acceldata.

Reference Architecture for Acceldata Deployments.

v0.5

Executive Summary

This document is a high-level design and best practices for Acceldata enterprise deployments for Data Lake Performance Monitoring. It describes the architecture and inner workings of the system, and provides recommendations for Acceldata deployments.

This reference architecture illustrates example AccelData configurations, default alerts and monitors in addition to default operability options. The scope of this document is limited to a single cluster deployment. Features stated in this document should be verified with the release documentation.

Audience and Scope

This guide is for Data Architects, Engineers and Business Process owners who are responsible for Hadoop data operations or those who collaborate with other stakeholders who are experts in the area. This document describes AccelData DLPM and defines deployment recommendations and usage in the following areas:

- Architecture Details
- Hardware Requirements
- Software requirements
- Networking considerations

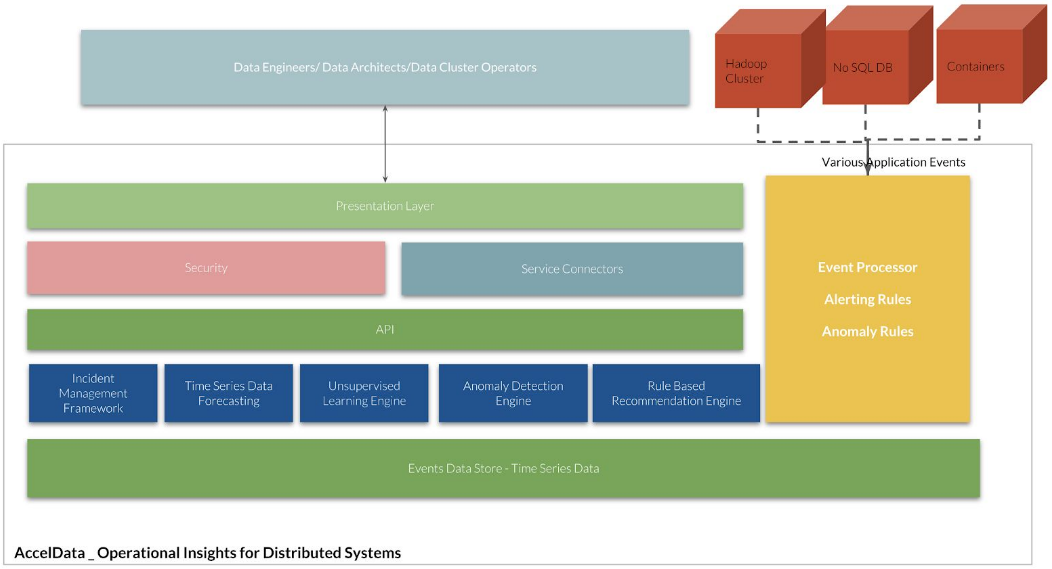
Abstract

Enterprises data is expanding rapidly, with more data getting collected from the Edge Devices, Users and Systems. The number of ETL, BI, analytical, ML & Data Science applications are expanding as well. As such, the performance requirements on data ingest, storage and compute are get stressed. Further, the large number of interacting open-source components combined with the interleaving of large number of various sized jobs make reliable and predictable operations complicated with frequent violations of committed SLAs.

Acceldata DLPM is an native BigData & Hadoop Performance Monitoring Solution. Acceldata provides insights into application performance, detects anomalies and generates alerts which can be used by Data engineers, Architects and Cluster Operators. Further, Acceldata analyzes performance bottlenecks and generates recommendation for remediation. Current offerings include the above abilities on Data Access engines such as - Hive, Tez, Mapreduce and Spark. In addition to this, AccelData monitors YARN capacity to provide usage insights and enables data backed purchase/expansion decisions. Enterprises can onboard new business process with confidence by monitoring performance SLAs, identify rouge users, applications and rectify the issues that cause Business SLA failures.

Architecture Details

The architecture for AccelData is built to collect data from multiple distributed sources and store them reliably on a time-series database (TSDB) . This TSDB is then used for generating future analytics and insights on the Datalake operations and provide actionable prompts to the Hadoop user groups. The insights are of the following nature - aggregated, formulaic and intelligent assessments for root-cause analysis and time series data based forecasting.


