Hortonworks RFI Response to US Cellular

Prepared by Mark Bolsoni, Adis Cesir
mbolsoni@hortonworks.com
630-841-1006

Project Background

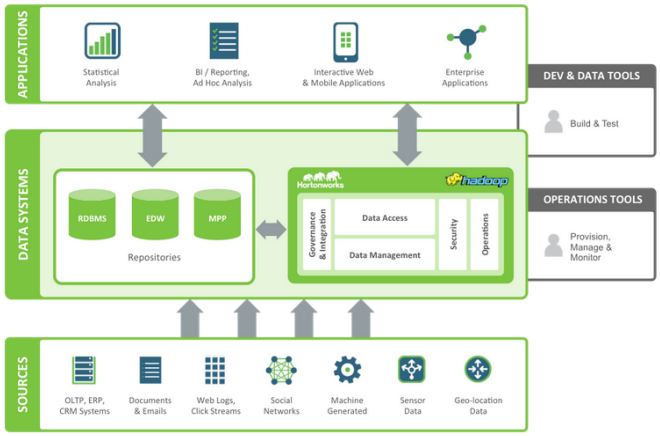
Hortonworks will provide responses to the RFI in the following word document, Supplier Workbook, and attachments. The document will follow the flow and format of the RFI.

3.2 Project Scope

U.S. Cellular® is specifically interested in the following use cases. For the following use cases please include diagrams to help explain the answers.

1. Data Management Approach
    a. Please describe the overall management approach for your solution. There are talks of Data Lake, Data Hub and other architectures. Please describe the most common implementations of your solution.
    b. Please describe how your solution can be used to managing this data Please be sure to include components for:
        i. Data Modeling
        ii. Validating Data Quality
        iii. Publishing Metadata to the end user
        iv. Handling of data lineage from the source system to the end user access
        v. Providing individual data access
        vi. Management of sensitive data in the system

Answer: **Data Lake with Hortonworks Data Platform**



The shortest definition of a data lake is commodity scaled-out storage with collocated compute. However, a data lake is far more than this. Data lakes are a living, breathing record of the business. They retain all data from all source systems. They are a point of consolidation and collaboration. All teams and all silos store data in the lake together. They are a consistent place to secure data. Data can be encrypted at rest and in motion, in a unique way per dataset and then data can be shared in both encrypted and decrypted form based on user access levels. And data lakes are a place data is made high quality and reliable for all to consume in an automated fashion. Data lakes should come with all the tools to label data as to schema and to send data through processing pipelines that log relationships amongst datasets and health of execution of those pipelines through all stages of transformation and derivation. Data lakes are a gather place where the enterprise comes to meet and work together on data.

Foremost, however, data lakes are not just secured multi-tenant storage but are, in fact, a compute cloud or farm where arbitrary applications can run on data without moving that data out of the lake. It is also important to understand that without multi-tenancy support from the very foundations of the lake, the data lake will fail. Multi-tenancy is defined in two ways:

- Security
- Resource fencing

A secure data lake is one that allows all its tenants to bring data to the infrastructure without risk of data leakage or loss. Data leakage occurs when members of one team or users with inappropriate credentials end up reading data belonging to mission critical or even regulated use cases and users. The aspirational architecture must be secure.

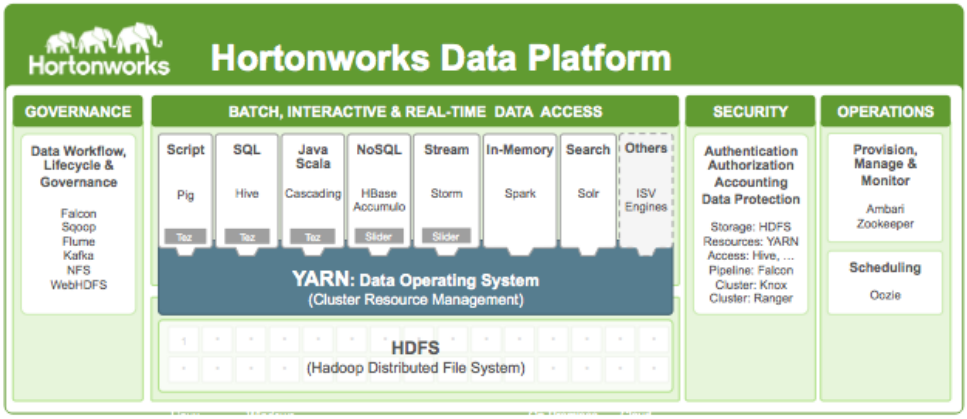
Resource fencing ensures that noisy users on the system do not cause disturbances to other users of the same-shared system. Teams and use cases that are used to a silo-based experience can still feel that level of isolation and throughput guarantee.

Our data lake must be a secure, isolatable environment so that we can ask use cases to pile on without fear of having a degraded user experience relative to their current solution. Only when users and data come to the lake can we begin to both tear down the walls of isolation where appropriate opportunities arise as well as begin to analyze super or über use cases where cross-dataset analysis may reap new and tremendous predictive value to the firm.

- Collect everything. A Data Lake contains all data, both raw sources over extended periods of time as well as any processed data.
- Dive in anywhere. A Data Lake enables users across multiple business units to refine, explore and enrich data on their terms.
- Flexible access. A Data Lake enables multiple data access patterns across a shared infrastructure: batch, interactive, online, search, in-memory and other processing engines.

As data continues to grow exponentially, Hortonworks Data Platform investment can provide a strategy for both efficiency in a modern data architecture, and opportunity in an enterprise Data Lake.

Hortonworks Data Platform 2.2



Governance

- **Apache Falcon:** a framework for simplifying and orchestrating data management and pipeline processing in Apache Hadoop. It enables automation of data movement and processing for ingest, pipelines, replication and compliance use cases. Falcon also leverages its integration with YARN—the architectural center of Hadoop—to centrally manage the cluster's data governance, maximize data pipeline reuse and enforce consistent data lifecycles.

- **Apache Flume**: is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of streaming data into the Hadoop Distributed File System (HDFS). It has a simple and flexible architecture based on streaming data flows; and is robust and fault tolerant with tunable reliability mechanisms for failover and recovery. Flume lets Hadoop users make the most of valuable log data. Specifically, Flume allows users to:

    - Stream data from multiple sources into Hadoop for analysis
    - Collect high-volume Web logs in real time
    - Insulate themselves from transient spikes when the rate of incoming data exceeds the rate at which data can be written to the destination
    - Guarantee data delivery
    - Scale horizontally to handle additional data volume

- **Apache Sqoop**: is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases. Sqoop imports data from external structured datastores into HDFS or related systems like Hive and HBase. Sqoop can also be used to extract data from Hadoop and export it to external structured datastores such as relational databases and enterprise data warehouses. Sqoop works with relational databases such as: Teradata, Netezza, Oracle, MySQL, Postgres, and SQL Server.
- **Apache Kafka**: is a fast, scalable, durable, and fault-tolerant publish-subscribe messaging system. Kafka is often used in place of traditional message brokers like JMS and AMQP because of its higher throughput, reliability and replication. Kafka offers functionality for:

- Stream Processing
- Website Activity Tracking
- Metrics Collection and Monitoring
- Log Aggregation

Data Access: Batch, Interactive, Real-Time, In-Memory and Search

- **Apache Pig:** allows you to write complex MapReduce transformations using a simple scripting language. Pig Latin (the language) defines a set of transformations on a data set such as aggregate, join and sort. Pig translates the Pig Latin script into MapReduce so that it can be executed within Hadoop®. Pig Latin is sometimes extended using UDFs (User Defined Functions), which the user can write in Java or a scripting language and then call directly from the Pig Latin.Pig was designed for performing a long series of data operations, making it ideal for:

    - Extract-transform-load (ETL) data pipelines,
    - Research on raw data, and
    - Iterative data processing.

- **Apache Hive and HCatalog**: Apache Hive is considered the defacto standard for interactive SQL queries over petabytes of data in Hadoop. And with the completion of the Stinger Initiative, and the first phase of Stinger.next, the Apache community has greatly improved Hive's speed, scale and SQL semantics. Throughout all the innovation, Hive easily integrates with other critical data center technologies using a familiar JDBC interface.

Apache HCatalog is a table and storage management layer for Hadoop that enables users with different data processing tools – Apache Pig, Apache MapReduce, and Apache Hive – to more easily read and write data on the grid. HCatalog's table abstraction presents users with a relational view of data in the Hadoop Distributed File System (HDFS) and ensures that users need not worry about where or in what format their data is stored. HCatalog displays data from RCFile format, text files, or sequence files in a tabular view. It also provides REST APIs so that external systems can access these tables' metadata.

    - Frees the user from having to know where the data is stored, with the table abstraction
    - Enables notifications of data availability
    - Provides visibility for data cleaning and archiving tools

- **Cascading**: is a application development framework for building data applications. Acting as an abstraction layer, Cascading does the heavy lifting and converts your applications built on Cascading into MapReduce jobs that run effectively on top of Hadoop. The Cascading SDK provides a collection of tools, documentation, libraries, tutorials and example projects from the greater Cascading community and enables the rapid development of batch and interactive data-driven applications.

    - Lingual. Simplifies systems integration through ANSI SQL compatibility and a JDBC driver
    - Pattern. Enables various machine learning scoring algorithms through PMML compatibility
    - Scalding. Enables development with Scala, a powerful language for solving functional problems
    - Cascading. Enables development with Clojure, a Lisp dialect

- **Apache HBase and Phoenix**: Apache HBase is a non-relational (NoSQL) database that runs on top of the Hadoop® Distributed File System (HDFS). It is columnar and provides fault-tolerant storage and quick access to large quantities of sparse data. It also adds transactional capabilities to Hadoop, allowing users to conduct updates, inserts and deletes.

Apache Phoenix is a relational database layer over HBase delivered as a client-embedded JDBC driver targeting low latency queries over HBase data. Apache Phoenix takes your SQL query, compiles it into a series of HBase scans, and orchestrates the running of those scans to produce regular JDBC result sets. The table metadata is stored in an HBase table and versioned, such that snapshot queries over prior versions will automatically use the correct schema. Direct use of the HBase API, along with coprocessors and custom filters, results in performance on the order of milliseconds for small queries, or seconds for tens of millions of rows.

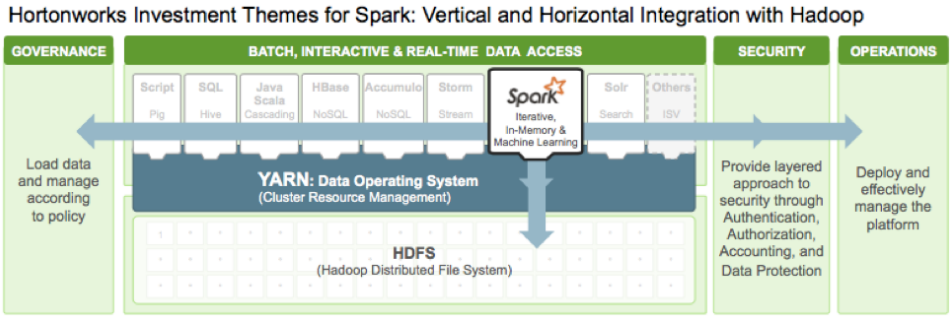
- **Apache Storm**: Storm is a distributed real-time computation system for processing large volumes of high-velocity data. Storm on YARN is powerful for scenarios requiring real-time analytics, machine learning and continuous monitoring of operations. Some of specific new business opportunities include: real-time customer service management, data monetization, operational dashboards, or cyber security analytics and threat detection. Because Storm integrates with YARN via Apache Slider, YARN manages Storm while also considering cluster resources for data governance, security and operations components of a modern data architecture.

Storm is extremely fast, with the ability to process over a million records per second per node on a cluster of modest size. Enterprises harness this speed and combine it with other data access applications in Hadoop to prevent undesirable events or to optimize positive outcomes.

- **Apache Solr**: is the open source platform for searches of data stored in HDFS in Hadoop. Solr powers the search and navigation features of many of the world's largest Internet sites, enabling powerful full-text search and near real-time indexing. Whether users search for tabular, text, geo-location or sensor data in Hadoop, they find it quickly with Apache Solr. Hadoop operators put documents in Apache Solr by "indexing" via XML, JSON, CSV or binary over HTTP. Then users can query those petabytes of data via HTTP GET. They can receive XML, JSON, CSV or binary results. Apache Solr is optimized for high volume web traffic. Features include:

    - Advanced full-text search
    - Near real-time indexing
    - Standards-based open interfaces like XML, JSON and HTTP
    - Comprehensive HTML administration interfaces
    - Server statistics exposed over JMX for monitoring
    - Linearly scalable, auto index replication, auto failover and recovery
    - Flexible and adaptable, with XML configuration

- **Apache Spark**: provides an elegant, attractive development API and allows data workers to rapidly iterate over data via machine learning and other data science techniques that require fast, in-memory data processing. And with Spark on YARN, they can simultaneously use Spark for data science workloads alongside other data access engines–all accessing the same shared dataset.



Spark powers a stack of high-level tools including Spark SQL, MLlib for machine learning, GraphX, and Spark Streaming. You can combine these libraries seamlessly in the same application.

- **Apache Tez:** is an extensible framework for building high performance batch and interactive data processing applications, coordinated by YARN in Apache Hadoop. Tez improves the MapReduce paradigm by dramatically improving its speed, while maintaining MapReduce's ability to scale to petabytes of data. Important Hadoop ecosystem projects like Apache Hive and Apache Pig use Apache Tez, as do a growing number of third party data access applications developed for the broader Hadoop ecosystem.

Apache Tez provides a developer API and framework to write native YARN applications that bridge the spectrum of interactive and batch workloads. It allows those data access applications to work with petabytes of data over thousands nodes. The Apache Tez component library allows developers to create Hadoop applications that integrate natively with Apache Hadoop YARN and perform well within mixed workload clusters.

Since Tez is extensible and embeddable, it provides the fit-to-purpose freedom to express highly optimized data processing applications, giving them an advantage over end-user-facing engines such as MapReduce and Apache Spark. Tez also offers a customizable execution architecture that allows users to express complex computations as dataflow graphs, permitting dynamic performance optimizations based on real information about the data and the resources required to process it.

- **Apache YARN**: Part of the core Hadoop project, YARN is the architectural center of Hadoop that allows multiple data processing engines such as interactive SQL, real-time streaming, data science and batch processing to handle data stored in a single platform, unlocking an entirely new approach to analytics. It is the foundation of the new generation of Hadoop and is enabling organizations everywhere to realize a modern data architecture.YARN is the prerequisite for Enterprise Hadoop, providing resource management and a central platform to deliver consistent operations, security, and data governance tools across Hadoop clusters.

YARN also extends the power of Hadoop to incumbent and new technologies found within the data center so that they to can take advantage of cost effective, linear-scale storage and processing. It provides ISVs and developers a consistent framework for writing data access applications that run IN Hadoop.

YARN is the central point of investment for Hortonworks within the Apache community. In fact, YARN was originally proposed (MR-279) and architected by one of our founders, Arun Murthy. Our engineers have been working within the Hadoop community to deliver and improve YARN for years. It has matured to become the solid, reliable architectural center of Hadoop and is a foundational component.

While relied upon by thousands, YARN can always be improved, especially with new engines emerging to interact with Hadoop data. To this end, Hortonworks has laid out the following investment themes for this foundational technology.

- Reliable Operations
    - Support for rolling upgrades: upgrade a YARN cluster without down time
    - Work-preserving restarts
    - Support NodeGroup layer topology
- Scheduling and Isolation
    - Admin node labels: allow applications to request specific nodes for scheduling tasks
    - YARN admission planner: allow users to reserve future capacity for applications
- Applications on YARN
    - Support Docker for packaging applications to run on YARN
    - Store application timeline data in secure clusters
- Multi-tenancy
    - YARN allows multiple access engines (either open-source or proprietary) to use Hadoop as the common standard for batch, interactive and real-time engines that can simultaneously access the same data set.
    - Multi-tenant data processing improves an enterprise's return on its Hadoop investments.
- Cluster utilization
    - YARN's dynamic allocation of cluster resources improves utilization over more static rules used in early versions of Hadoop
- Scalability
    - Data center processing power continues to rapidly expand. YARN's ResourceManager focuses exclusively on scheduling and keeps pace as clusters expand to thousands of nodes managing petabytes of data.
- Compatibility
    - Existing MapReduce applications developed for Hadoop 1 can run YARN without any disruption to existing processes that already work

- **Apache Slider:** Apache Slider allows users to create and run different versions of heterogeneous long-running applications in Hadoop with YARN. Each application instance can be configured differently, with its operational life cycle managed individually. On an on-demand basis, Slider can expand or shrink application instances while they are running. In the case of container failure, Slider transparently leverages YARN facilities to manage application recovery. All of this is available on Linux or Windows platforms.

    - Topologies: Support for complex application topology
    - Dynamic Scaling: Dynamic scaling of application or component instances
    - Application packaging tools: Support for Docker as a packaging mechanism
    - Application lifecycle management: Support for application upgrades, backup-recovery, relocation

Security

- **Apache Ranger:** Apache Ranger offers a centralized security framework to manage fine-grained access control over Hadoop data access components like Apache Hive and Apache HBase. Using the Apache Ranger console, security administrators can easily manage policies for access to files, folders, databases, tables, or column. These policies can be set for individual users or groups and then enforced within Hadoop.

Security administrators can also use Apache Ranger to manage audit tracking and policy analytics for deeper control of the environment. The solution also provides an option to delegate administration of certain data to other group owners, with the aim of securely decentralizing data ownership.

Apache Ranger currently supports authorization, auditing and security administration of following HDP components:

- Apache Hadoop HDFS
- Apache Hive
- Apache HBase
- Apache Storm
- Apache Knox

Apache Ranger has a decentralized architecture with the following internal components:

- Ranger portal: The portal is the central interface for security administration. Users can create and update policies, which are then stored in a policy database. Plugins within each component poll these policies at regular intervals. The portal also consists of an audit server that sends audit data collected from the plugins for storage in HDFS or in a relational database.

- Ranger plugins: Plugins are lightweight Java programs which embed within processes of each cluster component. For example, the Apache Ranger plugin for Apache Hive is embedded within Hiveserver2. These plugins pull in policies from a central server and store them locally in a file. When a user request comes through the component, these plugins intercept the request and evaluate it against the security policy. Plugins also collect data from the user request and follow a separate thread to send this data back to the audit server.

- User group sync: Apache Ranger provides a user synchronization utility to pull users and groups from Unix or from LDAP or Active Directory. The user or group information is stored within Ranger portal and used for policy definition.

- **Apache Knox:** is a Gateway which provides perimeter security so that the enterprise can confidently extend Hadoop access to more of those new users while also maintaining compliance with enterprise security policies. Knox also simplifies Hadoop security for users who access the cluster data and execute jobs. It integrates with prevalent identity management and SSO systems and allows identities from those enterprise systems to be used for seamless, secure access to Hadoop clusters.

A fully secure Hadoop cluster needs Kerberos. Kerberos requires a client side library and complex client side configuration. By encapsulating Kerberos, Knox eliminates the need for client software or client configuration and thus simplifies the access model. In this way, Knox aggregates REST/HTTP calls to various components within the Hadoop ecosystem.

Knox is a stateless reverse proxy framework and can be deployed as a cluster of Knox instances that route requests to Hadoop's REST APIs. Because Knox is stateless, it scales linearly by adding more Knox nodes as the load increases. A load balancer can route requests to multiple Knox instances. Knox also intercepts REST/HTTP calls and provides authentication, authorization, audit, URL rewriting, web vulnerability removal and other security services through a series of extensible interceptor pipelines.

Operations

- **Apache Ambari:** Apache Ambari is a completely open operational framework for provisioning, managing and monitoring Apache Hadoop clusters. Ambari includes an intuitive collection of operator tools and a set of APIs that mask the complexity of Hadoop, simplifying the operation of clusters. With hundreds of years of combined experience, Hortonworks, along with members of the Hadoop community have answered the call to deliver the key services required for enterprise Hadoop.

Ambari enables system administrators to provision, manage and monitor a Hadoop cluster, and also to integrate Hadoop with the existing enterprise infrastructure.

Provision a Hadoop Cluster

- No matter the size of your Hadoop cluster, the deployment and maintenance of hosts is simplified using Ambari. Ambari includes an intuitive Web interface that allows you to easily provision, configure and test all the Hadoop services and core components. Ambari also provides the powerful Ambari Blueprints API for automating cluster installations without user intervention.

Manage a Hadoop cluster

- Ambari provides tools to simplify cluster management. The Web interface allows you to control the lifecycle of Hadoop services and components, modify configurations and manage the ongoing growth of your cluster.

Monitor a Hadoop cluster

- Gain instant insight into the health of your cluster. Ambari pre-configures alerts for watching Hadoop services and visualizes cluster operational data in a simple Web interface.

Integrate Hadoop with the Enterprise

- Ambari provides a RESTful API that enables integration with existing tools, such as Microsoft System Center and Teradata Viewpoint, to merge Hadoop with your established operational processes.

- Hadoop cluster provisioning and ongoing management can be a complicated task, especially when there are hundreds or thousands of hosts involved. Ambari provides a single control point for viewing, updating and managing Hadoop service life cycles, with these important features:

Other Ambari Features

- Wizard-driven interface: Facilitates installation of Hadoop across any number of hosts
- API-driven installations: Ambari Blueprints for automated provisioning
- Granular control: Precise management of Hadoop services and component lifecycles
- Configuration histories: Ongoing management of Hadoop service configurations
- Extensible framework: Brings custom services under management via Ambari Stacks
- Usability improvements: Innovative user experiences via Ambari Views
- RESTful APIs: Enables integration with enterprise systems

- **Apache Oozie:** Oozie is a Java Web application used to schedule Apache Hadoop jobs. Oozie combines multiple jobs sequentially into one logical unit of work. It is integrated with the Hadoop stack and supports Hadoop jobs for Apache MapReduce, Apache Pig, Apache Hive,

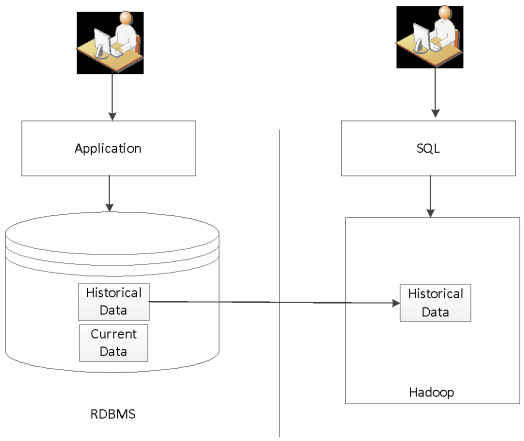
and Apache Sqoop. It can also be used to schedule jobs specific to a system, like Java programs or shell scripts.

There are two basic types of Oozie jobs:

- Oozie Workflow jobs are Directed Acyclical Graphs (DAGs), specifying a sequence of actions to execute. The Workflow job has to wait
- Oozie Coordinator jobs are recurrent Oozie Workflow jobs that are triggered by time and data availability.
- Oozie Bundle provides a way to package multiple coordinator and workflow jobs and to manage the lifecycle of those jobs

Requirement	**HDP-provided tools**	**Optional 3rd party tools**
• Data Modeling • Validating Data Quality	Hive - - Pig - - - - Perl (Streaming) - Cascading ***Declarative Procedural*** Orchestration: Falcon (Oozie)	• Informatica BDE "Governance" Edition • Talend • IBM InfoSphere etc.
Publishing Metadata to the end user	• Hue provides HCatalog metadata • Solr custom interface for searching metadata	Informatica BDE "Governance" Edition: *Business Glossary*
Handling of data lineage from the source system to the end user access	Falcon	Informatica BDE "Governance" Edition: *End-to-End Data Lineage*
Providing individual data access	Hive Clients, Hue (Hive/Pig), Edge Nodes, R, SAS	n/a
Management of sensitive data in the system	Ranger (cluster-wide user Authorization) Knox (cluster-wide user Authentication) Falcon (automated data tagging)	Active Directory (provides Groups/Members only – policies will still be applied by HDP tools)

2. Data Archiving
    a. USCC has a need to archive data out of databases to reduce the size of production databases. In this case, the data is no longer needed by the application, but is required to be kept for auditing purposes. To minimize the impact to the end users, a sql interface to the archived data is required. A sample high-level diagram of the desired Data Archiving solution is below.



b. Please describe how your solution can be used to meet this need including the technology components being used. For this use case, please use the following assumptions:
   i. Oracle 11gR2 or SQLServer 2012
   ii. Only subset of data to be archived
c. Please sure to include components for:
   i. Data extraction
   ii. Data loading to Hadoop
   iii. Publishing metadata to the end user
   iv. Providing access to the end user

Answer: The Hortonworks Data Platform provides a number of tools to facilitate the use of it in cost optimization patterns such as archival of RDBMS data.

