node-exporter-to-Kafka/Avro and cAdvisor-to-Kafka/Avro 3. Reference ONAP event dispatcher services for e2e demo 4. Demo spark analytics app 5. Reference configuration synchronization container
Demo repository			1. Helm Charts for analytics framework packages 2. Helm Charts for 'infra analytics base' 3. Helm Charts for various analytics applications 4. Helm Chart for Cloud infra Event/Alert/Alarm/Fault Normalization & Dispatching microservice
Multi-VIM /Cloud	Bin Yang		1. May require some changes in K8S plugin

Multi-VIM /Cloud	Bin Yang	Cloud infra Event/Alert/Alarm/Fault Normalization & Dispatching microservice development (see analytics intent example) 1. Integrate DMaAP (Kafka) client for communication to ONAP Central 2. Receive Alert from infra analytics application 3. Normalize from cloud specific Event/Alert/Alarm/Fault format to cloud agnostic (ONAP internal) Event/Alert/Alarm /Fault format 4. Dispatch Alert to ONAP central using DMaAP (Kafka) client
------------------	----------	--

*Each Requirement should be tracked by its own User Story in JIRA

Analytics Intent Example

- "Infrastructure Analytics as service for Alerts at Cluster Level and Host Level for a Cloud Region"

Capabilities (corresponding to Intent) Example:

- Cluster (OpenStack Host Aggregate) Level & Host Level Alerts for compute resources
 - Cluster has unexpected high CPU workload
 - Cluster has memory contention caused by less than half of the virtual machines
 - Cluster has memory contention caused by more than half of the virtual machines
 -
 - Note: Cluster CPU Threshold & Memory Threshold are defined separately
- Ref: <https://docs.vmware.com/en/vRealize-Operations-Manager/6.6/vrealize-operations-manager-66-reference-guide.pdf>

Testing

Current Status

1. Testing Blockers
2. High visibility bugs
3. Other issues for testing that should be seen at a summary level
4. Where possible, always include JIRA links

End to End flow to be Tested

there are many flows that can be tested. We have chosen one which we believe is comprehensive.

Scenario:

1. 3 Edge locations (E1, E2, E3)
2. 2 regional centers (where inferencing is supposed to happen) - R1 and R2
 - a. R1 does inferencing for E1 and E2. R2 does inferencing for E3.
3. 2 central centers (C1, C2)
 - a. C1 does training for App1
 - b. C2 does training for App2
 - c. C1 is also acting as Model repository entity.

ONAP translation:

- 7 Cloud regions. All K8S based.

User actions:

Onboarding

- Create following VSPs and services
 - Collection VSP and Collection Service
 - Training framework VSP and training-framework service
 - Inferencing framework VSP and inferencing-framework service
 - Visualization VSP and Visualization-service
 - Model-repo VSP and Model-repo-service
 - CMTCA VSP and CMTCA-service (CMTCA - CPU & memory threshold crossing application)
 - MNIST-training VSP and MNIST-training-service
 - MNIST-inferencing VSP and MNIST-inferencing-service.
- Create Day0 configuration profiles and upload them using SDC GUI.

Test Cases and Status

#	Test Case	Status
1	There should be a test case for each item in the sequence diagram	NOT YET TESTED
2	create additional requirements as needed for each discreet step	COMPLETE
3	Test cases should cover entire Use Case	PARTIALLY COMPLETE
4	Test Cases should include enough detail for testing team to implement the test	FAILED

- <To be filled>

Bring up & Configure (to be filled)

- Instantiate Collection-service in E1, E2 and E3.
- Instantiate training-framework in R1 and R2.
- Create topic T1 (and other related parameters) in R1 which is needed for HDFS-writer (as Day2 configuration using Kafka operator)
- Create topic T2 (and other related parameters) in R2 which is needed for HDFS-writer(as Day2 configuration using Kafka operator)
- Create DB table (and other related parameters) in R1 M3DB to store data coming from E1 and E2(as Day2 configuration using M3DB operator)
- Create DB table (and other related parameters) in R2 M3Db to store data coming from E3 (as Day2 configuration using M3Db operator)
- Configure Collection instance in E1 with topic T1 as remote-write and M3DB of R1 as remote-write/read