Data Engineers/ Data Architects/Data Cluster Operators
Hadoop Cluster
No SQL DB
Containers
Various Application Events
Presentation Layer
Security
Service Connectors
API
Event Processor
Alerting Rules
Anomaly Rules
Incident Management Framework
Time Series Data Forecasting
Unsupervised Learning Engine
Anomaly Detection Engine
Rule Based Recommendation Engine
Events Data Store - Time Series Data
AccelData _ Operational Insights for Distributed Systems

System Components

This section defines the building blocks of the system, the flow of data which includes all areas starting with data capture to insight generation.

Event Processing & Data Collection

Data is generated at different stages of the application execution. Data access jobs are creating data at various mapper and reducer stages, nodes are producing metrics for I/O and networking status, in addition to JMX and other sources of data. Acceldata Kafka based event pipeline has the ability to stream such unrelated data sources. It uses the following core constructs:

Monitors

Monitor is a central concept of AcceData which allows the user to define the data of interest via configurations. A monitor can be attached to a Datalake, a cluster service, process or workflow. Monitors can be composed of other monitors.

Connectors

Connectors are the building blocks of the Monitors which collect data of different kinds from the cluster, services and applications, e.g: Tez, Spark, MR connectors. In addition there are specific connectors for collecting JMX and System Data.

Acceldata Agent

Acceldata Agents are installed on every node of the cluster when there is need for collection of low level JMX, Hardware, and Networking related information. Such data is used for subsequent correlation between the application performance and the underlying hardware infrastructure.

Complex Event Processor

Acceldata uses Siddhi to detect complex conditions described via a Streaming SQL language, and triggers actions of filtering and accepting data. With the dual capability of Stream Processing and Complex Event Processing, administrators can add (or remove) data fields of interest.

Time Series Database

Acceldata uses MongoDB as the time series database to store collected data points and events. MongoDB allows various aggregates stored as a document of their own, which in turn makes querying that data for dashboarding easier.

Incident & Alert Management

Data Lakes have multiple events and incidents of interest to Administrators, Architects and Data Engineers. Acceldata allows such users to instrument on such incidents, which provides opportunity to act ahead of such incident. Acceldata allows users to create alerts on patterns with an ability to correlate multiple, seemingly unrelated set of parameters.

Rules Definition

Rules can be defined on the Acceldata guided interface, using Javascript. These rules are evaluated against the event stream as well as the TSDB. Rules are stored in the configuration repository with a traceback mechanism.

Integrations

These alerts are integrated with SMTP for sending out emails to the interested user group responsible for managing the infrastructure. Webhooks can be supported for limited cases for Jira and Slack.

Presentation Layer

AngularJS is used as the UI framework, along with GraphQL and Rest integrations with the Acceldata middleware services.

Intelligence

Heuristics

Acceldata produces heuristics about query execution repeatedly to produce insights in areas where optimizations can be applied. List of heuristics contain default configurations, which are easily extensible by system administrators.

Root Cause Analysis

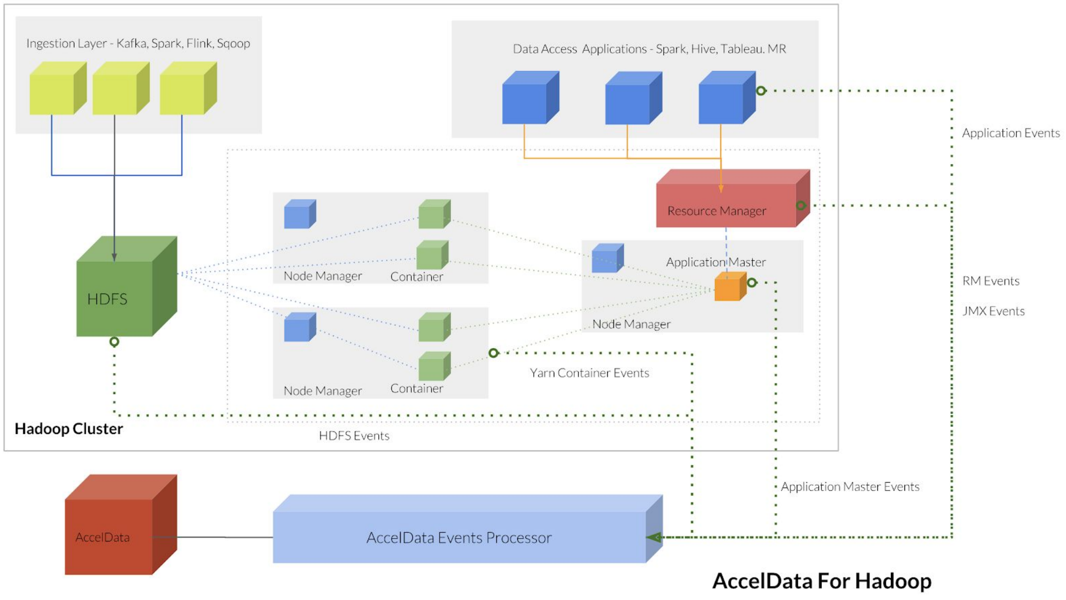
Acceldata collects data from various sources, and then applied unsupervised Machine Learning Algorithm to identify the root cause of the application. This processing is performed on the Application Node, to reduce the load on the TSDB instance