Data extraction / Data loading to Hadoop
HDP is capable of various methods of data extraction and loading, depending upon the required latency of replication into Hadoop.

	Data extraction	**Data loading to Hadoop**
Sqoop	Sqoop has the ability to target any RDBMS system for which you have a JDBC driver for. This means Sqoop can easily connect to any of the common Databases of today such as Oracle, or SqlServer of any version. Additionally MainFrame systems like DB2 and IMS that have corresponding JDBC drivers also support MainFrame offload usecases. Sqoop maintains state between delta loads using its own 'Metastore'.	Sqoop then lands this data into HDFS (as flat files) or preferably directly into Hive using the optimized ORC File Format, preserving the source metadata where it can be made available for any platform tool.
3rd Party Replication Tools	Replication technologies such as Oracle GoldenGate offer low-	The data would be then ingested from Apache Kafka into Apache Storm for filtering

e.g. GoldenGate	impact, real-time change data capture, distribution, and delivery for transactional data across heterogeneous systems including Hadoop. Data would be propagated into Apache Kafka via JMS.	and/or processing. Apache Storm is capable of persisting data into HBase or Hive.
Triggers/History Tables	DB Triggers would track changes to the operational table and keep updates in a separate "history" table (some apps do this for audits). For apps that do not, if this information is relevant for Hadoop, create DB triggers (CAUTION: Triggers add extra step to DML transactions). Ideal where a complete history of updates is beneficial to audit or used for calculation in Hadoop (i.e. want every change to a balance, not just balance itself)	Sqoop would then consume the "history" table in the source RDBMS, ingest it into Hive and then purge the history upon successful completion. Orchestration could be performed by Falcon or 3rd party ETL tools.
3rd Party ETL Tools e.g. Informatica	ETL Tools would identify the deltas, extracting them as raw text for landing into Hadoop The Data Lineage produced by these tools can be used to perform impact analysis across the whole ecosystem, not just Hadoop.	Raw files would be ingested using Hive using Sqoop. Orchestration could be performed by Falcon or 3rd party ETL tools.
Scripts e.g. Perl	Perl scripts using dbi and dbd modules can be scheduled using Apache Falcon or 3rd party schedulers or even 3rd party ETL Tools. They would need to be coded to identify the deltas, extracting them as raw text for landing into Hadoop The data lineage would be recorded by the appropriate scheduling tool.	Raw files would be ingested using Hive using Sqoop. Orchestration could be performed by Falcon or 3rd party ETL tools.

Raw text files can be loaded into HDFS via WebHDFS REST API calls, HDFS CLI Clients or simply copied into the HDFS NFS Gateway (appears to external tools as an NFS mount)

Publishing metadata to the end user

Apache HCatalog

Apache HCatalog provides flexible metadata services across tools and external access. Metadata created within it can be shared across any tool within the platform. File locations become abstract 'tables' (not hard-coded file paths). Data types become shared (not redefined per tool). HCatalog objects support partitioning and are HDFS-optimized (i.e. stored in the optimized ORD File Format)

It provides:

- Consistency of metadata and data models across tools (MapReduce, Pig, Hbase, and Hive)
- Accessibility: share data as tables in and out of HDFS
- Availability: enables flexible, thin-client access via REST API

To the end user, the data appears as a relational table, accessible by an ODBC/JDBC compliant query tool.

Solr (Google-like Search)

If more of a Google-like search interface is desired then a custom interface could be constructed to perform searches against the data indexed using custom jobs with Solr.

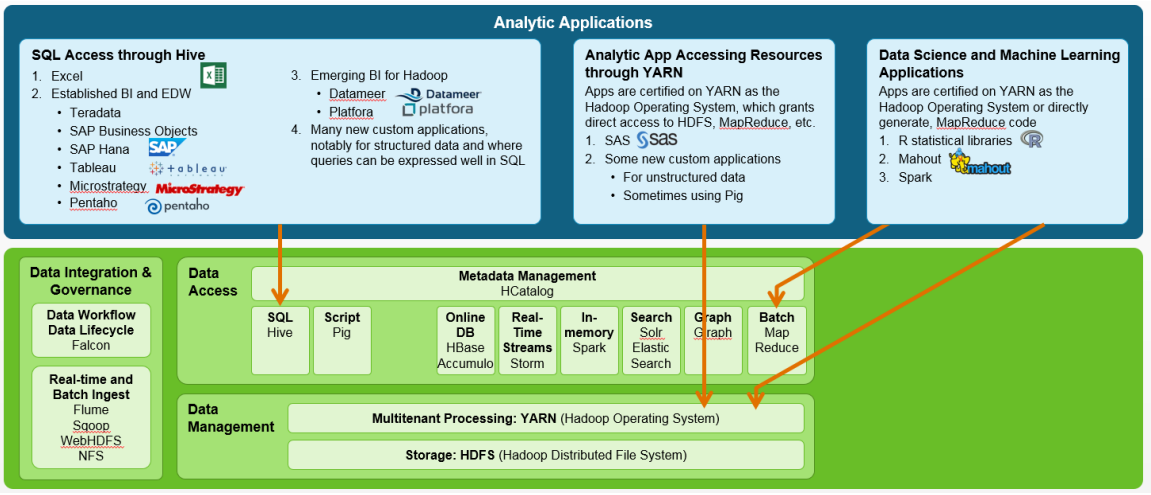
Providing access to the end user

SQL access

Many business users will prefer to access Hadoop data through SQL/Hive. Hive ODBC/JDBC Driver provides seamless integration with BI tools such as Excel, SAP Data Integration Services, MicroStrategy, and Tableau that efficiently maps advanced SQL functionality into HiveQL. Hortonworks provides an ODBC 3.52 standard compliant driver, supported on both Linux & Windows.

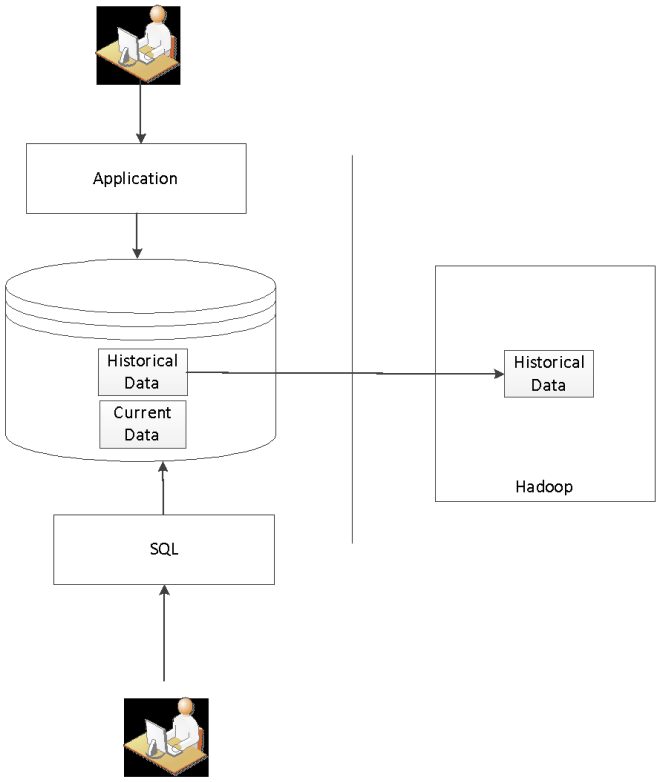
Statistical access

More statistically-inclined users may wish to access the data through SAS, R, Mahout or even the newer Spark Machine Learning libraries.



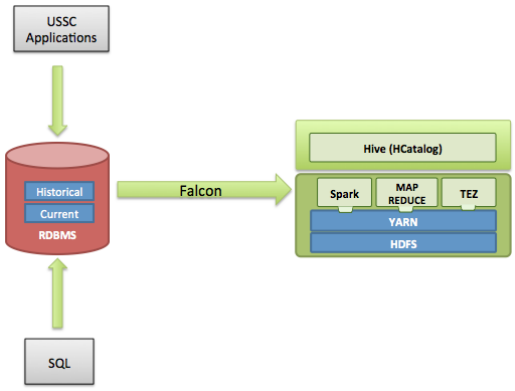
3. Data Tiering

    a. USCC wants to move out data from the RDBMS (relational database management system) and store it on a more economically appropriate storage tier. The difference in this use case is that the data needs to accessible via RDBMS engine as well. A sample high-level diagram of the desired Data Tiering solution is below.



b. Please describe how your solution can be used to meet this need including the technology components being used. Please sure to include components for:
   i. Data extraction
   ii. Data loading to Hadoop
   iii. Publishing the metadata to the end user
   iv. Providing access to the end user
   v. Ensuring that the data matches the source system, no changes in the level of data quality

Answer: Below is a diagram and explanation moving data from the RDBMS



Data Extraction and Loading: These two tasks would be handled by Apache Falcon, Apache Falcon is the Data feed processing and feed management system aimed at making it easier for end consumers to onboard their feed processing and feed management on Hadoop clusters. Under the hood Falcon would be calling and managing Sqoop which in turns utilizes a number of connectors for various RDBMS system in order to Ingest and Egress data.

Tiered Storage

With HDP 2.2, HDFS now provides the ability to utilize heterogeneous storage media within the HDFS cluster to enable Archival Storage: Utilize a set of storage dense nodes within the cluster to store less frequently accessed datasets.

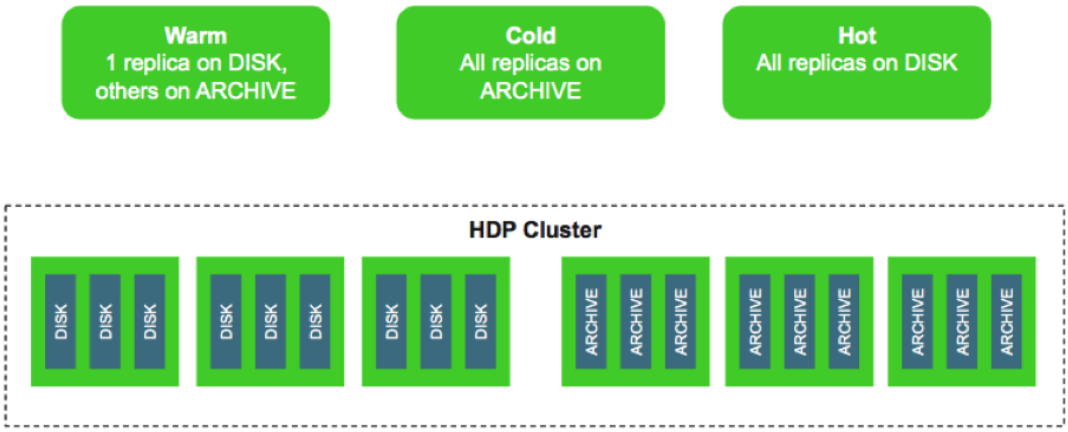
Each DataNode in HDFS is configured with a set of specific disk volumes as storage mount points on which HDFS files are persisted. With HDP 2.2, administrators can tag each volume with a Storage Type to identify the characteristic of storage media that represents the volume. For example, a mounted volume may be designated as an archival storage and another one as flash storage.

Storage Policies define the policy HDFS uses to persist block replicas of a file to Storage Types as well as the desired Storage Type(s) for each replica of the file blocks being persisted. They allow for fallback strategies, whereby if the desired Storage Type is out of space then a fallback Storage Type is utilized to store the file blocks. The scope of these policies extends and applies to directories, and all files within it.

Storage Policies can be enforced during file creation, and at any point during the lifetime of the file. For Storage Policies that have changed during the lifetime of the file, HDFS introduces a new tool called Mover that can be run periodically to migrate all files across the cluster to correct Storage Types based on their Storage policies.

Archival Storage Scenario
Over the lifetime of a dataset, the frequency of reads of a dataset in processing workloads decreases. That is, the dataset is deemed as “cold.” As the amount of data under storage grows, there is a need to optimize storage of such ‘cold’ datasets. An Archival storage tier, consisting of nodes with slow spinning high density storage drives and low compute power, provides cost efficiency for storing these cold datasets.

HDP 2.2 introduces an ‘ARCHIVE’ Storage Type and related Storage Policies – ‘Hot’, ‘Warm’, ‘Cold’.



e.g. A directory marked as 'WARM' by an administrator will have one replica stored on volumes marked as 'DISK', while the other replicas (two by default) will be stored on volumes (slower, cheaper drives) marked as 'ARCHIVE'.

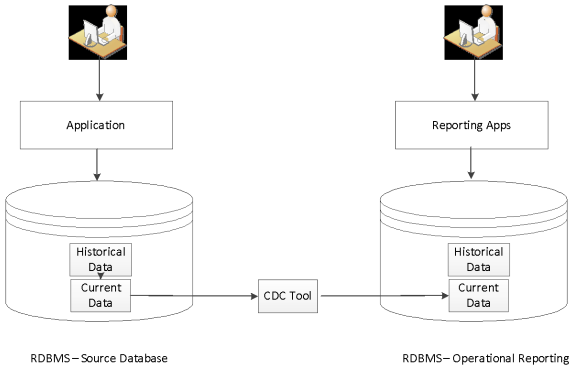
Publishing the metadata to the end user: The historical data extracted from the source USCC RDBMS system would be loaded into Hive, which is Hadoop's Data Warehouse component. All data ingested would have a clearly defined metadata in the format of SQL DDL that can easily be published to users. This metadata is stored in a component called HCatalog which Hive uses for processing SQL queries as well as it being used as the standard for all BI integration and external facing Analytical tools.

Providing access to the end user: Because the Hive/HCatalog metdata is exposed to external system traditional RDBMS systems (based on their corresponding connectors) can define external tables that point to Hive/Hcatalog and will process queries natively in Hadoop. This allows the implementation to be completely transparent to the user as the table is defined in the RDBMS store while the data and processing is executed on the Hadoop Cluster.

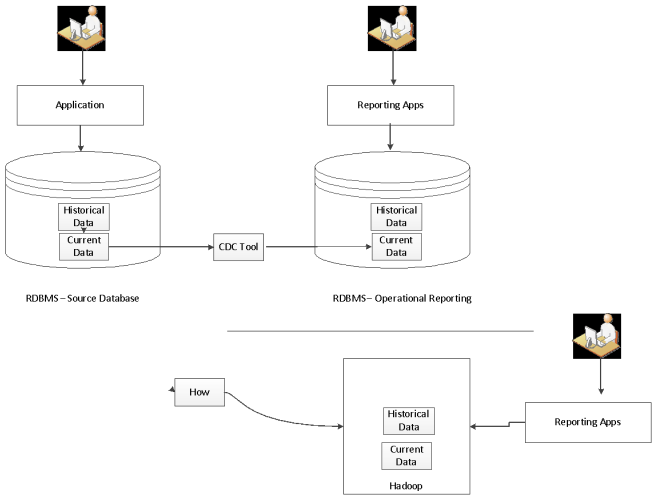
Ensuring that the data matches the source system, no changes in the level of data quality: Apache Hive is ANSI compliant and follows the SQL-92 standard. SQL Tables defined in Hive can have the same structure, Data types and lengths as those in the source RDBMS system. During the extract/load phase of Falcon and Sqoop all the data quality will be verified as data is transported and ingested automatically and any outliers will cause the process to raise flags in regards to schema and data type definitions that do not match, hence the data quality from the source system will be honored and preserved.

4. ODS Replacement
    a. USCC has a series of source application databases that are loaded into a single Operational Data Store (ODS) via a Change Data Capture process, Oracle GoldenGate. The ODS is used for both Operational reporting and Production database offloading. Loading

lag to the ODS is typically less than a minute. A sample high-level diagram of the desired ODS solution is below.



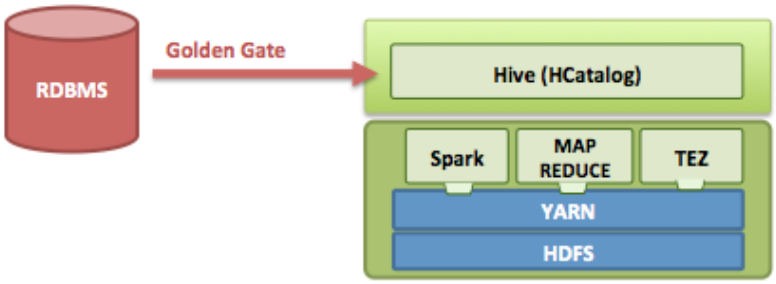
b. USCC wants to migrate the ODS to Hadoop based solution, but anticipates having to run a parallel ODS on a RDBMS and Hadoop for at least 6 months (see diagram below of potential parallel environments). The ODS must be a SOX complaint system. This provides a requirement of validating all data that is loaded is accurate and that no one can modify the data on the Hadoop side.



c. Please describe how your solution can be used to meet this ODS replacement including the technology components being used. Please sure to include components for:

i. Data extraction
ii. Data loading to Hadoop
iii. Publishing metadata to end user
iv. Providing access to the end user
v. Ensuring that the data is not modifiable on the Hadoop side
vi. Ensuring that the data matches the source system, no changes in the level of data quality

Answer:



For this solution USCC can continue using Golden gate as their primary CDC tool. Golden Gate would push data directly into Hive as we offer Hive Streaming Ingest capability with CDC which allows real-time streaming of data so the transition to Hadoop with this architecture would be seamless. All Data loaded into Hive will have common metadata exposed through HCatalog back to external users, applications and tools. Security around the data would be handled by Apache Ranger as it has native integration with all Hive for authorization policies across databases, tables and columns.

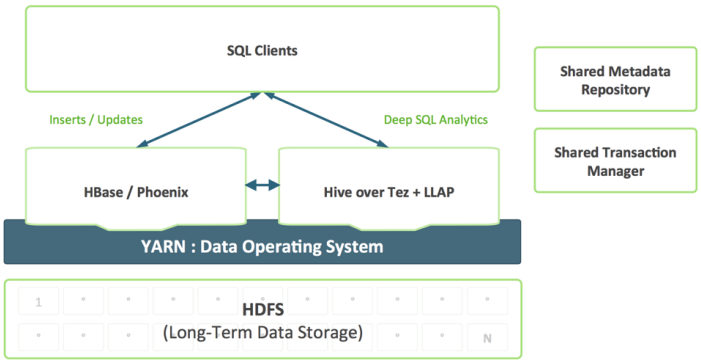
Hadoop as an ODS?
With the introduction of Phase #1 Transactions in HDP 2.2, Hive now supports INSERTs, UPDATEs and DELETEs (albeit not on the scale that an RDBMS OLTP system is capable of today).

To that end, we see our customers using these parts today to build applications with deep analytics, for example a very common pattern we see includes:

- Using HBase as the online operational data store for fast updates on hot data such as current partition for the hour, day etc.
- Executing operational queries directly against HBase using Apache Phoenix.
- Aging data in HBase to Hive tables using standard ETL patterns.
- Performing deep SQL analytics using Hive

We see four major development areas to help realize the vision of intelligent applications:

1. A Unified SQL Layer with Hive
Developers building SQL applications should not have to choose between different SQL solutions, each with its own strengths and weakness. We envision a unified SQL layer, enabled by Hive's support for SQL:2011, that transparently uses the appropriate engine based on the query access pattern.



This combination provides a single SQL dialect and single connector. Data architects and DBAs can determine where data should be stored based on usage patterns without burdening user applications with the need to connect to multiple systems.

2. Improving HBase as an Operational Store

HBase is rapidly maturing as an operational store and will be able to take on more and more demanding workloads. In the past year, HBase has added a SQL interface, secondary indexing and high availability. These features will continue to mature, and in addition, HBase will add additional enterprise-grade features like multi-table, cross-datacenter transactions and more.

3. Shared Metadata Catalog and Transaction Manager

Data created in HBase should be automatically visible in Hive and vice-versa. This capability renders data sharing between online and analytical completely trivial. A shared transaction manager allows Hive's new ACID feature and multi-table HBase transactions to work together seamlessly.

4. YARN-enabled Mixed Workload Support

Developing a closed-loop analytics system requires effective combination of operational and analytical workloads in a multi-tenant manner. With YARN we can effectively create a single-system by leveraging resource isolation and workload management primitives in YARN to support different forms of access to data. Slider makes use of these when it deploys HBase into YARN, while Hive LLAP & Tez are native YARN applications, thereby simplifying the process of running a closed-loop analytical system according to a predictable SLA.

Hive and HBase - Better Together

Enterprises are using already existing technologies available in HDP such Apache HBase, Apache Hive, Apache Phoenix etc. to deal with fast updates to current data and analytics over vast array of data-sets, all stored in HDFS to effect a closed-loop analytics system. We hope to leverage the same integration patterns to provide a seamless experience for customers by making Apache HBase and Apache Hive better – better together, rather than net new technologies for users to understand and consume.

	Today	Future
SQL Layer	• Hive for Deep Analytics OR • Phoenix for High TPS	Unified SQL layer offering Deep Analytics and High TPS
Multi-Row Transactions	Not Supported	Supported through HBase
Cross-Datacenter Transactions	Not Supported	Supported through HBase
Data Sharing	Manually mapped using external tables	Automatic and Transparent
Mixed Workload	Manually tuned and monitored	Automated through YARN and Slider

Data Extraction and Loading: These 2 tasks would be handled by Apache Falcon, Apache Falcon is the Data feed processing and feed management system aimed at making it easier for end consumers to onboard their feed processing and feed management on Hadoop clusters. Under the hood Falcon would be calling and managing Sqoop which in turns utilizes a number of connectors for various RDBMS system in order to Ingest and Egress data.

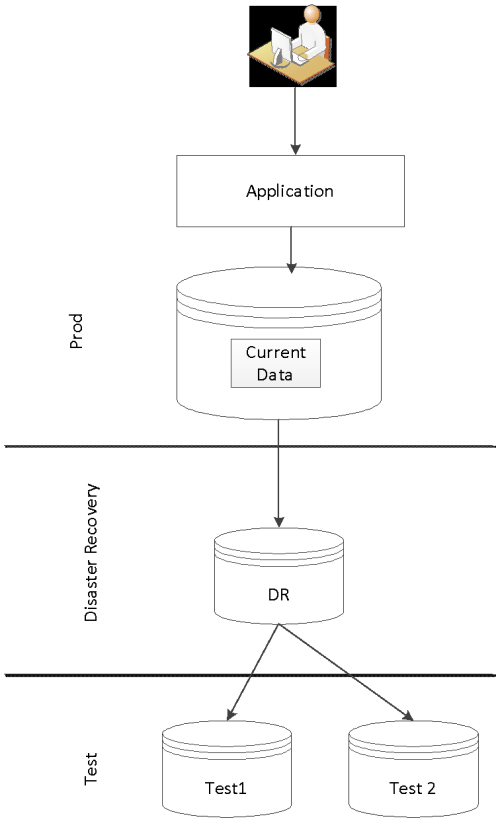
Publishing the metadata to the end user: The historical data extracted from the source USCC RDBMS system would be loaded into Hive, which is Hadoops Data Warehouse component. All data ingested would have a clearly defined metadata in the format of SQL DDL that can easily be published to users. This metadata is stored in a component called HCatalog which Hive uses for processing SQL

queries as well as it being used as the standard for all BI integration and external facing Analytical tools.

Providing access to the end user: Because the Hive/HCatalog metdata is exposed to external system traditional RDBMS systems (based on their corresponding connectors) can define external tables that point to Hive/Hcatalog and will process queries natively in Hadoop. This allows the implementation to be completely transparent to the user as the table is defined in the RDBMS store while the data and processing is executed on the Hadoop Cluster.

Ensuring that the data matches the source system, no changes in the level of data quality: Apache Hive is ANSI compliant and follows the SQL-92 standard. SQL Tables defined in Hive can have the same structure, Data types and lengths as those in the source RDBMS system. During the extract/load phase of Falcon and Sqoop all the data quality will be verified as data is transported and ingested automatically and any outliers will cause the process to raise flags in regards to schema and data type definitions that do not match

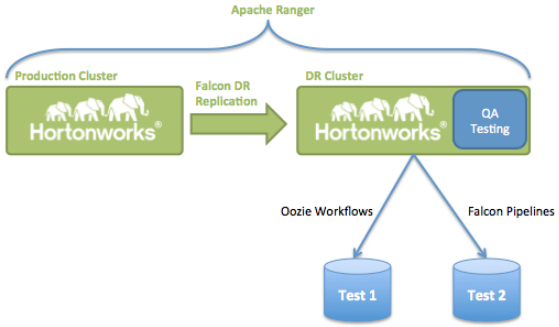
5. Testing Use Case
    a. For USCC's non-production environments, specifically testing, the databases are built as clones off of the Disaster Recovery ("DR") site and then obfuscate the data.



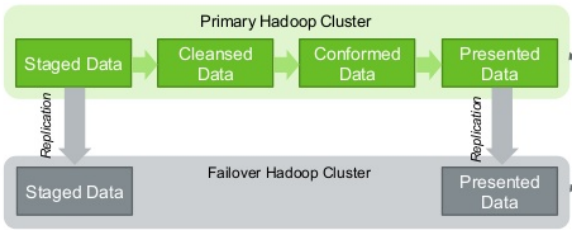
    b. USCC typically has 4-5 full copies to be used for testing purposes. In general, the full copies used in testing are not from the same point in time. Please describe how your solution can be used to meet this need including the technology components being used.

i. Assumptions around how data is replicated to Disaster Recovery Site
ii. Data propagation to a test instance of data
iii. Obfuscation of the data within the test instance
iv. Be able to support a test instance that is getting new data from the test applications
v. Assume that the data is in a combination of Hbase, Hive, flat file and SQL on Hadoop solution.

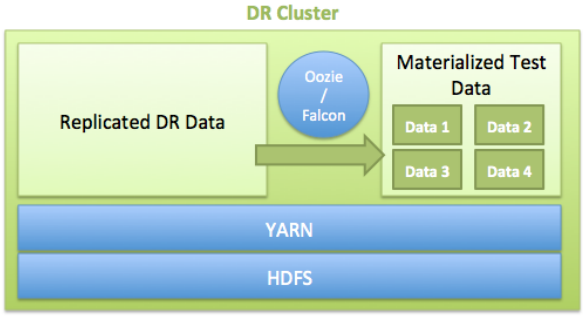
Answer:



Assumptions around how data is replicated to Disaster Recovery Site: Cluster replication is a functionality that Apache Falcon offers out of the box. Apache Falcon is a framework for simplifying and orchestrating data management and pipeline processing in Apache Hadoop. It enables automation of data movement and processing for ingest, pipelines, replication and compliance use cases. Falcon also leverages its integration with YARN—the architectural center of Hadoop—to centrally manage the cluster's data governance, maximize data pipeline reuse and enforce consistent data lifecycles.

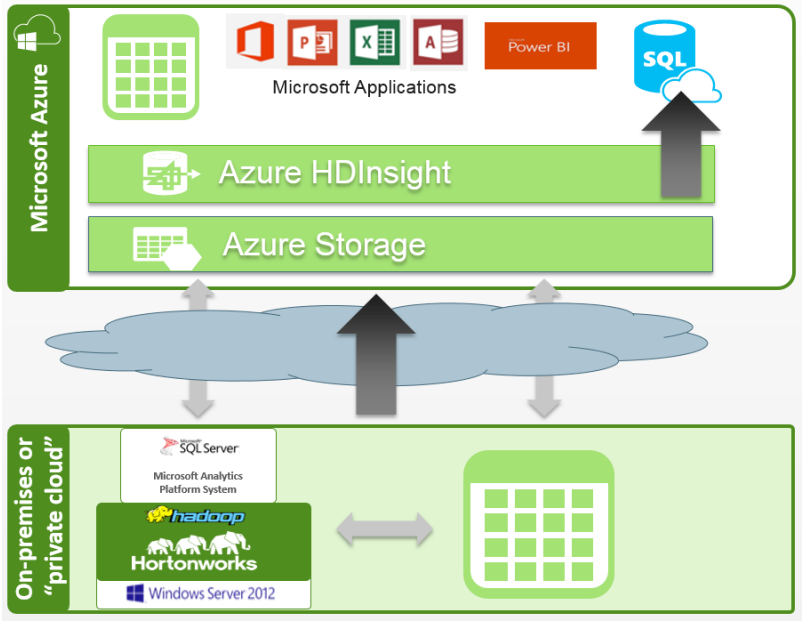


Data propagation to a test instance of data: Test instances of data can reside within USCC RDBMS systems or the DR/Failover cluster can be leveraged for testing as no production grade work is being performed. Data propagation for the DR cluster testing can easily be performed with a combination of Oozie Workflows or in a more automated fashion Falcon Pipeline definitions. Apache Falcon allows Hadoop administrators and developers to build complex data transformations out of multiple component tasks. This allows for greater control over complex jobs and also makes it easier to repeat those jobs at predetermined intervals. Falcon would be a great fit to build automated and scheduled data movement from source DR data into Materialized Marts that can further be used for testing as outlined in the image below:



DR in the Cloud? Giving you Options.

The use case didn't mention whether the DR and/or Test copies would remain on-prem but we have seen significant interest in backing up data into cheap Blob storage (e.g. AWS or Azure).



The above diagram demonstrates using Azure blob storage as low cost, offsite backup. With MS Azure, you have the choice to run HDP and HDInsight to power analytics on your data in the cloud.

- **Automated data upload & backup**
  Use Falcon to schedule data load rules, push data based on business needs
- **Global aggregation**
  Capture data centers around the world
  Run Hadoop local to a DC, or aggregate across DC's to query the entire dataset
- **Seamless transfer to other storage**
  Leverage Azure SQL DB & Azure storage as sources or destinations data

Obfuscation of data for testing

As data is Materialized by Falcon and Oozie additional obfuscation policies can be defined for the type of data access that will be used for testing (HDFS, HBase. Hive). Apache Ranger offers flexible and fined grained access controls across all HDP components and can be embedded into data movement as Oozie and Falcon build materialized sets.

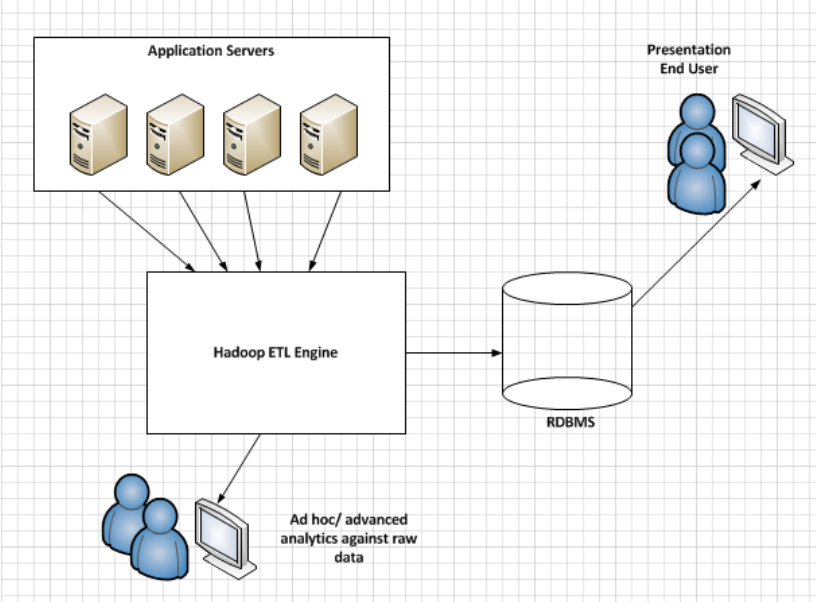
Falcon Recipes

In order to simplify Falcon configuration , Hortonworks is proposing the creation of Falcon 'Recipies'. A Falcon recipe is a static process template with parameterized workflow to realize a specific use case. Recipes are defined in user space. The user will provide a properties file with name value pairs that are substituted by falcon before scheduling it. Falcon translates these recipes as a process entity by
replacing the parameters in the workflow definition.

Some example use-cases might be:

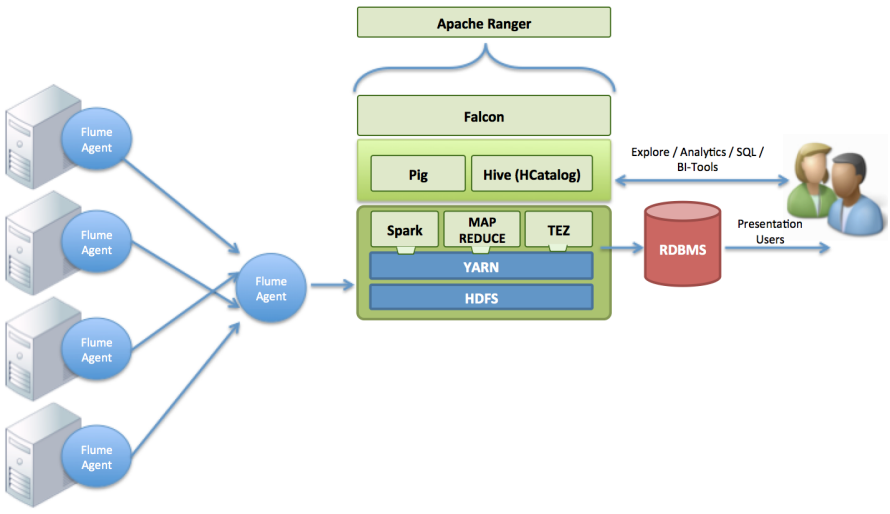
* Replicating directories from one HDFS cluster to another (not timed partitions)
* Replicating hive metadata (database, table, views, etc.)
* Replicating between HDFS and Hive - either way
* Data masking etc.

6. ETL Offload / Analytics use case
    a. USCC has a dozen application servers generating CSV data consisting of 2 billion records per day. These records needed to be aggregated on multiple dimensions, and the results need to be exported to an RDBMS (Oracle or Postgresql). The aggregated data must be accessible to the end user in < 4hours. This case also requires the retention of 90 days of raw data, at ~1TB per day in the Hadoop environment. This raw data must be accessible for ad-hoc analytics requests and exploratory analytic tools, as well as for the generation of new dimensions, without impacting the performance of the data aggregation jobs. A representative diagram of this case is immediately below.



b. Please describe how you solution can meet the following needs:
   i. Data extraction from application servers
   ii. Data loading into Hadoop
   iii. A SQL like environment to execute data aggregations against raw input, meeting required run times outlined above
   iv. A mechanism to control data access and retention of the raw data, including data expiration, and restricting / auditing of data access
   v. Ability/ tools for Ad-hoc analytics of the raw data, without impacting the performance of the data aggregation workload.

Answer:



Data Extraction and Loading from Application Servers: For the collection of the log data form disparate Application systems Apache Flume is a great candidate. Apache Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of

streaming data into the Hadoop Distributed File System (HDFS). It has a simple and flexible architecture based on streaming data flows; and is robust and fault tolerant with tunable reliability mechanisms for failover and recovery. Flume Agents would be deployed on the application servers and would collect in near-real time all log events as they are generated. Depending on the number of server and agents deployed USCC could adopt a fan in architecture with agents so that a collector agent would buffer together these events and ingest into HDFS.

A mechanism to control data access and retention of the raw data, including data expiration, and restricting / auditing of data access: Apache Falcon would handle all Processing once the data lands in hadoop ingested from Flume. It would handle any transformations using Pig (ETL scripting language) to cleanse the data for the presentation layer later while also persisting and tagging all versions with retention policies. Once data is cleansed/transformed Falcon would apply Metadata definitions (in the form of SQL DDL) for AdHoc SQL queries and additional tooling integration. Because metadata is in a centralized place all Enterprise analytics and BI tools can consume the data in a standard structure.

In order to apply authorization policies and monitor audits on the ingested data Apache Ranger would be used to enforce complex, role based access policies. Apache Ranger offers a centralized security framework to manage fine-grained access control over Hadoop data access components like HDFS, Apache Hive, Apache HBase, etc... Using the Apache Ranger console, security administrators can easily manage policies for access to files, folders, databases, tables, or column. These policies can be set for individual users or groups and then enforced within Hadoop. Security administrators can also use Apache Ranger to manage audit tracking and policy analytics for deeper control of the environment. The solution also provides an option to delegate administration of certain data to other group owners, with the aim of securely decentralizing data ownership.

A SQL like environment to execute data aggregations against raw input: As part of the Falcon processing definitions the defined schema's would be created in the form of SQL DDL tables and can be used across all SQL compliant interfaces as well as integrated with RDBMS.

Ability/ tools for Ad-hoc analytics of the raw data, without impacting the performance of the data aggregation workload: Hortonworks Data Platform certifies with majority of BI/ Analytic tooling vendors as well as supports Spark and R for analytical workloads. With HDP 2.2 we have the capability of labeling certain nodes for certain workloads like those of Spark. With a defined label and Spark defined workload USCC would be able to run high-performance in memory analytics without impacting the ingestion processes and degrading their performance.

7. Additional Hadoop Questions
    a. Vendor Overview
        i. Please provide a brief company overview.

Answer: 24 key engineers from the original Hadoop development team at Yahoo founded Hortonworks! Today we have over 200 engineers and are proud that every Hortonworks developer is an ASF or OpenStack Foundation Contributor. We lead and have contributed more lines of code to Apache Hadoop than any other.

We are responsible for more than half the code found across all major branches.

We partner with key industry players such as Microsoft, RedHat, SAS, SAP, HP, EMC, Pivotal and Teradata to help ensure Hadoop is easy to consume & use as part of broader big data solution architectures. Moreover, our commitment to a community-driven development model (via Apache Software Foundation) enables our partners & customers to participate directly & transparently in the

roadmap & development process of enhancing Hadoop for specific solution architectures & enterprise use cases.

This is a unique moment in time. Fueled by open source, Apache Hadoop has become an essential part of the modern enterprise data architecture and the Hadoop market is accelerating at an amazing rate.

The impressive thing about successful open source projects is the pace of the "release early, release often" development cycle, also known as **upstream** innovation. The process moves through major and minor releases at a regular clip and the **downstream** users get to pick the releases and versions they want to consume for their specific needs.

In the case of Apache Hadoop platforms like the Hortonworks Data Platform (HDP), we see the consumption of dozens of specific versions of Apache Software Foundation (ASF) projects:

Assembling a complete platform like HDP requires choosing the right stable version of Apache Hadoop as the foundation and then integrating, and packaging the optimal versions of all the other ASF components into a well-tested, certified data platform.

Working within the community for the enterprise

Since we are committed to delivering HDP completely in the open, we introduce enterprise feature requirements into the public domain and we code to address those requirements. We code and we contribute everything back to the wide array of ASF projects in HDP. Why? Because we want the Hadoop market to work at scale, and in order to make this happen, we know code is king, so we practice what we preach. We contribute everything.

Specifically, we:

1. **Innovate** *Innovate within existing ASF projects* to accelerate enterprise-focused innovation. Our work on Apache Hadoop YARN, HDFS, and the Stinger Initiative/Hive are great examples.

2. **Incubate** *Identify and create new ASF projects* that address security, management, operations, and other enterprise needs; Apache Ambari, Apache Falcon, Apache Ranger, Apache Knox, Apache Tez, and Apache Slider are great examples.

3. **Acquire and Contribute** *Acquire innovative companies and contribute the IP to the ASF* as an Apache incubator project. We acquired XA Secure in 2014 and flipped this commercial software for comprehensive security into open source as Apache Ranger.

4. **Partner and Deliver** *Establish joint engineering relationships to accelerate Enterprise Hadoop innovation*. Our deep joint engineering work with Microsoft, HP, SAS, Pivotal, Red Hat, Teradata and others are great examples. Microsoft's recent launch of its Azure HDInsight service on Linux is a great example of what comes from joint engineering.

5. **Rally the Ecosystem** *Found enterprise-focused initiatives that rally end users and vendors* towards common goals. The Stinger initiative, the Data Governance initiative and the Open Data Platform initiative are great examples.

The enterprise-focused initiatives are an important element of our approach. The Stinger Initiative has successfully rallied contributions from hundreds of developers across dozens of companies in order to address SQL in Hadoop needs for the enterprise. The Data Governance Initiative was

recently formed with Aetna, Merck, Target and SAS to address the data stewardship, lineage, lifecycle management, and privacy issues that are increasingly important.

Open Data Platform Initiative (ODP)
The Open Data Platform initiative (ODP), announced in February 2015, aims to rally enterprise end users and vendors alike around a well-defined common core platform (the ODP Core) against which big data solutions can be qualified.

How does the ODP relate to the ASF?
All **upstream production** happens within the ASF projects according to the ASF governance model. Individuals working for ODP member companies are encouraged to participate and contribute to ASF projects as they see fit and in accordance to ASF processes. Since Hortonworks engineers do all of their coding in these ASF projects, we're more than happy to help newcomers learn the Apache way and contribute.

The ODP, on the other hand, is focused on enabling **downstream consumption** of a common set of Hadoop-related components, and more importantly, **specific versions** of those components. Harmonizing the broader market around Apache Hadoop version 2.6 and Apache Ambari 2.0, for example, will help simplify the onboarding of manageable, YARN-based solutions that can ride atop the common core platform.

Increasing the compatibility among Hadoop-based platforms and solutions will free up the broader big data ecosystem to focus on more important things such as data-driven applications that deliver proactive insights for the business. Innovation will advance even faster in the market with all the ODP members building upon the same downstream Apache Hadoop kernel- Apache Hadoop, Yarn and Apache Ambari.

Modern platform standards are defined by open communities:
At Hortonworks, our founding belief is that innovation and adoption of platform technologies like Hadoop is best accomplished through collaborative open source development under the *governance model of an entity like the Apache Software Foundation (ASF)*.

In order to enable a data platform like Hadoop to be easy to use and enterprise-grade, you don't go it alone. You do it by working with your customers and the broader ecosystem to enable:

- *data architects* to deeply integrate existing systems with Hadoop; *developers, data workers, and analysts* to build applications quickly and easily; and
- *operators and security administrators* to deploy, manage, secure, and govern the platform and the applications deployed on it in a consistent way.

Our approach to the market is about enabling our customers to embrace Hadoop in a way that makes sense for their business. It's about enabling our partners in a way that drives joint value from the alliance in a way that's respectful to each other in the process. And it's about rallying a community in a way that drives innovation around shared goals.

ii. Do you prefer to do business with USCC directly or for us to work with channel partners? If the preference is to work with channel partners, are there specific channel partners that already have relationships with USCC?

Answer: Hortonworks does the majority of its business direct with customers but if desired has channel partners like SAP, HP, Microsoft, Teradata, SHI and others that can be leveraged.

iii. A copy of USCC's Master Agreement ("MSA") is attached. It is USCC's assumption that any award based on this RFI is in accordance with all of the terms and conditions contained within this MSA. If Vendor has an existing agreement in place with USCC, USCC reserves the right to amend such agreement at its sole discretion to align with the terms and conditions contained within the attached MSA.

Answer: See our attachment for the comments on the MSA

iv. What percentage of Vendor employees performs customer support?

Answer: Hortonworks employs over 200 engineers covering the complete support organization and engineering. Our engineering and support organization are combined to ensure that that any issue if it arises, it can quickly be escalated to the engineering (Tier 3/4) for resolution.

- Tier 1/2 Technical Support 60+ people
- Tier 3/4 Engineering 140+ people

That represents 33% of our total company. In addition, Hortonworks has over 175 Solution Engineers, Platform Engineers, and Consultants that perform support, architecture, and implementation services.

v. What percentage of Vendor employees performs product development?

Answer: Hortonworks currently has over 175 engineers that do product development. Our Tier 4 engineers are the developers and committers to Apache and/or the OpenStack Foundation.

vi. What is the percentage of your customers are in the Telecom industry?

Answer: Hortonworks has been the standard for every major Telecom in the United States and Canada. Verizon Wireless, AT&T, Sprint, T-Mobile, Rogers Communications, Comcast, Tellus Communications, Blackberry, Motorola Solutions, and Neustar have all selected Hortonworks for their data lake.

vii. How much does your company contribute to the Apache Hadoop open source projects? Please describe in detail your company's contributions.

Answer: Hortonworks contributes 100% of the development of our code back to the Apache Software Foundation. We have contributed and committed over 1MM lines of code and have committer status for all of the important Apache Projects regarding Hadoop. Our strategy and philosophy is never to lock a customer in with proprietary code. Whether we develop code in house or through acquisition, every line of code is contributed back. An example of that was the acquisition of XA Secure. We took the proprietary solution and gifted it to Apache, which is now Apache Ranger.

Below is a chart of the number of committers: Source: Apache Software Foundation. As of 1/10/2015.

Apache Project	Committers	PMC Members
Hadoop	**27**	21
Pig	5	5
Hive	18	6
Tez	16	15
HBase	6	4
Phoenix	4	4
Accumulo	2	2
Storm	3	2
Slider	11	11
Falcon	5	3
Flume	1	1
Sqoop	1	1
Ambari	36	28
Oozie	3	2
Zookeeper	2	1
Knox	13	3
Ranger	11	n/a
TOTAL	**164**	**109**

b. Vendor Support / Licensing Model

i. Does your support staff provide 24 hour support (follow the sun)?

Answer: Hortonworks has 5 regional support hubs: Santa Clara, CA, Raleigh, NC, London, UK, Bangalore India and Sydney, AU. These centers provide 7x24 follow the sun support coverage.

ii. Please provide detailed breakdown of customer support SLAs.

Answer:

Support Level	HDP Enterprise HDP Enterprise Plus HDP Enterprise Premier
Access via	Phone, Web
Hours	24x7 (S1 only) 24x5 (S2-S4)
S1 Initial Response	1 hour
S2 Initial Response	4 business hours
S3 Initial Response	8 business hours
S4 Initial Response	Next business day

iii. Does your support team have an escalation process? If so please describe in detail how a customer could escalate a support call.

Answer: Case management is critical to Hortonworks. The Hortonworks team is committed to getting resolution to issues as quickly as possible. If your needs or expectations are not being met on any open support case, please escalate in the following order:

- **1st Point of Contact** – Technical Support Engineer working the Case
- **2nd Point of Contact** – Rick Morris, Director of Technical Support
  The Director of Support will ensure the case is being treated with the highest priority and that the proper resources from support, engineering, product management, professional services are involved in resolving the issue
- **3rd Point of Contact** – Vice President of Services
- **4th Point of Contact** – Vice President of Engineering

VP's of Services and Engineering will be notified of all S1 production down issues that are not resolved within 4 hours.

Hortonworks uses both severity and priority to help manage your cases and to ensure timely resolution based on our business needs. The customer chooses the severity and priority.

- Severity is based on the objective definitions as presented below:
- Priority is more of a subjective measure and helps you to communicate the urgency of the issue based on your business needs and let's us know exactly when you need our assistance

Severity and Priority Definitions
Upon receipt of a properly submitted Error, Hortonworks shall prioritize it in accordance with the guidelines below. Error Severity may be re-evaluated upon submission of a workaround.