Anomaly Detection & Time Series Forecasting

Rules can be created to identify various kinds of anomalies in terms of capacity, memory, concurrency and throughput values. List of such anomalies contain default configurations, which are extensible by system administrators.

Deployment Architecture

Acceldata is recommended to be deployed in the edge nodes which have access to the cluster nodes and services. Acceldata streaming node requires bidirectional access to the cluster services and nodes for data collection. Limited data, dependent upon the streaming configurations is collected in the TSDB over Kafka stream and over HTTP.



Hardware Requirements

Instance Type	Memory/Cores (Min)	Memory/Cores (Recommended)	Storage	Operating System
Time Series Database	16GB/8Cores	32GB/8Cores	Data Dependent	RHEL (rpm), Ubuntu (.deb)
Application	16GB/8Cores	32GB/8Cores	Data Dependent	RHEL (rpm), Ubuntu (.deb)

Software Requirements

Software Component	Version	License
Java	8	Sun Java
MongoDB	4.0.0	AGPL, Apache
Messaging Kafka	2.0.0	Apache

Configurations

Service Discovery & Network access

Acceldata needs access to the following services, to enable data collection and monitoring. This process is automated, by connecting through Ambari API's or Cloudera Data Manager API's. Acceldata needs access to the following services, as listed in the table below. Deployment time decisions may change the default ports, which should be accessible through the edge nodes, where Acceldata is deployed. In addition to this, the configurations can be added manually as well:

Areas	Component	Endpoint	Protocol
		URL	
Scheduler	Yarn	Resource Manager URL	HTTP

Access	Hive Tez	Application Timeline Server	HTTP
	Hive MR	Map Reduce Job History server	HTTP
	Hive LLAP	Hive Server 2 URL	JDBC
	Spark	HDFS Logs	WebHDFS
	Hive Server 2	Hive Server 2 URL	JDBC
	Spark Thrift Server	HDFS Logs	JDBC
Ingestion	Kafka	Kafka Logs	

Security

Acceldata connectors need a kerberos account and a corresponding keytab to run in a kerberized cluster. The keytab contains the mandatory authentication information which is extracted from the kerberos database and stored locally with the service principal. Acceldata can access kerberized cluster services following the standard steps of creating the service principals and adding them to the respective component services in the cluster.

On-Prem Deployment

All nodes of the cluster need the acceldata agent, which is deployed using Ansible scripts. For this step, the installation process will need the ssh keys, shared with other hadoop management software such as Ambari or Cloudera Manager.

Cloud Deployment

All nodes of the cluster need the acceldata agent, which is deployed using Ansible scripts. For this step, the installation process will need the ssh keys, shared with other hadoop management software such as Ambari or Cloudera Manager. Acceldata can be deployed on AWS & Azure.

Supported Components

Following are the supported components in the supported Hadoop Distributions:

HDP Support

Areas	**Component**	**HDP**		
		On Prem	**Azure**	**AWS**
Scheduler	Yarn	Y	Y	Y
Access	Hive Tez	Y	Y	Y

	Hive MR	Y	Y	Y
	Hive LLAP	Y	Y	Y
	Spark	Y	Y	Y
	Hive Server 2	Y	Y	Y
Ingestion	Kafka	Y	Y	Y

CDH Support

Areas	**Component**	**CDH**		
		On Prem	**Azure**	**AWS**
Scheduler	Yarn	Y	Y	Y
Access	Hive Tez	Y	Y	Y
	Hive MR	Y	Y	Y
	Hive LLAP	Y	Y	Y
	Impala	Y	Y	Y
	Spark	Y	Y	Y
	Hive Server 2	Y	Y	Y
	Spark Thrift Server	Y	Y	Y
Ingestion	Kafka	Y	Y	Y

References

Acceldata