Severity 1 (S1) – An S1 is a major production Error within the Software that severely impacts the Customer's use of the Software for Production Purposes, such as the loss of production data or where production systems are down or not functioning and no work around exists. Hortonworks will use continuous efforts on a 24x7 basis to provide a resolution for any S1 problem as soon as is commercially reasonable. All S1 Errors must be called in via the Hortonworks Support Line.

Severity 2 (S2) – An S2 is a production Error within the Software where the Customer's system is functioning for Production Purposes but in a degraded or restricted capacity, such as a problem that is causing significant impact to portions of the Customer's business operations and productivity, or where the Software is exposed to potential loss or interruption of service. Hortonworks will use reasonable efforts during its normal hours of operation to provide a resolution for any S2 Errors as soon as is commercially reasonable.

Severity 3 (S3) – An S3 is a medium-to-low impact Error that involves partial and/or non-critical loss of functionality for Production Purposes or Development Purposes, such as a problem that that impairs some operations but allows the Customer's operations to continue to function. Hortonworks will use reasonable efforts during its normal hours of operation to provide a resolution for any S3 Error. All Developer level support tickets shall maintain a level of S3.

Severity 4 (S4) – An S4 is a low priority request for information where there is no impact to business operations.

Priority Level Definitions
Priority 1 (P1) – Immediate attention and immediate resolution is required
Priority 2 (P2) – High priority issue, same day response and next day resolution acceptable

Priority 3 (P3) – Medium priority issue, next day response and resolution in days acceptable
Priority 4 (P4) – Low priority issue, next day response and next week resolution acceptable

Severity	Target Initial Response Time Per Support Policy	Target Case Status Update Time	Target Workaround Time	Target Resolution Time
S1 - Production Down	1 hour	Every 2 hours	Within 4 hours	24 hours
S2 – Core Function Inoperative	4 hours	Three times a week	Within 72 hours	6-8 weeks (future maintenance release)
S3 – Minor Feature Inoperative	8 hours	Twice a week	5 to 8 days	6 to 12 months (consider for future minor or major release)
S4 – Request for Info	Next day	Once a week	n/a	n/a

iv. How can customers contact support web, phone, online chat, etc.?

Answer: All designated support contacts can access support through the Hortonworks Support Portal and Hortonworks Helpline.

v. Does a support contract entitle us to all components/features or is it a la carte? Please describe in detail (including pricing) all components/features of your solution

Answer: There are three levels of Support from Hortonworks. The attached graphic describes the differences between the offerings. Based on the Support level, the contract entities USCC to the following support components.

Additionally, each subscription support offering provides support for the HDP components :

	Enterprise Plus	Enterprise	Enterprise Premier
Apache Hadoop in HDP (HDFS, YARN, MapReduce)	•	•	**Inclusive of Enterprise Plus** **Additional Benefits:** **Resolution** • Designated support engineer • Weekly meetings to review support tickets • Coordinated triage • Fast-track escalations • Root-cause analysis of Severity 1 incidents **Mitigation** • Regular briefings on releases & roadmap • Quarterly activity reviews • Onsite deployment support • Planning assistance for critical milestones • Active coordination of 3rd party relationships
Apache Tez in HDP	•	•	
Apache Hive in HDP	•	•	
Apache Pig in HDP	•	•	
Apache Sqoop in HDP	•	•	
Apache Flume in HDP	•	•	
Apache Mahout in HDP	•	•	
Apache Ambari in HDP	•	•	
Apache Oozie in HDP	•	•	
Apache Falcon in HDP	•	•	
Apache Knox in HDP	•	•	
Apache HBase in HDP	•	•	
Apache Phoenix in HDP	•	•	
Apache Accumulo in HDP	•		
Apache Storm in HDP	•		
Apache Ranger in HDP	•		
Apache Spark in HDP	•		
Apache Kafka in HDP	•		
Support for HDP Search is also available as an add-on to an Enterprise or Enterprise Plus subscription support offering			

Pricing is based on 4 nodes and annual basis.

Enterprise Subscription – 4 nodes - $10,000 annual
Enterprise Plus Subscription – 4 nodes - $18,000 annual
Enterprise Premier (100 node min) – 4 nodes - $25,000 annual

vi. Does your company have managed services offerings? If so, please describe the capabilities (including pricing) for this.

Answer: Hortonworks does not have managed services but it is offered through our partner network at companies like Microsoft, Rackspace and CSC.

vii. Do you work with 3rd party service and support providers? If so who are they?

Answer: Hortonworks has deep relationships with numerous 3rd party service and support providers. The complete list can be found at www.hortonworks.com/partners

viii. How does your company charge for support? Please describe your licensing model in detail.

Answer: Hortonworks is model is based on an annual subscription providing access to product updates, support and self paced online content library (training)

There are three levels of Support from Hortonworks. The attached graphic describes the differences between the offerings.

Additionally, each subscription support offering provides support for the HDP components :

	Enterprise Plus	Enterprise	Enterprise Premier
Apache Hadoop in HDP (HDFS, YARN, MapReduce)	•	•	**Inclusive of Enterprise Plus** **Additional Benefits:** **Resolution** • Designated support engineer • Weekly meetings to review support tickets • Coordinated triage • Fast-track escalations • Root-cause analysis of Severity 1 incidents **Mitigation** • Regular briefings on releases & roadmap • Quarterly activity reviews • Onsite deployment support • Planning assistance for critical milestones • Active coordination of 3rd party relationships
Apache Tez in HDP	•	•	
Apache Hive in HDP	•	•	
Apache Pig in HDP	•	•	
Apache Sqoop in HDP	•	•	
Apache Flume in HDP	•	•	
Apache Mahout in HDP	•	•	
Apache Ambari in HDP	•	•	
Apache Oozie in HDP	•	•	
Apache Falcon in HDP	•	•	
Apache Knox in HDP	•	•	
Apache HBase in HDP	•	•	
Apache Phoenix in HDP	•	•	
Apache Accumulo in HDP	•		
Apache Storm in HDP	•		
Apache Ranger in HDP	•		
Apache Spark in HDP	•		
Apache Kafka in HDP	•		

Support for HDP Search is also available as an add-on to an Enterprise or Enterprise Plus subscription support offering

Pricing is based on 4 nodes and annual basis.

Enterprise Subscription – 4 nodes - $10,000 annual
Enterprise Plus Subscription – 4 nodes - $18,000 annual
Enterprise Premier (100 node min) – 4 nodes - $25,000 annual

ix. Please describe how support works with 3rd party components. (Example, between you and a hardware vendor and an o/s vendor). Please describe the methodology and support options.

Answer: Hortonworks has 3rd party relationships with Microsoft, Teradata, HP, EMC, Hitachi, Pivotal, SAP and others. We have support and development relationships with most.

x. Please describe your software development cycle. How frequently are new versions of your software being released?

Answer: Hortonworks uses agile development has one major release and 4 to 5 minor (maintenance releases) per year.

xi. Does your company make available a software development roadmap to licensed customers? How frequently?

Answer: Hortonworks makes our roadmap available to customers as required. Customers that are part of our Customer Advisory Board or industry consortiums also have input into our forward-looking roadmap.

xii. What OS does your product support? Please provide an OS compatibility matrix.

Answer:

OS	Version	Limitations
Red Hat Enterprise Linux	5.x/6.x	64 bit
CentOS	5.x/6.x	64 bit
Oracle Linux	5.x/6.x	64 bit
SUSE Linux Enterprise Serve	11	64 bit, SP1 only
Ubuntu	12.04	64 bit. Forthcoming HDP 2.2 only.
Windows Server	2008 R2 or 2012	64 bit.

xiii. Does your solution include hardware? If so please provide details on hardware.

Answer: Hortonworks Data Platform is a software solution. We have partnerships with EMC Isilon, HP, and many others to provide Hardware based solutions. Please refer to our www.hortonworks.com/partners for the complete list.

xiv. What hardware/os certification is done for your Hadoop distribution?

Answer: We certify against the Operating Systems listed in the table in xii. For hardware, Hortonworks has partnerships with HP, Cisco, and SuperMicro but can be run on any commodity hardware. We have customers running on IBM, Dell, HP, Cisco, SuperMicro and many others.

Hortonworks has the recommended guidelines for servers.

Master Nodes – *NameNode, Resource Manager , HBase Master*
Dual Intel Xeon E5-2650v2 (8c) or E5-2660v2 (10c) Processors
128GB or 256GB RAM per chassis
4+ – 1TB NL-SAS/SATA Drives RAID10+ Spares

Worker Nodes – DataNode, Node Manager and Region Server
Dual Intel Xeon E5-2650v2 (8c) or E5-2660v2 (10c) Processors
12 – 1-4 TB NLSAS/SATA Drives
128GB RAM or 256GB RAM

xv. What type of hardware delivery patterns (virtualized, custom designed, appliance, standard building blocks) do you support? If you partner with a Partner, please provide who that is.

Answer: The majority of our customer base (90%) is on physical commodity hardware with direct attached storage. We are seeing some customers begin to test Scale Out NAS from companies like Isilon as a back end to Hortonworks Data Platform. We have an engineering based partnership with Isilon to enable this delivery pattern if desired.

The rest of the customers are deployed in the cloud via services from Microsoft Azure, Rackspace, Google, AWS and others or they are virtualized inside the data center. Most of the virtualized clusters are for Test/Dev.

A very common pattern we see is virtualized test and dev clusters and physical Production and Disaster Recovery.

xvi. What is most common hardware/os implementation for your distribution? What is the development hardware/os stack for your distribution.

Answer: Hortonworks most commonly sees Redhat or CentOS as the Operating system because that is the company's standard. We support any of the OS listed in the table in xii. From a hardware perspective we see primarily HP, Cisco UCS, and Dell X86 Hardware.

xvii. Please describe testing process used for hardware/os compatibility

Answer: One of the critical aspects of our development philosophy is applying enterprise rigor and leadership within the community. We execute more than 4,500 test cases (and growing) on a daily basis for each and every build of HDP across a number of Linux distributions including CentOS, RedHat, Ubuntu, SuSe and Debian and we are the ONLY company providing Hadoop on Windows. When we uncover a defect, we can assist the community in finding this rapidly and addressing it as quickly as possible. In addition, our engineering team has deep experience not only within the open source environment but also decades of experience with enterprise software. This means we are driving consistency across multiple individual projects within the open source

community like ensuring that fundamental expectations of enterprises around security are addressed. Examples include: ensuring that passwords are not stored in clear text, processes do not run as 'root', and more...

xviii. Does your product support cloud based implementations? Please provide companies that provide information regarding who provides this service. If you provide cloud based solutions, please provide details about the capabilities.

Answer: Our platform is fully supported in the cloud. It can be brought to a cloud provider of your choice or if desired we have partnerships with Microsoft, Rackspace, Google Cloud Platform, and others. Please refer to www.hortonworks.com/partners for complete details.

xix. If cloud based architectures are supported, can a private implementation be extended by adding cloud resources? Please describe in detail how this is architected and any limitations that exist.

Answer: We do see customers successfully implementing hybrid models, but we currently do not advocate spanning a cluster across multiple data centers or on-premise/cloud. The whole premise of Hadoop is to recognize that data will grow to such significant mass that moving it around unnecessarily will create issues. Hortonworks is working towards Hive supporting multi-geographic data centers in an intelligent way - 'Hive Cross-Geo Query' will allow users to query and report on datasets distributed across geography due to legal or efficiency constraints. Users currently are unable to do this and need to write their own application code that stitches together multiple results.

In the meantime we recommend that customers who run on-premise clusters should replicate all of their data into blob storage within the cloud using *Apache Falcon*. Falcon enables data replication across on-premises and cloud-based storages targets: Microsoft Azure and Amazon S3.

Then if additional computing resources are required (say, for end-of-quarter/year additional workloads) then they can spin up virtual compute instances to work on the data stored purely in the cloud.

xx. Any hardware vendor limitations? Please provide hardware compatibility matrix.

Answer: Below are minimum recommendations for each type of node

Master Nodes – *NameNode, Resource Manager , HBase Master*
Dual Intel Xeon E5-2650v2 (8c) or E5-2660v2 (10c) Processors
128GB or 256GB RAM per chassis
4+ – 1TB NL-SAS/SATA Drives RAID10+ Spares

Worker Nodes – DataNode, Node Manager and Region Server
Dual Intel Xeon E5-2650v2 (8c) or E5-2660v2 (10c) Processors
12 – 1-4 TB NLSAS/SATA Drives
128GB RAM or 256GB RAM

xxi. Please provide a list of any product features that you feel differentiates your product from your competitors.

Answer: Hortonworks delivers 100% Apache Open Source Hadoop Platform for your data lake needs. The Apache Software Foundation governs everything we deliver. We develop code and generate new projects but never hold back proprietary bits or charge for features. We feel this is not how Enterprise Hadoop will gain momentum nor will it benefit our customers. Our belief is that the community will always out develop a single company and that Hadoop is a platform decision that is based on flexibility, compatibility, and enterprise readiness. We will never release early. Five unique things that Hortonworks brings to Hadoop functionality are

1: **YARN:** Apache Hadoop YARN is the data operating system for Hadoop 2. YARN enables a user to interact with all data in multiple ways simultaneously, making Hadoop a true multi-use data platform and allowing it to take its place in a modern data architecture. YARN, which was created and developed by Hortonworks offers Hadoop:

Other distributions will not center around YARN. They will be packaged with YARN but running natively on YARN eliminates the needs for multiple clusters. Every component we have will be certified to run and be managed by YARN. We have seen many customers switch to Hortonworks because of the cluster sprawl from other distributions because each processing paradigm needs a separate cluster.

2: **Enterprise Security**: With the emergence of Hadoop as a business-critical data platform, more stringent requirements for data security are now being required by the enterprise. Hadoop already includes many of these security requirements, but the work is not done.

Hortonworks has already contributed a wide range of security functions, including Kerberos within Apache Hadoop, GRANT/REVOKE commands in Apache Hive and the Apache Knox project for perimeter security among many other features. Recently, we purchased XA Secure to extend this already rich set of features with central administration and coordinated enforcement of security policy across the entire Hadoop ecosystem of projects. And as part of our promise to keep HDP completely open, we will incubate the XA Secure functionality as a project governed by the Apache Software Foundation.

With the open source community, we will continue to pursue three security goals.

Comprehensive Security
Meet all security requirements across authentication, authorization, audit & data protection for all HDP components.

Central Administration
Provide one location for administering security policies and for viewing and managing audit across the platform.

Consistent Integration
Integrate with other security and identity management systems, for compliance with IT policies.

Through individual Apache projects and the acquisition of XA Secure, Hortonworks has already delivered key pieces of the security roadmap in five areas.

Centralized Security Administration: Security best practices should be consistently applied across the platform, and they should be managed centrally with a single user interface. With XA Secure, HDP Advanced Security now provides a security administration console that is unique to HDP but will be

delivered completely in the open for all. Now Hadoop administrators can easily manage all security policies related to access control, in one place.

Authentication: HDP Advanced Security provides support for Kerberos-based authentication. Kerberos can be connected to corporate LDAP environments to centrally provision user information. HDP also provides perimeter authentication through Apache Knox for REST APIs and Web services.

Authorization: Authorization or entitlement is the process of ensuring that users have access only to data as per corporate policies. Hadoop already provides fine-grained authorization via file permissions in HDFS, resource-level access control for YARN and MapReduce, and coarser-grained access control at a service level. HBase provides authorization with ACL on tables and column families, while Accumulo extends this further to cell-level control. Apache Hive provides Grant/Revoke access control on tables.

With the addition of Apache Ranger, Hadoop now includes authorization features that help enterprises securely use varied data with multiple user groups while ensuring proper entitlements. It provides an intuitive way for users to specify entitlements policies for HDFS, HBase, and Hive with a centralized administration interface and extended authorization enforcement. Our goal is to provide a common authorization framework for the HDP platform, providing security administrators with a single administrative console to manage all the authorization policies for HDP components.

Audit: One of the cornerstones for any security system is accountability, or having audit data for auditors to control the system and check for regulatory compliance, for example in Healthcare around HIPAA compliance. Healthcare providers would look within audit data for access history for sensitive data such as patient records, and provide the data if requested by patient or any regulatory authority. Having a robust audit data would help enterprises manage their regulatory compliance needs better as well as control the environment proactively. Ranger provides a centralized framework for collecting access audit history and easy reporting on the data. The data can be filtered based on various parameters. Our goal is to enhance the audit information that is captured within various components within Hadoop and provide insights through the centralized reporting.

Data protection: Data protection involves protecting data at rest and in motion, including encryption and masking. Encryption provides an added layer of security by protecting data when it is transferred and when it is stored (at rest), while masking capabilities enable security administrators to desensitize PII for display or temporary storage. We will continue to leverage the existing capabilities in HDP for encrypting data in flight, while bringing forward partner solutions for encrypting data at rest, data discovery, and data masking.

3. **Stinger.next initiative – Sub Second Queries natively in 100% open source.** Hive is the de facto standard for SQL-in-Hadoop with more enterprises relying on this open source project than any alternative. The Stinger.next initiative is a broad, community-based effort to drive the future of Apache Hive, delivering true enterprise SQL at Hadoop scale.

In April 2014, the final phase of the initial Stinger Initiative was delivered on schedule. This bought interactive SQL query to Apache Hive, advancing Hadoop's SQL capabilities at petabyte scale in pure open source. Over 13 months, 145 developers from 44 companies delivered contributed over 390,000 lines of code to the Hive project alone. This work has already had tremendous impact for the Hadoop ecosystem

Stinger.next is a continuation of this initiative to further speed, scale and SQL in Hive across a familiar and attainable three-phase delivery schedule all in the open Apache Hive community, with these three objectives:

Speed
Deliver sub-second query response times.

Scale
The only SQL interface to Hadoop designed for queries that scale from Gigabytes to Terabytes and Petabytes.

SQL
Enable transactions and SQL:2011 analytics for Hive. Hive has always been the defacto standard for SQL in Hadoop and these advances will surely accelerate the production deployment of Hive across a much wider array of scenarios. Explicitly, some of the key deliverables that will enable these new business applications of Hive include:

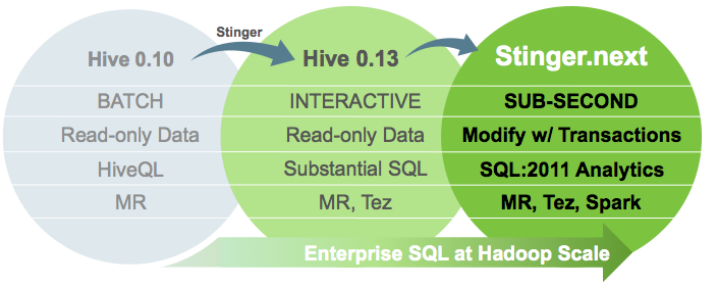
Transactions with ACID semantics allow users to easily modify data with inserts, updates and deletes. They extend Hive from the traditional write-once, and read-often system to support analytics over changing data. This enables reporting with occasional corrections and modifications and allows operational reporting with periodic bulk updates from an operational database.

Sub-second queries will allow users to deploy Hive for interactive dashboards and explorative analytics that have more demanding response-time requirements.

SQL:2011 Analytics allows rich reporting to be deployed on Hive faster, more simply and reliably using standard SQL. A powerful cost based optimizer ensures complex queries and tool-generated queries run fast. Hive now provides the full expressive power that enterprise SQL users have enjoyed, but at Hadoop scale.

In addition to these primary use cases, some additional enhancements include:

- Hive Streaming Ingest helps Hive users expand operational reporting on the latest data.
- Hive Cross-Geo Query allows users to query and report on datasets distributed across geography due to legal or efficiency constraints. Users are currently are unable to do this or write application code to stitch together multiple results.
- Materialized views allow storing multiple views of the same data allowing faster analyses. The views can be held speculatively in-memory and discarded when memory is needed.
- Usability improvements will help users work more simply with Hive.
- Simplified deployment will focus on providing near plug and play deployment solutions for the most common use cases.



4. **Apache Slider** is a new framework that makes it easy to package and deploy a wide array of real-time "always on" services that need to run "IN Hadoop" in a way that does NOT monopolize the cluster and impact other workloads

5: **Real time Complex Event Processing with Storm** YARN opened up Hadoop for data access by applications other than MapReduce. One of the most commonly demanded use cases was the antithesis of batch: stream processing in Hadoop. Apache Storm is a fully certified component of HDP, and our customers are using stream processing for real-time analysis of some of the most common new types of data such as sensor and machine data.

xxii. Please provide a list of products and descriptions of the products that is not a part of the base Hadoop Open Source package.

Answer: Hortonworks is 100% open source and everything is governed by the Apache Software Foundation. We don't include any proprietary components, code, or patches.

Hortonworks Data Platform 2.2	Apache Version
Data Management	
Apache Hadoop and Yarn	2.6
Data Access	
Apache Tez	0.6
Apache Pig	0.14
Apache Hive & H Catalog	0.14
Apache Hbase	.098.4
Apache Phoenix	4.2.0
Apache Accumulo	1.6.1
Apache Slider	0.5.1
Apache Storm	0.9.3
Apache Mahout	0.9
Apache Solr	4.10.0
Apache Spark	1.2
Governance and Integration	
Apache Falcon	0.6.0
Apache Kafka	0.8.1
Apache Sqoop	1.4.5
Apache Flume	1.5

Operations	
Apache Ambari	1.7.0
Apache Oozie	4.1
Apache Zookeeper	3.4.5
Data Security	
Apache Knox	0.5
Apache Ranger	0.4

xxiii. Please describe how your solutions scales based on the following:
   1. Additional user processes
   2. Additional data feeds
   3. Storage capacity

Answer: Our solution scales on a per node basis. Our SKU's are sold in increments of 4 nodes each. Based on the use case, some clusters require more processing heavy nodes and some require storage heavy nodes. Each use case is analyzed by the data ingest patterns and data type, processing requirements, storage capacity, and business output to determine the best configuration of data nodes. Our solution supports heterogeneous data node types, heterogeneous storage within a node and node labeling so specific use cases can be aligned to the correct node configuration.

c. Data Access / Data Protection
   i. Can we set central policies for data retention? If so, please describe in detail, specifically in regards to different data retentions based on data owner and data type.

Answer: Falcon defines data movement and retention policies through *Entity Definitions*. There are three types of entities in Falcon:

- **Cluster**
  Represents interfaces to a Hadoop Cluster and defines colos, clusters, services (Resource Manager, Oozie, HDFS);
- **Feed**
  Defines "dataset" with location, replication schedule, and retention policy;
- **Process**
  Defines the configuration required to run workflow(s).

Falcon allows the user to retain data in the system for a specific period of time for a scheduled **feed**. The user can specify the retention period in the respective feed for each cluster.

   ii. Can we set central policies for managing cluster resources across different workloads? Explain how.

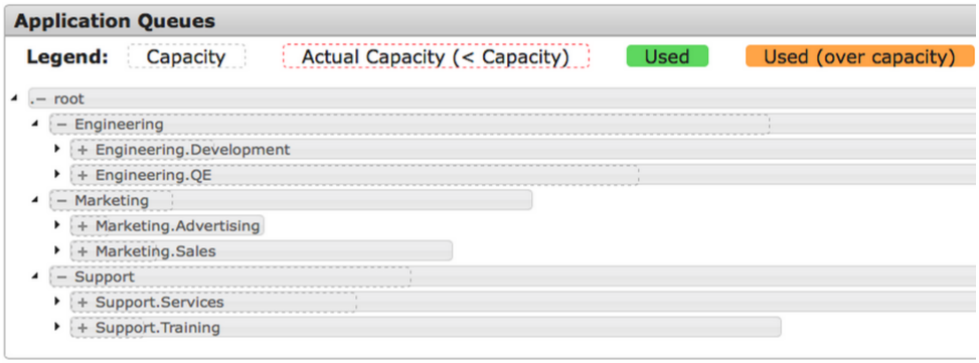
Answer: YARN is the prerequisite for Enterprise Hadoop, providing resource management and a central platform to deliver consistent operations, security, and data governance tools across Hadoop clusters.

YARN's **CapacityScheduler** is designed to run Hadoop applications in a shared, multi-tenant cluster while maximizing the throughput and the utilization of the cluster.

Traditionally each organization has it own private set of compute resources that have sufficient capacity to meet the organization's SLA. This generally leads to poor average utilization. Also there is heavy overhead of managing multiple independent clusters. Sharing clusters between organizations allows economies of scale . However, organizations are concerned about sharing a cluster in the fear of not getting enaough available resources that are critical to meet their SLAs.

The CapacityScheduler is designed to allow sharing a large cluster while giving each organization capacity guarantees. There is an added benefit that an organization can access any excess capacity not being used by others. This provides elasticity for the organizations in a cost-effective manner. Sharing clusters across organizations necessitates strong support for multi-tenancy since each organization must be guaranteed capacity and safe-guards to ensure the shared cluster is impervious to single rogue application or user or sets thereof. The CapacityScheduler provides a stringent set of limits to ensure that a single application or user or queue cannot consume disproportionate amount of resources in the cluster. Also, the CapacityScheduler provides limits on initialized/pending applications from a single user and queue to ensure fairness and stability of the cluster.

The primary abstraction provided by the CapacityScheduler is the concept of queues. These queues are typically setup by administrators to reflect the economics of the shared cluster.



iii. Can we set QOS for individual workloads, users, or groups?

Answer: To provide further control and predictability on sharing of resources, the CapacityScheduler supports hierarchical queues to ensure resources are shared among the sub-queues of an organization before other queues are allowed to use free resources, thereby providing affinity for sharing free resources among applications of a given organization.

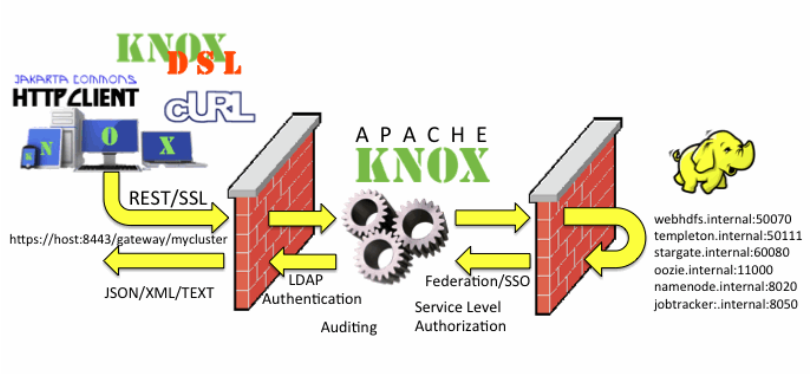
Users are granted access to queues via ACLs, effectively providing QoS.

iv. Can we set central security policies around authentication/authorization? Please describe in detail.

Answer:

Central Authentication policies

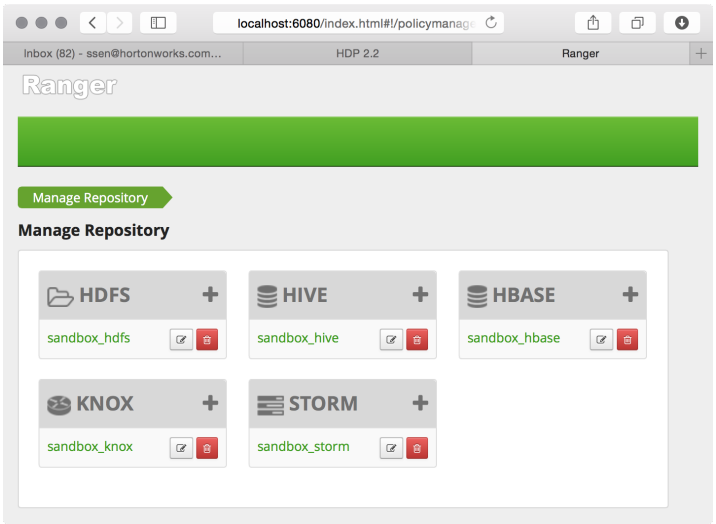
Knox also simplifies Hadoop security for users who access the cluster data and execute jobs. It integrates with prevalent identity management and SSO systems and allows identities from those enterprise systems to be used for seamless, secure **authentication** to Hadoop clusters.



Central Authorization policies

Apache Ranger delivers a comprehensive approach to security for a Hadoop cluster. It provides central security policy administration across the core enterprise security requirements of **authorization**, accounting and data protection.

Apache Ranger offers a centralized security framework to manage fine-grained access control over Hadoop data access components like Apache Hive and Apache HBase. Using the Apache Ranger console, security administrators can easily manage policies for access to files, folders, databases, tables, or column. These policies can be set for individual users or groups and then enforced within Hadoop.



Security administrators can also use Apache Ranger to manage audit tracking and policy analytics for deeper control of the environment. The solution also provides an option to delegate administration of certain data to other group owners, with the aim of securely decentralizing data ownership. Apache Ranger currently supports authorization, auditing and security administration of following HDP components:

- Apache Hadoop HDFS
- Apache Hive
- Apache HBase
- Apache Storm
- Apache Knox

v. Can we define access control for specific data sets within the cluster based on user roles, or groups? Please provide capabilities for file, row and column level access.

Answer: Apache Ranger delivers a comprehensive approach to security for a Hadoop cluster. It provides central security policy administration across the core enterprise security requirements of authorization, accounting and data protection.

What Ranger Does
Apache Ranger offers a centralized security framework to manage fine-grained access control over Hadoop data access components like Apache Hive and Apache HBase. Using the Apache Ranger console, security administrators can easily manage policies for access to files, folders, databases, tables, or column. These policies can be set for individual users or groups and then enforced within Hadoop.

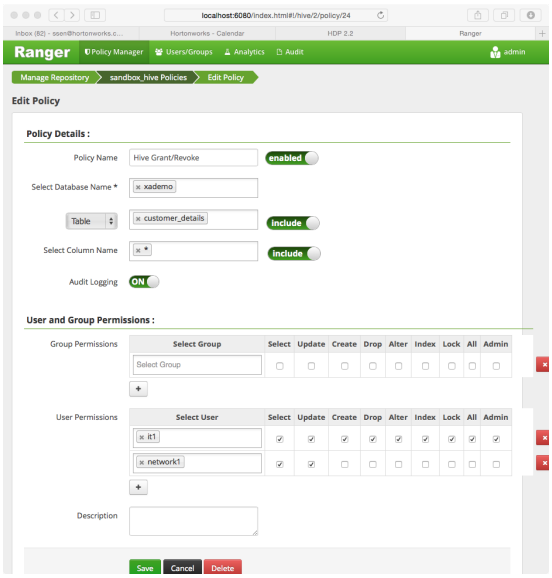
Security administrators can also use Apache Ranger to manage audit tracking and policy analytics for deeper control of the environment. The solution also provides an option to delegate administration of certain data to other group owners, with the aim of securely decentralizing data ownership. Apache Ranger currently supports authorization, auditing and security administration of following HDP components:

- Apache Hadoop HDFS
- Apache Hive
- Apache HBase
- Apache Storm
- Apache Knox

Group sources
Apache Ranger provides a user synchronization utility to pull users and groups from Unix or from LDAP or Active Directory. The user or group information is stored within Ranger portal and used for policy definition.

Fine-grained policies…down to column-level
In your question above, you enquire about 'column-level access". Because Hive (and HBase) policies allow us to define rights down to the column, we would define an HCatalog object on the particular file allowing Hive to treat it as a table, then define a policy around it.

In the following screen shot you can see a Hive policy created for the "x" column within the "customer_details" table within the "xademo" database.



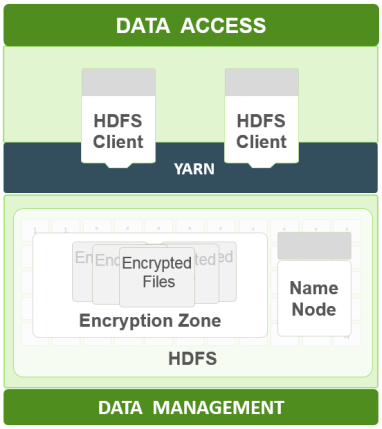
This particular policy has no groups mapped (although it could) but allows the "it1" user full writes the "x" column whilst the "network1" user only has permissions to SELECT and UPDATE it.

vi. Can we define encryption at the file, data element, and disk level? Please describe.

Answer: File-level encryption

Encryption is not just critical for Data Protection within the Hadoop environment, but across the entire ecosystem. That's why Hortonworks was the instrumental force behind introducing a new abstraction to HDFS: "Transparent Encryption in HDFS" that was release in January 2015 in Tech preview with a GA expected date in HDP 2.3, our next release.

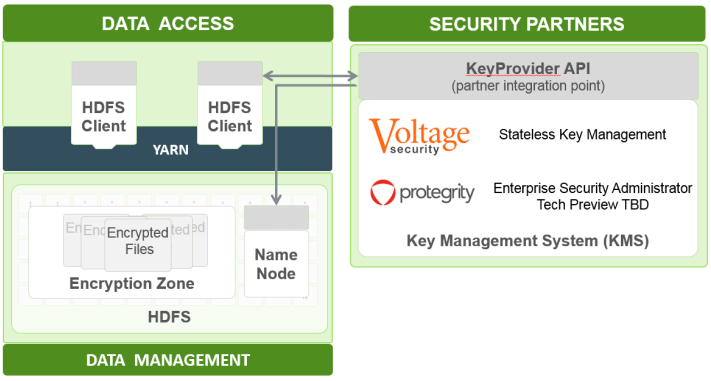
Central to this feature is the *encryption zone* that enables file-by-file encryption. An encryption zone is a special directory whose contents will be transparently encrypted upon write and transparently decrypted upon read. Each encryption zone is associated with a single encryption zone key which is specified when the zone is created. Each file within an encryption zone has its own unique EDEK.



In order to provide encryption of data at rest you need a KMS (Key Management System). Hortonworks provides a fully open-source KMS implementation that enables all files to be encrypted with a master key with Hadoop. However many organizations require comprehensive Key

Management *across all their platforms*, for which we partnered with experienced security technology vendors to deliver enterprise scale Key Management.

- Voltage
  - Partner engineering resources committed to effort
  - Voltage Stateless Key Management integrated with KeyProvider API
- Protegrity
  - Partner currently focused on Avatar™ for Hortonworks



Data Element encryption

Data element encryption (at the column level) can be provided by enabling Voltage's Voltage Format-Preserving Encryption™ (FPE), providing high-strength encryption of data without altering the original data format and preserving business value and referential integrity across distributed data sets. Protection is applied at the **field** or even **partial-field level**, leaving non-sensitive portions of fields available for applications while protecting the sensitive parts.

FPE preserves referential integrity, which means protected data can still be consistently referenced and joined across tables and data sets – a major requirement for proper operations with the mix of data entered into Hadoop, and especially critical where common identifiers like Social Security Numbers or ID's are used as common references across disparate data sets. Policy controlled secure reversibility enables data to be selectively re-identified in trusted systems which need live data, enabling full end-to-end data processes to be secured without resorting to risky and cumbersome mapping databases. Format-Preserving Encryption has also received strong attention from the government, and is recognized as a mode of standard AES encryption, specifically FF1 mode AES defined in NIST 800-38G. This provides users confidence in the security proofs and standards underpinning Voltage FPE. Voltage has built a robust eco-system around FPE, providing support across multiple enterprise platforms with proven implementation tools, delivering 'always-on' data protection.

Disk-level encryption

For disk-level encryption, LUKS (Linux Unified Key Setup) is the standard. LUKS specifies a platform-independent standard on-disk format for use in various tools. This not only facilitates compatibility and interoperability amongst different programs, but also assures that they all implement password management in a secure and documented manner.

Solution	Type	Description

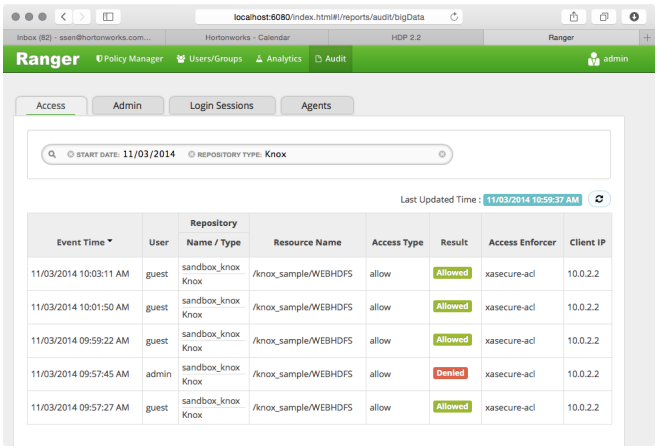
LUKS	File System/Volume	• Linux Unified Key Setup-on-disk-format (or LUKS) allows you to encrypt partitions on your Linux; volume encryption technology • **Advantage:** Reduced complexity since encryption is seamlessly done by OS.
HDFS Transparent Encryptions	HDFS	Open Source encryption that any HDP engine can leverage
Voltage Security	HDFS	• Encryption vendor/partner provides seamless encryption of data at MR level via Pig and Hive UDFS • **Advantage:** Format Preserving Encryption
Protegrity	HDFS	• Provides encryption for full stack • **Advantage:** Key management, centralized authorization policies

vii. Does your solution have any way of providing audit capabilities for data access at file, row and data element? Please describe audit capabilities.

Answer: Apache Ranger can be used to manage audit tracking and policy analytics for deeper control of the environment. The solution also provides an option to delegate administration of certain data to other group owners, with the aim of securely decentralizing data ownership.
Apache Ranger currently supports authorization, auditing and security administration of following HDP components:

- Apache Hadoop HDFS
- Apache Hive
- Apache HBase
- Apache Storm
- Apache Knox

Below is a snapshot of how audited events are reviewed in HDP:



You can see from the above screenshot the user/time is recorded along with the file (for HDFS objects) or the column/table (for Hive objects).

viii. How can we restrict access to data node directly?

Answer: Data nodes should be housed in a secure data center that is only accessible by authorized users. Remote access to data nodes (SSH) should be limited to only a subset of Linux/Hadoop admins and controlled by a firewall. Their authorization and access should be audited too.

ix. Does your solution support the auditing of super user access? Please describe

Answer: Yes super user access is audited through a variety of ways. The Hadoop User Experience (HUE) super user access is audited and managed through Apache Ranger's auditing capabilities. For Ambari, any changes made by any user including super users is versioned and tracked. Operating system logs provide auditing for when Sudo users sudo to a super user account (e.g. hdfs).

x. Does your solution support auditing of operational activities (upgrades, bounces, user creation, file creation, deletion, etc.)? Please describe.

Answer: Apache Ambari provides auditing of all Hadoop Configuration changes. It has version control capabilities as well as the ability to rollback to previous versions if needed.

xi. Does your system provide an upgrade of software without upgrading the whole distribution?

Answer: Hortonworks typically releases two system updates a year. These updates normally include the latest stable, GA versions of each hadoop component. Furthermore, Hortonworks will push out additional updates (e.g. Ambari) when a new stable, GA release of a component is available that are decoupled from the Hortonworks Data Platform. Hortonworks releases 4-5 minor releases also that encompasses any addition upgrades that are recommended.

xii. Does your solution provide auto rollback for upgrades?

Answer: HDP 2.2 supports side-by-side installation of HDP 2.2 and above releases, which lets you perform rolling upgrades on your cluster and enables you to run multiple versions concurrently. To support side-by-side installation, the HDP package version naming convention for both RPMs and Debs has changed to include the HDP 2.2 product version. For example, hadoop-hdfs is now hadoop-2.2.0.1-hdfs. HDP 2.2 marks the first release where HDP rpms, debs, and directories contain versions in the names to permit side-by-side installations of later HDP releases.

In the event of an unsuccessful upgrade, Ambari provides the ability to rollback upgrades.

xiii. How can we protect our data in your solution? Please describe options available for data backup/protection as well as data replication.

Answer: Data replication in a single cluster is built into the data redundancy philosophies that Hadoop was built on. Hadoop provides 3x (which is configurable) replication of data stored on the cluster. In the event that a data node goes down, there are still two copies of the data available. Moreover, Hadoop will automatically create a 3rd back up again.

Apache Falcon simplifies the development and management of data processing pipelines with a higher layer of abstraction, taking the complex coding out of data processing applications by providing out-of-the-box data management services. This simplifies the configuration and orchestration of data motion, disaster recovery and data retention workflows. Apache Falcon provides replication across on-premises and cloud-based storages targets: Microsoft Azure and Amazon S3

xiv. Can the data protection and replication options be configured to run automatically, if so please describe how?

Answer: HDFS block level replication is a configurable parameter in the Ambari UI. Data replication is defined within Falcon, but orchestrated using Oozie. Replication jobs therefore run automatically.

xv. Can we restrict few access interfaces for few users? Like restricting command hive shell… for few users.

Answer: It is suggested that all users interact with the Hadoop cluster via a UI (Ambari, HUE, etc). Access to the command line interfaces should be restricted to only those users with privileges to remotely access particular services.

Many services are exposed via RESTful APIs provided by Knox, a reverse proxy. Access to any of these services can be restricted via ACLs.

xvi. Does your solution have portal for data access? This portal can be used as self-service portal by business users for data access without IT involvement?

Answer: HDP 2.2 supports HUE (Hadoop User Experience) a Web based UI that allows users to view, retrieve and upload files into HDFS. Ambari although primarily our administration console currently, provides 'Ambari Views' that can expose much of the functionality provided by Hue. Our direction will be to shift users from Hue to Ambari's Views as soon as the functionality becomes available.

xvii. What is an interface available for interactive SQL (Non M/R, Considering M/R has lazy start problem)?

Answer: HDP 2.2 provides interactive SQL capabilities through Hive on Tez. Upon the successful completion of the Stinger initiative, Hive has seen 100x performance. These vast performance improvements have allowed Hive to be used for interactive SQL. Furthermore, Hive is ANSI SQL:2011 compliant and it integrates via ODBC/JDBC with BI tools including Tableau, SAP BO, MS SSRS, Qlikview, Platfora, and Microstrategy.

xviii. Does your solution provide SQL client (other than command shell) for executing ad-hoc SQL queries, if needed?

Answer: Yes, HDP 2.2 comes with HUE (Hadoop User Experience) that provides a web UI for executing Hive SQL queries.

xix. Is your solution designed with high availability (HA)? If so, please describe the HA features provided in your solution. Describe any single points of failure in your system.

Answer: HDP provides High Availability for the following services:

NameNode (HDFS) High Availability:

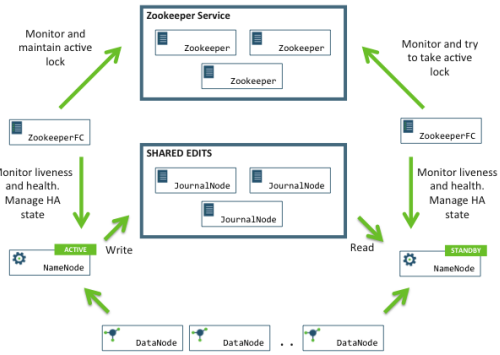
Both Active and Standby NameNodes have up to date HDFS metadata, ensuring seamless failover even for large clusters – which means no downtime for your HDP cluster!
Full Stack Resiliency: The entire HDP stack (MapReduce, Hive, Pig, HBase, Oozie) has been certified to handle a NameNode failure scenario without losing data or the job progress. This is vital to ensure long running jobs that are critical to complete on schedule will not be adversely affected during a NameNode failure scenario.
With HDP, this entire functionality is built into the HDP product. This means:

- No need for shared storage
- No need for external HA frameworks
- Certified with a secure Kerberos configuration

Active Name Node

The Active NameNode is responsible for all client operations in the cluster. The Standby NameNode maintains enough state to provide a fast failover. In order for the Standby node to keep its state synchronized with the Active node, both nodes communicate through a group of separate daemons called JournalNodes. The file system journal logged by the Active NameNode at the JournalNodes is consumed by the Standby NameNode to keep it's file system namespace in sync with the Active.

In order to provide a fast failover, it is also necessary that the Standby node have up-to-date information of the location of blocks in your cluster. DataNodes are configured with the location of both the NameNodes and send block location information and heartbeats to both NameNode machines.



The ZooKeeper Failover Controller (ZKFC) is responsible for HA Monitoring of the NameNode service and for automatic failover when the Active NameNode is unavailable. There are two ZKFC processes – one on each NameNode machine. ZKFC uses the Zookeeper Service for coordination in determining which is the Active NameNode and in determining when to failover to the Standby NameNode.

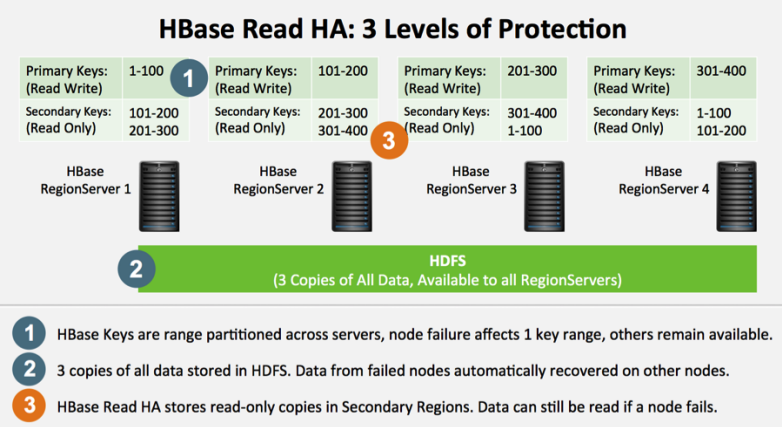
Quorum journal manager (QJM) in the NameNode writes file system journal logs to the journal nodes. A journal log is considered successfully written only when it is written to majority of the journal nodes. Only one of the Namenodes can achieve this quorum write. In the event of split-brain scenario this ensure that the file system metadata will not be corrupted by two active NameNodes.

In HA setup, HDFS clients are configured with a logical name service URI and the two NameNodes corresponding to it. The clients perform source side failover. When a client cannot connect to a NameNode or if the NameNode is in standby mode, it performs fail over to the other NameNode.

HBASE High Availability

With the Hortonworks Data Platform 2.2, HBase High Availability has taken a major step forward, allowing apps on HBase to deliver 99.99% uptime guarantees.

Hortonworks worked within the HBase community to deliver this major step forward in HA by introducing Timeline-Consistent Region Replicas, also called “HBase Read HA” and tracked by HBASE-10070. At a high level this new HA feature maintains multiple copies of the same data in Primary Region Replicas and Secondary Region Replicas that are spread throughout the HBase cluster. With HBase Read HA, if a RegionServer fails, its data can still be read from a separate RegionServer. In other words you retain read availability while only losing write availability during automatic recovery. This makes HBase Read HA a great option for read-heavy workloads that need consistency and no downtime. Combined with best practices such as using 2 replicas and rack awareness, HBase Read HA allows HBase to deliver 99.99% availability for these mission-critical applications.



Yarn ResourceManager High Availability

The YARN High Availability (HA) feature added in HDP 2.2 provides redundancy in the form of an Active/Standby RM pair to remove this single point of failure. Furthermore, upon failover from the Standby RM to the Active, the applications can resume from their last check-pointed state; for example, completed map tasks in a MapReduce job are not re-run on a subsequent attempt. This allows events such the following to be handled without any significant performance effect on running applications.:

- Unplanned events such as machine crashes
- Planned maintenance events such as software or hardware upgrades on the machine running the ResourceManager.

Hive High Availability

Hive's Metastore Service can be deployed in HA mode, each Hive metastore client will read the configuration property *hive.metastore.uris* to get a list of metastore servers with which it can try to communicate.. The HiveServer2 service can already be deployed manually in HA mode and will be available for easier-deployment within Ambari 2.0 onwards. Since Hive's HCatalog persists its configuration to a relational database, that too must be made highly available.

Storm High Availability

Storm Nimbus service is not a single point of failure in the strictest sense (i.e. loss of the Nimbus node will not affect running topologies). However, the loss of the Nimbus node does degrade functionality for deploying new topologies and reassigning work across a cluster.

Upcoming releases of HDP will eliminate this "soft" point of failure by supporting an HA Nimbus. Multiple instances of the Nimbus service run in a cluster and perform leader election when a Nimbus node fails.

Solr High Availability

Extra shard copies can be used for high availability and fault tolerance, or simply for increasing the query capacity of the cluster. SolrCloud can continue to serve results without interruption as long as at least one server hosts every shard.

Zookeeper High Availability

Running multiple zookeeper servers in concert (a zookeeper ensemble) allows for high availability of the zookeeper service. Every zookeeper server needs to know about every other zookeeper server in the ensemble, and a majority of servers are needed to provide service. For example, a zookeeper ensemble of three servers allows any one to fail with the remaining two constituting a majority to

continue providing service. Five zookeeper servers are needed to allow for the failure of up to two servers at a time.

Oozie High Availability
High Availability (HA) for Oozie has been available since version 4.0.1 (HDP 2.2 ships version 4.1.0). Since Oozie persists its configuration and job's progress state to a relational database, that too must be made highly available.

Knox High Availability
Knox is a stateless Reverse Proxy, so multiple instances can be deployed behind a highly-available web server for failover and scalability.

xx. Is HA implemented automatically, or is there manual intervention needed to perform failover of components?

Answer: All of the services listed in the previous question provide automatic failover to standby services.

xxi. Does your solution allow customers to implement apache open source components? In doing so, does it change existing support agreements?

Answer: Customers are free to install additional components on their development and test clusters. Hortonworks can issue no guaranteed support for these components except for "best effort". If a customer can demonstrate that an issue issues on a reference platform then Hortonworks will provide support.

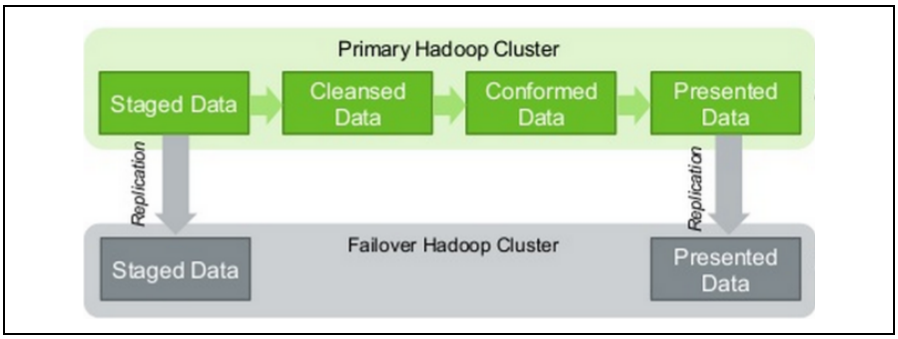
xxii. Are there options to implement geo-redundant architectures for you solution? If so, please describe in detail including any additional costs associated with a geo-redundant solution.

Answer: It is quite common to implement a geographically redundant cluster where data is replicated between two geographically-separated sites so that applications can switch from one site to another (for example, in case of the catastrophic failure of the primary site) and still have all the configuration data (account information, system SLAs, and budgets) necessary for SLA compliancy available at the second, remote site.

Data replication is handled by Apache Falcon within HDP from one cluster to another to provide multi-cluster failover. In this case Falcon manages the flow of taking staged data and cleansing, conforming and presenting that data in the primary cluster. Falcon does this by leveraging what are called Entity Definitions. There are three types of entities in Falcon:

Cluster	Represents interfaces to a Hadoop Cluster Define colos, clusters, services (Resource Manager, Oozie, HDFS)
Feed	Defines "dataset" with location, replication schedule, and retention policy
Process	Defines the configuration required to run workflow(s)

In this case you would define two (or more) **clusters**, at two **feeds** (one representing the source path within the primary cluster, and a second representing the target path within the DR cluster) and at least one **Process** that would copy the data over on a scheduled basis.



xxiii. Does your solution provide any Sql on Hadoop? Please describe the main components on the Sql on Hadoop solution.

Answer: Since its incubation in 2008, Apache Hive is considered the defacto standard for interactive SQL queries over petabytes of data in Hadoop. And with the completion of the Stinger Initiative, and the first phase of Stinger.next, the Apache community has greatly improved Hive's speed, scale and SQL semantics. Throughout all the innovation, Hive easily integrates with other critical data center technologies using a familiar JDBC interface.

Stinger.next: Hortonworks Focus for Apache Hive
The Stinger Initiative successfully delivered a fundamental new Apache Hive, which evolved Hive's traditional architecture and made it faster, with richer SQL semantics and petabyte scalability. We continue to work within the community to advance these three key facets of hive:

- **Speed**: Deliver sub-second query response times
- **Scale**: The only SQL interface to Hadoop designed for queries that scale from Gigabytes, to Terabytes and Petabytes
- **SQL**:Enable transactions and SQL:2011 Analytics for Hive

Stinger.next is focused on the vision of delivering enterprise SQL at Hadoop scale, accelerating the production deployment of Hive for interactive analytics, reporting and and ETL. More explicitly, some of the key areas that we will invest in include:

Focus	Planned Enhancements
Speed	• LLAP, a process for multi-threaded execution, will work with Apache Tez to achieve sub-second response times • Sub-second queries will support interactive dashboards and explorative analytics • Materialized views will allow multiple views of the same data and speed analysis
Scale	Cross-geo query will allow users to query and report on geographically distributed datasets
SQL Semantics	• Transactions with ACID semantics will allow users to easily modify data with inserts, updates and deletes • SQL:2011 Analytics will allow rapid deployment of rich Hive reporting • A powerful cost-based optimizer will ensure that complex queries run quickly
Spark Machine Learning Integration	Allow users to build and run machine learning models via Hive, using Spark Machine learning libraries
Streaming Ingest	Help users expand operational reporting on the latest data by replicating from operational databases

Goal	Description
Transactions with ACID semantics	Delivered in HDP 2.2, ACID transactions will allow users to easily modify data with inserts, updates and deletes. They extend Hive from the traditional write-once, and read-often system to support analytics over changing data. This enables reporting with occasional corrections and modifications and allows operational reporting with periodic bulk updates from an operational database.
Sub-second queries	Will allow users to deploy Hive for interactive dashboards and explorative analytics that have more demanding response-time requirements
SQL:2011 Analytics	Will allow rich reporting to be deployed on Hive faster, more simply and reliably using standard SQL. A powerful cost based optimizer ensures complex queries and tool-generated queries run fast. Hive now provides the full expressive power that enterprise SQL users have enjoyed, but at Hadoop scale.

xxiv. Give direct comparisons of interactive SQL performance –

Answer: We have performed extensive testing of queries against "Hive on Tez using ORC format", "Hive on Spark using Parquet format and "Spark-sql using Parquet format"; for a summary see here, for the details see this presentation on Slideshare (where the presentation can also be downloaded).

xxv. Specifically compare HIVE/TEZ/Presto/Impala on similar ad hoc and interactive queries from COGNOS and Tableau?

Answer: We have performed extensive testing of queries against "Hive on Tez using ORC format", "Hive on Spark using Parquet format and "Spark-sql using Parquet format"; for a summary see here, for the details see this presentation on Slideshare (where the presentation can also be downloaded).

In addition, since moving to Hive 13 in HDP 2.1 and now Hive 14 in HDP 2.2 , there have been additional significant improvements that both Cognos and Tableau will take advantage of. Our goal in Stinger.next is to provide sub second query for TB of data natively in Hive utilizing LLAP, ORC file format, Tez, and Hive improvements. The following table represents the improvements between Hive 13 and Hive 14 on **30TB** samples of TPC-DS queries. Hortonworks is the only distribution currently on Hive 14 and planning on going to Hive 15 in our HDP 2.3 release.

TPC- DS Query	Hive 14 (seconds)	Hive 13 (seconds)	Improvement Factor
3	39	226	579%
7	86	263	306%
12	41	163	398%
13	1,136	8,528	751%
17	235	2,957	1258%
19	90	215	239%
20	43	267	621%
21	34	46	135%
25	229	3,187	1392%
26	67	178	266%
27	78	252	323%
28	710	2,227	314%
29	802	2,902	362%
31	368	1,020	277%
32	78	351	450%
34	100	465	465%

39	63	118	187%
40	214	2,462	1150%
42	35	148	423%
43	326	512	157%
45	52	1,094	2104%
46	169	407	241%
49	112	2,304	2057%
50	1,097	3,125	285%
52	38	149	392%
54	99	1,525	1540$
55	34	164	482%
56	68	407	599%
58	86	1,904	2214%
60	105	403	384%
66	138	1,331	964%
68	83	327	394%
71	76	374	492%
73	48	391	815%
75	2,004	3,511	175%
76	249	452	182%
79	262	535	204%
82	1,256	2,223	177%
84	70	494	706%
85	448	2,252	503%
87	1,012	1,672	165%
88	1,031	1,767	171%
89	200	273	137%
90	97	131	135%
91	49	100	204%
92	759	1,028	135%
96	140	201	144%
97	895	1,258	141%
98	61	1,085	1779%

xxvi. Does your solution support any Mapreduce alternatives such as Tez and Spark? Please describe how these fit in your ecosystem and the capabilities they provide to your solution.

Answer:

Tez

Tez improves the MapReduce paradigm by dramatically improving its speed, while maintaining MapReduce's ability to scale to petabytes of data. Important Hadoop ecosystem projects like Apache Hive and Apache Pig use Apache Tez, as do a growing number of third party data access applications developed for the broader Hadoop ecosystem.

Spark

Hortonworks has outlined a set of initiatives to work on some of the current challenges with Spark that will make it easier for users to consume as an enterprise-ready part of the completely open source Hortonworks Data Platform (HDP). While delivery is planned into discrete phases laid out below, the work can be categorized into two distinct categories:

- **YARN-enabled Spark**

Deeper integration of Spark with YARN will allow it to become a more efficient tenant along side other engines, such as Hive, Storm and HBase and others, simultaneously, all on a single data platform. This avoids the need to create and manage dedicated Spark clusters to support that subset of applications for which Spark is ideally suited and more effectively share resources within a single cluster.

xxvii. How is DR handled in your solution? What is typical Recovery Point Objective/ Recovery Time Objective (RPO/RTO)?

Answer: Disaster Recovery (DR) is provided by replicating data using Apache Falcon. Apache Falcon is a framework for simplifying and orchestrating data management and pipeline processing in. It is fully customizable and has the ability to specific different point in time data replication frequencies for different data sets

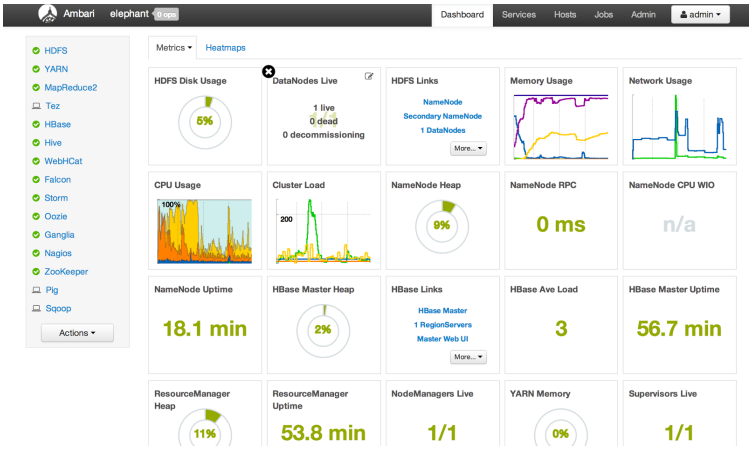
xxviii. If solution provide geo-redundant architecture then, does the solution have automatic failover / load balance features?

Answer: Geo-redundant architecture is managed through Apache Falcon, which can be used to provide point in time back-up/synching up of data from one hadoop cluster to another hadoop cluster. The switchover would have to be manually initiated in the event of a catastrophic failure of the primary site.

d. Operations / Management

i. Does your solution provide a web-based management utility/portal for monitoring the cluster?

Answer: HDP provides *Apache Ambari,* a completely open operational framework for provisioning, managing, and monitoring Hadoop clusters.



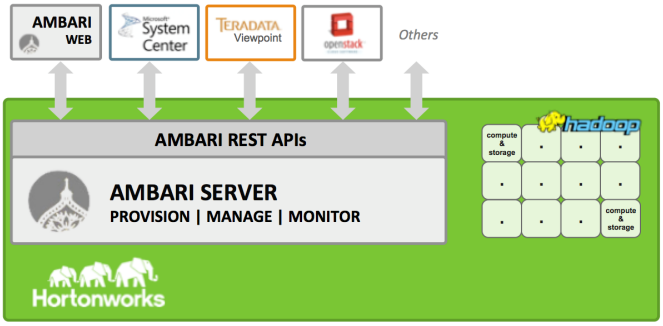
Ambari includes an intuitive collection of operator tools and a set of APIs that mask the complexity of Hadoop, simplifying the operation of clusters.

- **Provision a Hadoop Cluster** No matter the size of your Hadoop cluster, the deployment and maintenance of hosts is simplified using Ambari. Ambari includes an intuitive Web interface

that allows you to easily provision, configure and test all the Hadoop services and core components. Ambari also provides the powerful Ambari Blueprints API for automating cluster installations without user intervention.

- **Manage a Hadoop cluster** Ambari provides tools to simplify cluster management. The Web interface allows you to control the lifecycle of Hadoop services and components, modify configurations and manage the ongoing growth of your cluster.
- **Monitor a Hadoop cluster** Gain instant insight into the health of your cluster. Ambari pre-configures alerts for watching Hadoop services and visualizes cluster operational data in a simple Web interface.
- **Integrate Hadoop with the Enterprise.** Ambari provides a RESTful API that enables integration with existing tools, such as Microsoft System Center and Teradata Viewpoint, to merge Hadoop with your established operational processes.

Apache Ambari is a 100% open source *framework* for provisioning, managing and monitoring Hadoop clusters



Hadoop cluster provisioning and ongoing management can be a complicated task, especially when there are hundreds or thousands of hosts involved. Ambari provides a single control point for viewing, updating and managing Hadoop service life cycles, with these important features:

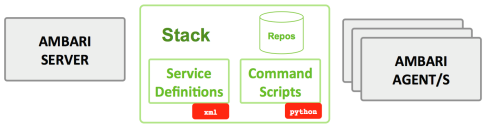
- **Wizard-driven interface**
  Facilitates installation of Hadoop across any number of hosts
- **API-driven installations**
  Ambari Blueprints for automated provisioning
- **Granular control**
  Precise management of Hadoop services and component lifecycles
- **Configuration histories**
  Ongoing management of Hadoop service configurations
- **Extensible framework**
  Brings custom services under management via Ambari Stacks
- **Usability improvements**
  Innovative user experiences via Ambari Views

RESTful APIs
Enables integration with enterprise systems

Ambari Stacks Ambari provides Stacks functionality which allows you to define and add your own custom services to be managed by Ambari (MySQL, MongoDB etc.):

- **Each Service has a definition**
  What Components are part of the Service
- **Each Service has defined lifecycle commands**
  start, stop, status, install, configure

- **Lifecycle is controlled via command scripts**
- **Ability to define "custom" commands**



Ambari Blueprints

Ambari provides Blueprint functionality to provision clusters from scratch based on a "blueprint" produced from another working cluster.

Ambari Views

Ambari also provides Views to provide custom, pluggable views on top of Ambari for administrators and knowledge workers. Views servers can be created to provide this functionality in a protected environment.

ii. Describe the density per TB (Rack Unit's used and expansion increments)

Answer: Hortonworks has recommended configurations for TB per Node.

Node Options

Performance	
Processor	E5-2660v2 (10c)
Disk	12 * 1TB
Memory	256GB
Network	2x 10GbE

Balanced	
Processor	E5-2650v2 (8c)
Disk	12 * 2TB
Memory	128GB
Network	2x 10GbE

Storage	
Model	E5-2650v2 (8c)
Disk	12 * 4TB
Memory	128GB
Network	2x 10GbE

iii. What tools are available with your solution for application development? Example, like sql developer, etc.

Answer: Hortonworks has partnered with multiple 3rd party application development software tools including Informatica and Talend which allow clients to continue using their standard enterprise tools for rapidly developing Hadoop applications. In addition to 3rd party application development tools, HDP 2.2 includes support for HUE (Hadoop User Experience) which is a web based UI for developing Hive and Pig applications.

iv. What tools are available as part of your solution to ease/ mitigate transition of traditional SQL developers to you solution?

Answer: Hortonwork's mission is to enable our customers to continue to leverage the tools and technologies they already know. Apache Hive was created by the community to provide SQL on Hadoop. Since it's inception, Apache hive has become the defacto SQL on Hadoop engine due to it's robust features. As of HDP 2.2, Hive is ANSI SQL:2011 compliant providing just about every key function a SQL developer would need to develop in Hive as they would to access data in any standard RDBMS

v. Does your solution provide support for integration in MS SqlServer? Please describe if the integration is a part of the open source or provided by Microsoft? Please describe the capabilities of the drivers. Please describe the requirements and limitations of the drivers.

Answer: We are the only hadoop distribution that can be deployed in Windows and we have a strong engineering partnership with Microsoft. There are few ways to integrate with SQL server. We provide Hive JDBC/ODBC drivers (available at no cost from Hortonworks) that can be leveraged to integrate with MS SQL server. The HDP platform also provides *Apache Sqoop with* which you can download the free JDBC MSSQL connector to SQL Server.. These Sqoop connectors could be used to ingest data into and out of relational databases like MSSQL sever.

Also Microsoft has a technolgy called polybase that helps users perform seemlessly run T-SQL queries between their Parallel datawarehouse , MS Analytics platform system and HDP. On top of this, Microsoft System center, that provides monitoring for MS components, integrates with HDP Ambari and provides users a single GUI interface to monitor both the MS components and the HDP cluster.

vi. Does your solution provide support for integration in Oracle? Please describe if the integration is a part of the open source or provided by Oracle? Please describe the capabilities of the drivers. Please describe the requirements and limitations of the drivers.

Answer: Refer to the previous answer (Connectivity to MS SQL Server) - the only difference would be to download the free JDBC driver from Oracle. Oracle offers oracle big data connector that can be used to integrate with HDP. Oracle supports that connector and Hortonworks has certified it to work with HDP 2.1.

vii. Please describe how your solution works with a 3rd Party ETL (Extract/Transform/Load) tool, Informatica or Talend.

Answer: Hortonworks has multiple partners that provide UI based ETL tools. Some partners, including Informatica and Talend, are YARN certification partners which means that their applications can sit on top of an HDP cluster. They will leverage the processing and redundant storage capabilities of Hadoop to store and process the data. Below is some additional information regarding our Partnerships with Informatica and Talend:

Talend
Talend Open Studio for Big Data is a powerful and versatile open source data integration tool. Talend provides data managers, operators, and analysts a graphical tool that abstracts the underlying Hadoop complexities and dramatically improves the efficiency of job design through an easy-to-use Eclipse development environment.

Informatica and Hortonworks Data Platform:

- Moves data into and out of Hadoop in batch or real time using universal connectivity to all types and latencies of data.
- Streamlines data ingestion, parsing, profiling, ETL, cleansing, and matching without the complexity of manual hand-coding on Hadoop while leveraging Hadoop's power of distributed processing.
- Ensures timely data delivery between Hadoop and the rest of the enterprise, including streaming, messaging, changed data capture, data virtualization, and batch methods; supports large data distribution with high throughput and concurrency
- Aligns business and IT through a common taxonomy of business and technical metadata definitions, and reduces risk and compliance exposure through data monitoring and audits.

viii. What options does your solution have for alerting on failed software, hardware components, and jobs/applications?

Answer: Apache Ambari is the Hadoop operations, maintenance and monitoring tool. It provides a dashboard overview of the health of the entire cluster. Ambari also provides notifications (in the tool as well as email notifications) of any distressed nodes. Furthermore, Ambari provides links to detailed job and application status to help debug any failed job/application.

ix. Do you have experience integrating with monitoring tools such as CA Nimsoft, IBM Tivoli, or HP Openview, etc.? Please describe in detail.

Answer: Nagios is an open source network monitoring system designed to monitor all aspects of your Hadoop cluster (such as hosts, services, and so forth) over the network. It can monitor many facets of your installation, ranging from operating system attributes like CPU and memory usage to the status of applications, files, and more. Nagios provides a flexible, customizable framework for collecting data on the state of your Hadoop cluster.

Nagios is primarily used for the following kinds of tasks:

- Getting instant information about your organization's Hadoop infrastructure
- Detecting and repairing problems, and mitigating future issues, before they affect end-users and customers
- Leveraging Nagios' event monitoring capabilities to receive alerts for potential problem areas
- Analyzing specific trends; for example: what is the CPU usage for a particular Hadoop service weekdays between 2 p.m. and 5 p.m

Nagios is fully extendable via a simple plugin design that allows users to easily develop their own service checks depending on needs, by using their tools of choice (shell scripts, C++, Perl, Ruby, Python, PHP, C#, etc.). Scripts such as this can be written to attach Nagios to an HP-Openview instance through the notification functions of Nagios.

x. Does you solution provide alerts/traps via snmp?

Answer: Nagios provides management of SNMP traps - including the ability to read, process, and generate alerts from SNMP traps it receives. Nagios can also send SNMP traps to other management hosts, which allows seamless integration with other Network Management Systems.

Implementing effective SNMP Trap management with Nagios offers the following benefits:

- Agentless monitoring
- Increased server, services, and application availability
- Fast detection of network outages and protocol failures

These Nagios solutions provide SNMP Trap management capabilities and benefits:

- Nagios XI
- Nagios Core

Furthermore our Professional Services Organization has implemented Nagios alerts convertor to SNMP Traps and sent to a SNMP server. Contact Kris Kane within our organization for more details.

xi. Do you provide snmp mibs (Management information bases) as part of your solution?

Answer: Ganglia and Nagios are used to monitoring a metric collection over the cluster. Ganglia monitors typical metrics like network, disk io, disk storage, cpu usage, jvm heap, jvm garbage collection, memory cache and swap, etc.

Specifically Hortonworks provides a **plugin for Ganglia** that allows it to monitor specific Hadoop metrics like **RPC traffic**.

xii. What are the options available to manage high concurrency of the system to suffice bulk of request concurrently?

Answer: YARN is the element that enables the modern data architecture as it turns Hadoop into a truly multi-purpose, multi-tenant data platform. It becomes the architectural center of Hadoop.

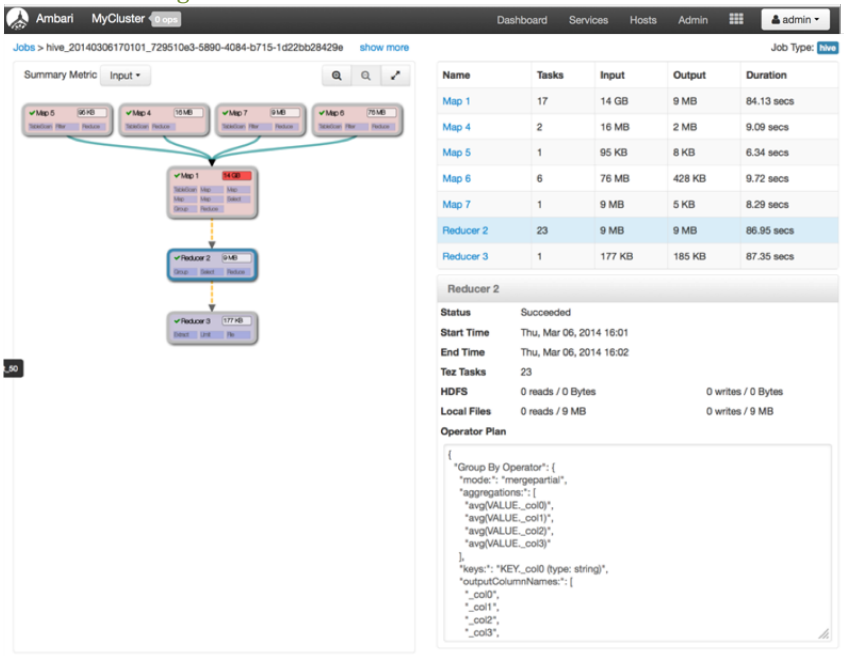
The Capacity Scheduler (CS) ensures that groups of users and applications will get a guaranteed share of the cluster, while maximizing overall utilization of the cluster. Through elastic resource allocation, if the cluster has available resources then users and applications can take up more of the cluster than they're guaranteed minimum share. The CS can then be rebalanced in real time based on other submitted jobs using Preemption to 'preempt' other tasks running to free up a users guaranteed capacity should another user have grown into it with elastic resource allocation.

By creating queues in YARN via the Ambari console, we can set limits on exactly how much cluster resources can be used by each group or individual user. Users will submit jobs and be prioritized accordingly and given only the resources they are allowed. These limits can be refreshed without restarting any services. A more detailed explaination can be found here.

Users can still view other jobs running in the queue, but not influence them unless they have the admin rights to do so. Limiting file viewing is accomplished by setting POSIX permissions on HDFS allowing each file to have rights for its owner, group and other. Using these POSIC permissions we can prevent anyone but the owner from viewing their datasets. Additionally this can be accomplished by the HDP Advanced Security functionality mentioned above.

xiii. What is the interface to see the break points (code where one job is failed) in active production failed jobs?

Answer: In addition to MR's traditional "MR Job-History server", we introduced new functionality called the "YARN Timeline Server" back in HDP 2.1. It is exposed through Ambari, where we have the Hive-on-Tez jobs' views that are built by pulling the events and metrics and visualizing it. As such, you can get a DAG's view of the Tez's runtime engine. You can see metrics for each DAG job and visualize the performance of each job. Over time, we plan to add more visualization by harnessing the power of common monitoring framework for HDP.



xiv. What tools will ensure the code backward compatibility after upgrade?

Answer: HDP 2.2 supports side-by-side installation of HDP 2.2 and above releases, which lets you perform rolling upgrades on your cluster and enables you to run multiple versions concurrently. To support side-by-side installation, the HDP package version naming convention for both RPMs and Debs has changed to include the HDP 2.2 product version. For example, hadoop-hdfs is now hadoop-2.2.0.1-hdfs. HDP 2.2 marks the first release where HDP rpms, debs, and directories contain versions in the names to permit side-by-side installations of later HDP releases.

To select from the releases you have installed side-by-side, Hortonworks provides hdp-select, a script that symlinks directories to hdp-current and modifies paths for configuration directories that lets you select the active version of HDP from the versions you have selected.

xv. Does your solution provide code version control out of box?

Answer: HDP does not provide code version control out of the box but we would use SVN and correlate the SCRUM cycle(release #) with the DB Object (ie. table, stored procedure, bteq script), etc. and manage component compatibility by a matrix that laid out every component and the release versions with a YES/NO; for compatibility to ensure all objects were being tested together and all would play together nicely for each release/domain.

As soon as every object has the YES next to it for a given release, all objects (including DDLs & re-persist scripts) are checked in to the SVN (release folder) and used for the QA rollout. (The QA rollout

was to ensure that there weren't objects that needing persisted that were blown away like a 'drop/create vs. of an alter!', etc.)

After this, the production release would be scheduled.

xvi. Does your solution provide any automated testing capabilities? If so, please describe in details.

Answer: Continuous Integration tools like Jenkins and Fusion ITAS are used by some of our clients. Jenkins monitors the develop environment/branch and once a change is made immediately runs a CI build that executes any tests that don't require Hadoop.

xvii. Does your solution provide to compile the code (especially DSLs (Domain Specific Languages) like PIG, Hive) without running? Ex - RDBMS system provides the feature where, we can compile all objects to know if there an error in code.

Answer: Both Apache Pig and Hive provide the ability to run an Explain plan prior to running the query. Running an explain plan will attempt to compile the code and notify the user if there are any syntax errors in his/her code. Explain plans are available for both tools through the HUE Web UI.

xviii. Does your solution provide ability to perform on-line upgrades? Please describe upgrade process in detail.

Answer: HDP provides the option of installing/upgrading from our hosted, on-line repository or via a local, mirrored repository. In our experience, the on-line repository is used for transient clusters and POC while 'Local repositories' are used for production deployments (installing a cluster from a local repository means that the software is only downloaded from the internet **once** and hosted on a local web server, rather than being downloaded **for every node in the cluster**. Local repositories also enable organizations to practice rigid version control of their clusters; by maintaining a local repository organizations can guarantee that addition nodes added later will be using exactly the same version of software as the original nodes whereas nodes installed from the online repository may be of a later point release)

The process for upgrading Ambari (the cluster installation, management & operations component) to the latest GA version Ambari 1.7 and the cluster's components to the latest GA release of HDP 2.2 are described in detail here.

However please note that HDP 2.2 introduces "rolling upgrades" which means that the upgrade procedure will change for any new releases from now on. HDP 2.2 marks the first release where HDP rpms, debs, and directories contain versions in the names to permit side-by-side installations of later HDP releases. To transition between previous releases and HDP 2.2, Hortonworks provides hdp-select, a script that symlinks your directories to hdp/current and lets you maintain using the same binary and configuration paths that you were using before. The following instructions have you remove your older version HDP components, install hdp-select, and install HDP 2.2 components to prepare for rolling upgrade.

xix. Do you provide any tool which can enforce the governance (not the process side but technology side)?

Answer: If by referring to 'technology goverance' you mean, "Information and technology (IT) governance" (i.e. *a framework used to identify, establish and link the mechanisms to oversee the use of information and related technology to create value and manage the risks associated with using information and technology*) then we provide central security policy administration across the core enterprise security requirements of authorization, accounting and data protection through **Apache Ranger**. Apache Ranger offers a centralized security framework to manage fine-grained access control over almost all of the HDP data access components such as Apache Hive and Apache HBase. Using the Apache Ranger console, security administrators can comprehensively manage policies for access to files, folders, databases, tables, or column. These policies can be set for individual users or groups and then enforced within Hadoop.

Security administrators can also use Apache Ranger to manage audit tracking and policy analytics for deeper control of the environment. The solution also provides an option to delegate administration of certain data to other group owners, with the aim of securely decentralizing data ownership.

HDP also provides **Apache Falcon** which is one of the key component for data governance. Falcon provides a framework for simplifying and orchestrating data management and pipeline processing in Apache Hadoop. It enables automation of data movement and processing for ingest, pipelines, replication and compliance use cases. Falcon also leverages its integration with YARN—the architectural center of Hadoop—to centrally manage the cluster's data governance, maximize data pipeline reuse and enforce consistent data lifecycles. Planned enhancements to Falcon include automated security policies based upon tagging of ingested data furthering our story around technology governance.

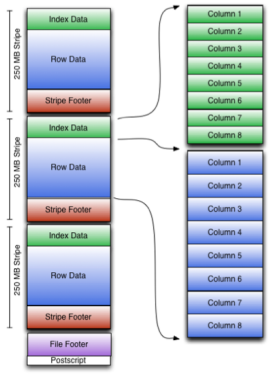
xx. What are the options available to index the data for faster access?

Answer: The defacto SQL standard to access data in hadoop is Hive. Hive provides a SQL interface data stored within HDFS. The Optimized Row Columnar (ORC) file format provides a highly efficient way to store Hive data. It was designed to overcome limitations of the other Hive file formats. Using ORC files improves performance when Hive is reading, writing, and processing data.

Compared with RCFile format, for example, ORC file format has many advantages such as:

- a single file as the output of each task, which reduces the NameNode's load
- Hive type support including datetime, decimal, and the complex types (struct, list, map, and union)
- **light-weight indexes stored within the file**
  - **skip row groups that don't pass predicate filtering**
  - **seek to a given row**
- block-mode compression based on data type
  - run-length encoding for integer columns
  - dictionary encoding for string columns
- concurrent reads of the same file using separate RecordReaders
- ability to split files without scanning for markers
- bound the amount of memory needed for reading or writing
- metadata stored using Protocol Buffers, which allows addition and removal of fields

To efficiently read data from HDFS, we have added indexes to ORC File. Because the speed of loading data into Hive and storage efficiency are important to a file format in Hive, in the design of ORC File, we decided to only use sparse indexes. The two kinds of sparse indexes are *data statistics* and *position pointers*. The following diagram illustrates the ORC file structure, indicating the lightweight indexing mechanism:



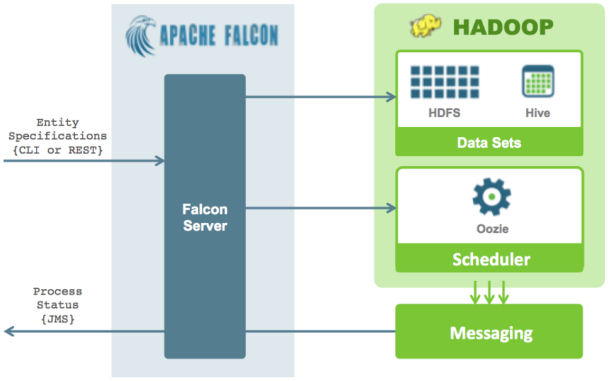
As shown in the diagram, each stripe in an ORC file holds index data, row data, and a stripe footer. The stripe footer contains a directory of stream locations. Row data is used in table scans. Index data includes min and max values for each column and the row positions within each column. (A bit field or bloom filter could also be included.) Row index entries provide offsets that enable seeking to the right compression block and byte within a decompressed block. Note that ORC indexes are used only for the selection of stripes and row groups and not for answering queries.

Having relatively frequent row index entries enables row-skipping within a stripe for rapid reads, despite large stripe sizes. By default every 10,000 rows can be skipped. With the ability to skip large sets of rows based on filter predicates, you can sort a table on its secondary keys to achieve a big reduction in execution time. For example, if the primary partition is transaction date, the table can be sorted on state, zip code, and last name. Then looking for records in one state will skip the records of all other states.

xxi. What option you provide for workflow management besides Oozie?

Answer: Falcon simplifies the development and management of data processing pipelines with a higher layer of abstraction, taking the complex coding out of data processing applications by providing out-of-the-box data management services. This simplifies the configuration and orchestration of data motion, disaster recovery and data retention workflows.

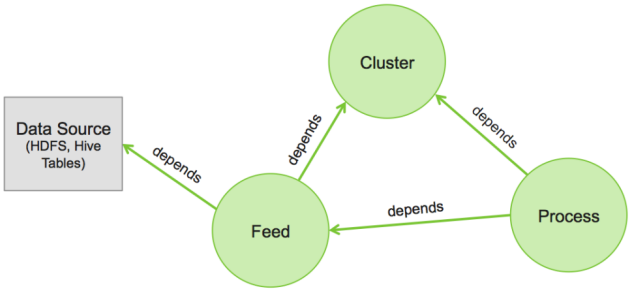
Falcon runs as a standalone server as part of your Hadoop cluster.



A user creates entity specifications and submits to Falcon using the Command Line Interface (CLI) or REST API. Falcon transforms the entity specifications into repeated actions through a Hadoop

workflow scheduler. All the functions and workflow state management requirements are delegated to the scheduler. By default, Falcon internally uses Apache Oozie as it's scheduler underneath.

The following diagram illustrates the entities defined as part of the Falcon framework:



- Cluster: Represents the "interfaces" to a Hadoop cluster
- Feed: Defines a dataset (such as HDFS files or Hive tables) with location, replication schedule and retention policy.
- Process: consumes Feeds and processes Feeds

xxii. What option(s) your provide to facilitate data extraction/ingestion besides Sqoop/Flume like projects?

Answer: *Apache Kafka* is a fast, scalable, durable, and fault-tolerant publish-subscribe messaging system. Kafka is often used in place of traditional message brokers like JMS and AMQP because of its higher throughput, reliability and replication.

Apache Kafka supports a wide range of use cases as a general-purpose messaging system for scenarios where high throughput, reliable delivery, and horizontal scalability are important. Apache Storm and Apache HBase both work very well in combination with Kafka. Common use cases include:

- Stream Processing
- Website Activity Tracking
- Metrics Collection and Monitoring
- Log Aggregation

Some of the important characteristics that make Kafka such an attractive option for these use cases include the following:

- **Scalability**: Distributed system scales easily with no downtime
- **Durability**: Persists messages on disk, and provides intra-cluster replication
- **Reliability**: Replicates data, supports multiple subscribers, and automatically balances consumers in case of failure
- **Performance**: High throughput for both publishing and subscribing, with disk structures that provide constant performance even with many terabytes of stored messages

Kafka's system design can be thought of as that of a distributed commit log, where incoming data is written sequentially to disk. There are four main components involved in moving data in and out of Kafka:

- Topics
- Producers
- Consumers
- Brokers

In Kafka, a *Topic* is a user-defined category to which messages are published. Kafka *Producers* publish messages to one or more topics and *Consumers* subscribe to topics and process the published

messages. Finally, a Kafka cluster consists of one or more servers, called *Brokers* that manage the persistence and replication of message data (i.e. the commit log).

One of the keys to Kafka's high performance is the simplicity of the brokers' responsibilities. In Kafka, topics consist of one or more Partitions that are ordered, immutable sequences of messages. Since writes to a partition are sequential, this design greatly reduces the number of hard disk seeks (with their resulting latency).

Another factor contributing to Kafka's performance and scalability is the fact that Kafka brokers are not responsible for keeping track of what messages have been consumed – that responsibility falls on the consumer. In traditional messaging systems such as JMS, the broker bore this responsibility, severely limiting the system's ability to scale as the number of consumers increased.

xxiii. What options does your solution support for data compression. Please describe.

Answer: We introduced indexes and compression schemes in the ORC File Format. ORC File uses a two-level compression scheme. A stream is first encoded by a stream type specific data encoding scheme. Then, an optional general-purpose data compression scheme can be used to further compress this stream. For a column, it is stored in one or multiple streams. Based on the type of a stream, we can divide streams to four primitive types. Based on its type, a stream has its own data encoding scheme. These four types of primitive streams are introduced as follows.

- Byte Stream: A byte stream basically stores a sequence of bytes and it does not encode data.
- Run Length Byte Stream: A run length byte stream stores a sequence of bytes. For a sequence of identical bytes, it stores the repeated byte and the occurrences.
- Integer Stream: An integer stream stores a sequence of integers. It can encode these integers with run length encoding and delta encoding. The specific encoding schemes used for a sub-sequence of integers are determined based on the pattern of it.
- Bit Field Stream: A bit field stream is used to store a sequence of boolean values. In this stream, a bit represents a boolean value. Under the cover, a bit field stream is backed by a run length byte stream.

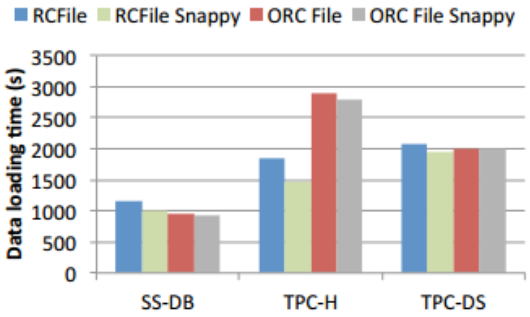
In ORC File, besides those stream type specific schemes, users can further ask the writer of an ORC file to compress streams with a general-purpose codec among ZLIB, Snappy and LZO. For a stream, the general-purpose codec will compress this stream to multiple small compression units. In the current implementation, the default size of a compression unit is 256 KB.

During testing we compared the sizes of datasets of SS-DB, TPC-H and TPC-DS stored by RCFile and ORC. For each file format, we also stored datasets with and without using Snappy compression (referred to as RCFile Snappy and ORC File Snappy). Without Snappy compression, datasets of SS-DB and TPC-DS have already had smaller sizes than RCFile with Snappy, which shows the effectiveness of data type specific encoding schemes in ORC File. With Snappy compression, sizes of datasets stored with ORC File were further reduced. For datasets of SS-DB and TPC-DS, this further reduction on sizes is not as significant as that shown in the dataset of TPC-H. It is because every table in TPC-H has a column of comment, which contains random strings. The cardinality of
such a column is high and the dictionary encoding scheme is not effective. Thus, the size of the dataset of TPC-H can be significantly reduced by using a general-purpose data compression technique, such as Snappy.

Sizes of datasets (GB) stored by Text, RCFile, RCFile with Snappy compression, ORC File, and ORC File with Snappy compression.

Compression type	SS-DB	TPC-H	TPC-DS
Text	248.35	323.84	279.87
RCFile	128.23	269.00	159.69
RCFile Snappy	55.15	118.33	105.28
ORC File	53.51	168.96	102.24
ORC File Snappy	39.20	86.67	94.05

Elapsed times of loading datasets to file formats of RCFile, RCFile with Snappy compression, ORC File, and ORC File with Snappy compression.



Taken from "Major Technical Advancements in Apache Hive"

xxiv. Does the solution provide adequate codecs (splittable) tocompress/uncompress data?

Answer: Within an ORC file, Streams are compressed using a codec, which is specified as a table property for all streams in that table To optimize memory use, compression is done incrementally as each block is produced. Compressed blocks can be jumped over without first having to be decompressed for scanning. Positions in the stream are represented by a block start location and an offset into the block.

The codec can be Snappy, Zlib (default), or *none*.

- Snappy is significantly faster than LZO for decompression, and comparable for compression, meaning the total round-trip time is superior. Second, Snappy is BSD-licensed, which means that it can be shipped with Hadoop, unlike LZO which is GPL-licensed, and therefore has to be downloaded and installed separately since it may not be included in Apache products.

xxv. What are the main roles required to operate your solution. What type of skill sets is required for each of these roles? Who a typically fills this roles? Please describe the day in the life of each of these roles.

Answer: The following table maps existing technical resources to their new role in working with the data lake:

Existing Role	**New Role**

Java data developer	Java data developer (using Java, Scala, MapReduce, etc. where appropriate)
Linux system administrator (DB servers and the like)	Linux systems administrator (HDFS and YARN nodes are analogous to database servers)
DBA (managing server optimization, scheduling, tuning, and DB security)	Hive, Falcon, HCat and YARN administrator managing schemas, pipeline automation, workload prioritization, etc.)
DB Dev (managing stored proc development and SQL table design, etc.)	Hive Table and UDF development (esp. if they are used to writing stored procs in Java)
Network administrator	Same role
Datacenter administrator	Same role
DevOps leads	Hadoop overall cluster health, debugging experts, etc.
Architects	Silo designers (select application stack that runs in YARN, size the cluster resources necessary, design security regimens, etc.)
Stats analytics, quants, and scientists	Same role

xxvi. How do you estimate the number of FTE's required to fill in each of the roles? What are the drivers for the number of FTE's?

Answer: Without knowing more about the scale of operations, we only only estimate that the number of FTEs would be similar in number to the existing roles above.

xxvii. Please provide training options for each of the roles and their associated costs.

Answer: **Hortonworks University**
We offer a wide range of training options backed by experts and designed to evolve your teams Hadoop proficiency
Custom Coursework

- On-site training for your team
- Customized for your requirements

Public Courses ($2,500-$3,000 per student)

- Offered in all geographies
- Hadoop Architect
- Hadoop Developer
- Hadoop Analyst
- Hadoop Operations
- Data Science

Hortonworks Course Offerings: ($20,000-25,000 per 10 students)

Apache Hadoop Essentials 2.0: This one-day course provides a technical overview of Apache Hadoop for decision makers and business users. Students will obtain a deeper understanding of what

is Big Data, Hadoop 2.0, the architecture and various technologies in the Hadoop ecosystem, and the business value that Hadoop provides.

Operations Management with the Hortonworks Data Platform: This 4-day course covers administration tasks for Hadoop 2.0 clusters.
The course presents content related to the deployment lifecycle for a multi-node Hadoop cluster including: installation, configuration, monitoring, scaling and how Hadoop works with Big Data

Developing Solutions for Apache Hadoop on Windows: Students will learn to develop applications and analyze big data stored in Apache Hadoop running on Microsoft Windows. Students will learn the details of the Hadoop Distributed File System (HDFS ™) architecture and MapReduce framework, as well as learn how to develop applications on Hadoop® using tools like C#, Pig ™, Hive™, HCatalog, Sqoop, Oozie and Microsoft Excel.

Data Science for the Hortonworks Data Platform: This 3-day hands-on training course teaches the fundamentals of Data Science and how to apply those concepts in Hadoop using machine learning, Mahout, Pig, Python and various machine learning libraries like SciPy and Scikit-Learn. Data Science for the Hortonworks Data Platform covers data science principles and techniques through lecture and hands-on experience. During this three-day class, students will learn the processes and practice of data science, including machine learning and natural language processing. Students will also learn the tools and programming languages used by data scientists, including Python, IPython, Mahout, Pig, NumPy, pandas, SciPy, Scikit-Learn and Spark MLlib.

Apache Hadoop 2.0: Data Analysis with the Hortonworks Data Platform using Pig and Hive: This 4-day hands-on training course teaches students how to develop applications and analyze Big Data stored in Apache Hadoop 2.0 using Pig and Hive. Students will learn the details of Hadoop 2.0, YARN, the Hadoop Distributed File System (HDFS), an overview of MapReduce, and a deep dive into using Pig and Hive to perform data analytics on Big Data. Other topics covered include data ingestion using Sqoop and Flume, and defining workflow using Oozie.

Apache Hadoop 2.0: Developing Applications with the Hortonworks Data Platform using Java: This advanced four-day course provides Java programmers a deep-dive into Hadoop 2.0 application development. Students will learn how to design and develop efficient and effective MapReduce applications for Hadoop 2.0 using the Hortonworks Data Platform. Students who attend this course will learn how to harness the power of Hadoop 2.0 to manipulate, analyze and perform computations on their Big Data.

Developing Custom YARN Applications: This 2-day hands-on training course teaches students how to develop custom YARN applications for Apache Hadoop. Students will learn the details of the YARN architecture, the steps involved in writing a YARN application, the details of writing a YARN client and ApplicationMaster, and how to launch Containers. Applications are developed using Eclipse and Gradle connected to a 7-node HDP 2.1 cluster running in a virtual machine that the students can keep for use after the training.

xxviii. Please describe the main user groups that your company is involved in.

Answer: Hortonworks is involved in many user groups around the country and world. We chair and sponsor many Hadoop User Groups, NoSQl user groups, Big Data Users Groups, and many others. Hortonworks also leads industry wide consortiums to enable customers and partners to better define the technology roadmap to meet their business needs. An example of these are the Open Data Platform and Data Governance Inititaive. Open Data Platform initiative enables collaboration between vendors and end users of Big Data technology. Founding members are Hortonworks and

Pivotal. Data Governance Initiative addresses enterprise requirements for comprehensive data governance. Founding members are Aetna, Merck, Target, Hortonworks and SAS.

xxix. Does your company host a trade show (ex. Informatica World)?

Answer: Hortonworks hosts Hadoop World in San Jose and in Europe.

xxx. Please describe the metadata management capabilities for your solution? Can your solution handle traditional RDBMS's (ex. SqlServer, Oracle) as well? If so, please provide which ones and the versions that are supported. Please describe how it works.

Answer: Metadata is managed by Apache HCatalog in HDP and is consistently available across all toolsets native in the platform, externally connected BI toolsets, and any other 3rd party consumers through a REST API. Since HCatalog information is available easily via multiple means, it simplifies metadata documentation as well.

Key features of the HDP HCatalog include:

- Creates metadata on top of the raw data in HDFS
- Location of raw data is abstracted from consumers
- Schema structure defined as tables and partitions – i.e., a "table-centric" view that is easy for clients to understand and consume
- Supports "late-binding" – e.g., can create new schemas and data models at any time on top of the raw data. Multiple and/or changing bindings are supported.
- Create partitions within tables to logically separate data
    - Partitions can have different schemas and formats as source data evolves without affecting consumers of the data
    - E.g., Partition '2012-01-01' of Table X can have schema with 30 fields while partition '2013-01-01' can have 35 fields
- Can provide notifications in support of monitoring, dashboards, etc.

From a client data access perspective, the metadata in the catalog can be used by the client portal to describe the data sets and structures that are available. In addition, meta-data stored alongside the data itself can be provided to clients as part of data provisioning.

xxxi. Does your solution provide capabilities for mapping data lineages? Does it have the ability to include non-Apache items (ex. Informatica, Oracle, etc?)? If so, please provide which ones and the versions that are supported. Please describe how it works.

Answer: HDP comes with Apache Falcon that provides data pipeline lineage, so that you can look at how dataset reached a particular state. With 3rd party tools such as *Informatica BDE* or *Talend* you can build data integration jobs that span both HDP and external sources gathering data lineage at the same time.

xxxii. Does your solution provide data presentation capabilities? Please describe the capabilities.

Answer: HDP is more of a data platform and you can use Hue to provide a GUI based data presentation layer. However HDP does come with ODBC/JDBC drivers that could be used by your enterprise data presentation tools to provide a data presentation that meets your enterprise standards.

xxxiii. What 3rd party data presentation engines are most commonly used with your distribution?

Answer: All the major BI vendors offer Hadoop integration (using hive ODBC/JDBC drivers), and specialized analytics vendors offer niche solutions for specific data types and use cases (for example, such as Platfora and Datameer). We also maintain a strong technological partnership with SAS who have YARN-enabled their applications. Please refer to www.hortonworks.com/partners for complete list.

xxxiv. Does your solution allow connectivity from BI tools such as Cognos and Tableau, and if so, please describe how.

Answer: Cognos and Tableau can connect to HDP via HiveServer2's JDBC/ODBC 3.52 standard compliant driver (supported on Linux & Windows) that Hortonworks supplies. Hive provides a SQL interface to data stored in HDP and with the help of the drivers any standard BI tool could be used for data presentation layer.

e. Analytics

i. Do you provide a faster SQL-like environment than Hive, explain in detail?

Answer: Since the completion of the Stinger initiative Hive performance has improved over 100x by using Tez and the ORC file format, amongst other things.

Apacha Tez is an extensible framework for building high performance batch and interactive data processing applications, coordinated by YARN in Apache Hadoop. Tez improves the MapReduce paradigm by dramatically improving its speed, while maintaining MapReduce's ability to scale to petabytes of data. Important Hadoop ecosystem projects like Apache Hive and Apache Pig use Apache Tez, as do a growing number of third party data access applications developed for the broader Hadoop ecosystem

Furthermore, we have since embarked on on a sequel to the stinger initiative, called Stinger.next, which once completed (scheduled for H2 2015) Hive will be able to provide sub-second response times.

In addition to greatly improving the performance of Hive, HDP 2.2 supports Spark which includes the Spark SQL Module. Furthermore through our partnership with Pivotal, HAWQ is currently being certified to run on top of HDP. We can run HP Vertica on HDP with our co engineering partnership with HP.

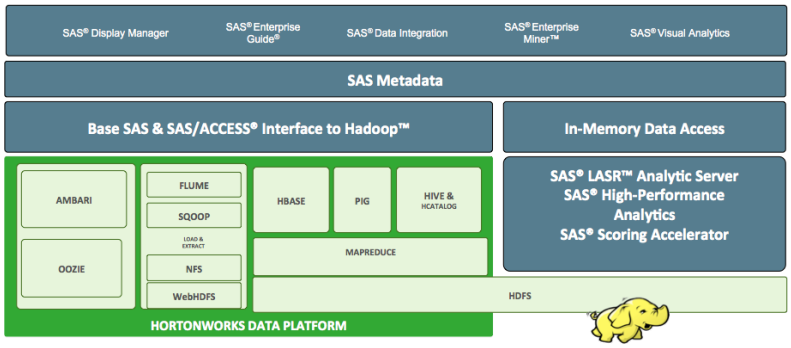
ii. Do you support R/SAS code to be executed in the cluster, explain in detail?

1. What machine learning algorithms do you support?
2. What statistical procedures do you support?
3. What data mining methodologies do you support?

Answer: The kinds of analytics that can and will be done in Hadoop are endless. Fundamentally there are four categories of purposes or outputs from analytics in Hadoop.

- Data discovery – the purpose of data discovery is to understand what is contained in your data.
    - For example:
        - Summaries
        - clusters/ groupings
        - assessments of completeness
- Build a model so that the model can be deployed
    - Either in Hadoop or
    - Somewhere else, such as in an enterprise data warehouse, real-time transaction system such as a payment processing system, website, call center application, claims system, etc.
- Generate a results table – use an analytic technique to make more meaningful data from the "raw data" in Hadoop. For example,
    - For every attribute, calculate its predictive value (importance)
    - For every observation, generate a prediction or score
    - Likelihood to churn
    - Likelihood to buy
    - Likelihood of being fraudulent
    - Likelihood to fail within the next month
- Visualize the results of a model – run the model and show the result to people directly using a Business Intelligence tool, for example.

SAS



HDP supports all of the SAS products listed below. They are fully certified on the platform enabling all of their PROCs and other algorithms to be leveraged as normal based on the tool's access method.

Examples of these algorithms are listed below:

Restricted Boltzmann Machines
K-Means Clustering
Fuzzy K-Means
Canopy Clustering
Mean Shift Clustering
Hierarchical Clustering
Dirichlet Process Clustering
Latent Dirichlet Allocation
Collocations
Dimensional Reduction
Expectation Maximization
Bayesian
Gaussian Discriminative Analysis

Logistic Regression
Neural Network
Independent Component Analysis
Random Forests
Boosting
Principal Components Analysis
Online Viterbi
Parallel Viterbi
Locally Weighted Linear Regression
Hidden Markov Models
Perceptron and Winnow
Support Vector Machines
Minhash Clustering
Spectral Clustering
Parallel Frequent Pattern Mining
Online Passive Aggressive
RowSimilarityJob
Collaborative Filtering with ALS-WR
Machine Learning Resources
Top Down Clustering
Itembased Collaborative Filtering
Stochastic Singular Value Decomposition

SAS Access

- SAS Access provides real time data access to 3rd party storage services. SAS Access integrates with HiveServer2, enabling access to hive tables as if they were native SAS data sets. The SAS/ACCESS interface to Hadoop allows SAS to access the data in Hadoop and integrate it with data from other sources. SAS/ACCESS makes Hive-based tables appear native to SAS, providing seamless connectivity to Hadoop from any SAS product or capability.

SAS Base

- *SAS Base is the core statistical modeling and analysis tool. SAS Base provides direct integration with HDFS, MapReduce & Pig, enabling users to leverage hadoop's scale-out processing from within SAS environment.*

SAS LASR

- *SAS LASR is a scale out in-memory grid for deep analytical processing. SAS Visual Analytics (VA) integrates with SAS LASR for low latency data access and analysis. SAS LASR integrates with HDFS using parallel data access for fast bi-directional data movement between SAS LASR & Hadoop. Today SAS LASR can also be deployed directly inside of YARN enabling it to be managed by the same scheduling processes as other jobs running within the cluster.*

SAS Data Management

- *SAS Data Management is an ETL engine. It supports Hive & HDFS access along with MapReduce and Pig job submission.*

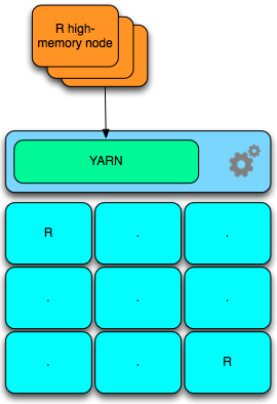
SAS HPA

- *SAS HPA is a MPI grid, supporting computation of advanced statistical models using scale out architecture. SAS HPA integrates with HDFS using parallel data access for fast bi-directional data movement between SAS HPA & Hadoop.*

Project R (CRAN)

R is an open source statistical language that is used by more than 2 million people worldwide. It contains thousands of packages for all kinds of data mining, predictive statistics and machine learning techniques. As a scripting language, it is open and offers endless possibilities for data

analysis and model building. It is, however, memory constrained and therefore is rarely used for Big Data Big Analytics.



Interfaces to Hadoop

- RMR: run map-reduce with R
- RHDFS: access HDFS files from R
- RHIVE: run hive queries from R
- RHBase: Hbase from R

The R language can be installed into an HDP installation in a fairly straightforward manner. This is done by installing R and related packages on the nodes which the user will target with YARN labels or over all nodes in the entire cluster.

The R packages *rHDFS* and *rmr* enable R to be run in a parallel manner across the cluster. A YARN-enabled version of R will ease the dependency on rmr and distribute R applications in a Hadoop cluster more easily than today. RHive is an R extension facilitating distributed computing via HIVE query. It provides an easy to use HQL like SQL and R objects and functions in HQL. Rhbase provides database management for the HBase distributed database from within R. This interface provides basic connectivity to HBASE, using the Thrift server. R programmers can browse, read, write, and modify tables stored in HBASE.

iii. What is your roadmap for R/SAS integration, explain in detail?

Answer: Two methods for integrating R and/or SAS are a) Single-threaded on an Edge Node or b) within the cluster itself running across parallel nodes.

SAS Access

- SAS Access provides real time data access to 3rd party storage services. SAS Access integrates with HiveServer2, enabling access to hive tables as if they were native SAS data sets. The SAS/ACCESS interface to Hadoop allows SAS to access the data in Hadoop and integrate it with data from other sources. SAS/ACCESS makes Hive-based tables appear native to SAS, providing seamless connectivity to Hadoop from any SAS product or capability.

SAS Base

- *SAS Base is the core statistical modeling and analysis tool. SAS Base provides direct integration with HDFS, MapReduce & Pig, enabling users to leverage hadoop's scale-out processing from within SAS environment.*

SAS LASR

- *SAS LASR is a scale out in-memory grid for deep analytical processing. SAS Visual Analytics (VA) integrates with SAS LASR for low latency data access and analysis. SAS LASR integrates with HDFS using parallel data access for fast bi-directional data movement between SAS LASR & Hadoop. Today SAS LASR can also be deployed directly inside of YARN enabling it to be managed by the same scheduling processes as other jobs running within the cluster.*

SAS Data Management

- *SAS Data Management is an ETL engine. It supports Hive & HDFS access along with MapReduce and Pig job submission.*

SAS HPA

SAS HPA is a MPI grid, supporting computation of advanced statistical models using scale out architecture. SAS HPA integrates with HDFS using parallel data access for fast bi-directional data movement between SAS HPA & Hadoop

iv. Do you support HiveServer/SharkServer connectivity methods?

Answer: Yes, Hortonworks provides ODBC and JDBC Connectivity to HiveServer2 via both BINARY (native) and HTTP (via RESTful APIs such as Apache Knox) Transports

v. Do you provide support for Mahout or other machine learning libraries?

Answer: Yes HDP 2.2 not only provides support but also includes multiple machine learning/statistical libraries in it's distribution. Those libraries include Mahout (v1.0.0), Spark's MLlib (v1.2) and Pig's DataFu (v1.2)

vi. Which Streaming/Complex-event-processing (CEP) engine you support/provide?

Answer: HDP 2.2 currently supports two streaming processing engines. Apache Storm and Spark Streaming.

Apache storm: Storm is a distributed real-time computation system for processing large volumes of high-velocity data. Storm on YARN is powerful for scenarios requiring real-time analytics, machine learning and continuous monitoring of operations. Some of specific new business opportunities include: real-time customer service management, data monetization, operational dashboards, or cyber security analytics and threat detection. Because Storm integrates with YARN via Apache Slider, YARN manages Storm while also considering cluster resources for data governance, security and operations components of a modern data architecture

Apache Spark Streaming: park Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. Data can be ingested from many sources like Kafka, Flume, Twitter, ZeroMQ, Kinesis or TCP sockets can be processed using complex algorithms expressed with high-level functions like map, reduce, join and window. Finally, processed data can be pushed out to filesystems, databases, and live dashboards. In fact, you can apply Spark's machine learning and graph processing algorithms on data streams.

vii. Can you clearly describe the Apache Hadoop projects you do/do-not support?

Answer: Hortonworks only include Apache-governed projects as part of HDP. For HDP 2.2 the following Apache projects are supported and incorporated into our distribution:

- Apache Hadoop 2.6.0
- Apache DataFu 1.2.0
- Apache Falcon 0.6.0
- Apache Flume 1.5.2
- Apache HBase 0.98.4
- Apache Hive 0.14.0
- Apache Knox 0.5.0
- Apache Mahout 1.0.0
- Apache Oozie 4.1.0
- Apache Pig 0.14.0
- Apache Phoenix 4.2.0
- Apache Ranger 0.4.0
- Apache Slider 0.60
- Apache Sqoop 1.4.5
- Apache Storm 0.9.3
- Apache Tez 0.5.2
- Apache Zookeeper 3.4.6
- Apache Accumulo 1.6.1 (Linux only)
- Apache Kafka 0.8.1.1 (Linux only)

viii. Do you provide support for real time Analytics (RTE) as part of your Hadoop Offering?

Answer: HDP provides a variety of ways to do real time analytics including using Storm, Spark Streaming and Solr/Banana. Below are overviews of all three RTE offerings:

Apache Storm

Storm is a distributed real-time computation system for processing large volumes of high-velocity data. Storm on YARN is powerful for scenarios requiring real-time analytics, machine learning and continuous monitoring of operations. Some of specific new business opportunities include: real-time customer service management, data monetization, operational dashboards, or cyber security analytics and threat detection. Because Storm integrates with YARN via Apache Slider, YARN manages Storm while also considering cluster resources for data governance, security and operations components of a modern data architecture.

Storm is extremely fast, with the ability to process over a million records per second per node on a cluster of modest size. Enterprises harness this speed and combine it with other data access applications in Hadoop to prevent undesirable events or to optimize positive outcomes.
Here are some typical "prevent" and "optimize" use cases for Storm.

	"Prevent" Use Cases	"Optimize" Use Cases
Financial Services	• Securities fraud • Operational risks & compliance violations	• Order routing • Pricing
Telecom	• Security breaches • Network outages	• Bandwidth allocation • Customer service
Retail	• Shrinkage • Stock outs	• Offers • Pricing
Manufacturing	• Preventative maintenance • Quality assurance	• Supply chain optimization • Reduced plant downtime
Transportation	• Driver monitoring • Predictive maintenance	• Routes • Pricing
Web	• Application failures • Operational issues	• Personalized content

Now with Storm in Hadoop on YARN, a Hadoop cluster can efficiently process a full range of workloads from real-time to interactive to batch. Storm is simple and developers can write Storm topologies using any programming language.
Five characteristics make Storm ideal for real-time data processing workloads. Storm is:

- Fast – benchmarked as processing one million 100 byte messages per second per node
- Scalable – with parallel calculations that run across a cluster of machines
- Fault-tolerant – when workers die, Storm will automatically restart them. If a node dies, the worker will be restarted on another node.
- Reliable – Storm guarantees that each unit of data (tuple) will be processed at least once or exactly once. Messages are only replayed when there are failures.
- Easy to operate – standard configurations are suitable for production on day one. Once deployed, Storm is easy to operate.

Apache Solr
Apache Solr is the open source platform for searches of data stored in HDFS in Hadoop. Solr powers the search and navigation features of many of the world's largest Internet sites, enabling powerful full-text search and near real-time indexing. Whether users search for tabular, text, geo-location or sensor data in Hadoop, they find it quickly with Apache Solr.

What Solr Does
Hadoop operators put documents in Apache Solr by "indexing" via XML, JSON, CSV or binary over HTTP.
Then users can query those petabytes of data via HTTP GET. They can receive XML, JSON, CSV or binary results. Apache Solr is optimized for high volume web traffic.
Top features include:

- Advanced full-text search
- Near real-time indexing
- Standards-based open interfaces like XML, JSON and HTTP
- Comprehensive HTML administration interfaces
- Server statistics exposed over JMX for monitoring
- Linearly scalable, auto index replication, auto failover and recovery
- Flexible and adaptable, with XML configuration

Solr is highly reliable, scalable and fault tolerant. Both data analysts and developers in the open source community trust Solr's distributed indexing, replication and load-balanced querying capabilities.

Spark Streaming
Spark Streaming is a library built on the core Apache Spark processing engine. It provides micro-batched processing of data as it is being ingested into the cluster. By running on Spark, Spark Streaming lets you reuse the same code for batch processing, join streams against historical data, or run ad-hoc queries on stream state. Build powerful interactive applications, not just analytics.

f. Implementation
   i. What type of implementation services is offered? Please describe in details.

Answer: Hortonworks Professional Services. The team works as a unit during the development of the modern data architecture and places an emphasis on understanding how Hadoop is going to be operationalized.

Hortonworks Service Offerings:

Hortonworks has developed a structured yet flexible methodology for the implementation of Hortonworks Data Platform with our clients. This methodology is based on lessons learned from an array of successful enterprise engagements. Effective customer training is at the forefront of this methodology followed by the formation of a Center of Excellence that together set the stage for successful phased implementation. In addition to being a framework for the successful development of the modern data architecture, the Hortonworks methodology places special emphasis on organizational preparedness to ensure the greatest chance of maximum long-term return on a client's investment.

Hortonworks Professional Services offers a comprehensive set of mature offerings built from our own experiences with the largest and most complex Hadoop clusters in the world. Our team can support projects targeted to any of the following areas: strategy, architecture, application design, proof of concepts or pilots, installations, use case capability development and data science.

Upon completion of the platform engineering work streams, clients work with Hortonworks on the application/workload design aspects of the platform. In these cases, Hortonworks Hadoop Architects work with the client's architecture and development resources to brainstorm, analyze, challenge and ultimately jointly agree upon the best solution for a specific use case. There are many different tools in the Hadoop ecosystem to perform core capabilities such as data ingest or transformation and it is imperative that the right tool be leveraged for the specific data set or type, access patterns, et cetera. Most clients of Hortonworks use this as an opportunity to "ask the expert" regarding Hadoop-specific topics and questions. It also serves as a good opportunity to learn about new technologies like Apache Knox Gateway or Apache Falcon and the details and patterns used for high availability, disaster recovery, security, et cetera.

With the platform architecture defined, the corresponding hardware procured, the application/work load design approved, some customers continue to leverage Hortonworks Professional Services to configure and deploy the use case capability in conjunction with their own staff, or if a particular client has the skills in house already, they may look for Hortonworks to move into more of an advisory role for targeted architectural or design questions, while leveraging Hortonworks Support for the day-to-day operational and development support.

Hortonworks Roles and Responsibilities:
Hortonworks Professional Services resources typically fall into one of the following four categories, though given the breadth of the Hadoop ecosystem there are many sub-specialties within these respective groups.

Hadoop Architect

- Assess the readiness of the technical environment, including the systems, tools, data, and analytical requirements to support various projects of the initiative
- Propose, recommend or facilitate the selection of appropriate tools, techniques, and resources for the capability and business applications
- Establish the reference architecture, processes, standards, and technical framework
- Define the overall architecture for the solutions/business applications and prepare the environment to support the analytical use cases
- Work with client analysts to define and execute the cases, including the provision of data and analytics subject matter expertise to ensure practical hypotheses, approaches, designs, and applications are pursued

Hadoop Solutions Consultant

- Provide design and development expertise for large-scale, clustered data processing systems
- Resolve technical issues in the environment
- Assist with the preparation of technical deliverables, and review and demonstrate the system and applications

Data Scientist

- Design and develop statistical procedures and algorithms around data sources
- Recommend and build models for various data studies, data discovery and hypothesis
- Implement any software required for accessing and handling data appropriately
- Work with developers to integrate and pre-process data for inputs into models

Project Manager

- Define, create and manage the project plans for the entire duration of the project
- Coordinate resource scheduling for onsite-offsite work
- Initiate and manage escalation procedures should such be needed for any variance between actuals and plans

ii. What are the most common implementation approaches (in house led, SI partner supported, complete outsourced, etc.) that people use in deploying you solution? Please describe.

Answer: Hortonworks utilizes in-house resources as well as qualified SI partners to deliver its consulting projects. The mix is determined based on project skill set, duration and staffing availability.

iii. What are the most common implementation mistakes made? Please describe.

Answer: The most common mistakes are not clearly defining the use cases and SLAs in the architecture sessions. Without proper planning incorrect implementation plans may be implemented.

iv. Please describe the typical implementation process. Please provide typical timelines. What are the biggest factors that influence timelines?

Answer: Implementation timeframes will vary from client to client based on complexity, integration work, use case work, number of nodes, clusters, data sizes. The biggest factors are use case definition, complexity of existing tool integration, and pre-requisites being met by the client.

v. What is the biggest implementation you have done for a client – Data volume, cluster size?

Answer: Hortonworks has deployed 1,700 node clusters and implemented clusters of over 4 PB of data.

SECTION 8 – USCC AGREEMENT TERMS AND CONDITIONS

8.1 No Executed Master with USCC

8.2 Please see attached Master with comments

NOTE: This MSA does not contain standard provisions specifically related to Hortonworks' services, support and intellectual property related thereto. Hortonworks has provided its standard Master Services Agreement for USCC to reference terms that are unique to Hortonworks' solution. The notes contained herein do not reflect all Hortonworks comments and edits. If awarded the contract described in the RFP, Hortonworks will negotiate in good faith certain terms and conditions with USCC to govern Hortonworks' Services and Support.]

SECTION 9 – SUPPLIER'S INFORMATION

Section 9.1 – 9.12 are completed in the Supplier Workbook

9.13 Warranties

Each Supplier should include its standard representations and warranties for services, software and equipment that apply to its response, including any optional warranties available.

Answer: Hortonworks warrants that the Services and Support will be performed by qualified personnel in a professional and workmanlike manner consistent with applicable industry standards. Customer must notify Hortonworks in writing of any alleged failure by Hortonworks to perform Support or Services in accordance with the foregoing warranty within thirty (30) days of the delivery of the affected Services or Support. Hortonworks' entire liability and Customer's sole remedy for Hortonworks' failure to perform in accordance with the above warranty shall be for Hortonworks to: (i) use commercially reasonable efforts to cure or correct such failure, or (ii) if Hortonworks is unable to cure or correct such failure, terminate the affected Services or Support and refund that portion of fees paid by Customer to Hortonworks that corresponds to such failure to perform

SECTION 10 – SUPPLIER'S FINANCIAL AND INSURANCE INFORMATION

Answers to all questions and other information requests in this Section 9 (subsections 9.1 and 9.2) are to be (a) attached as one information bundle and (b) identified as Section 9 Information and (c) included with your RFI response.

10.1 Financial Assessment of Supplier

As part of USCC's required policies, a financial assessment will be completed on each supplier. This assessment will be based on the financial information requested below and provided by each supplier.

A. Copies of your company's past two years and year to date, audited or CPA prepared financial statements (including Income Statements, Balance Sheets and Statements of Cash Flow).

Answer: Hortonworks S1 filing which includes audited financials through the first 9 months of 2015. Hortonworks recently reported Q4 and 2014 reports on February 24th, 2015. Attached are the links to both statements.

S1 Filing:
http://www.sec.gov/Archives/edgar/data/1610532/000119312514405390/d748349ds1.htm.

Q4 Earnings Press Release: Attached as Appendix

B. Description of your company's cash flow position over the past three years. Indicate specifically if your company will have enough cash to support its operations for the next twelve months.

Answer: Our cash position has increased from ~$8.7MM to $129MM from April 30, 2013 to December 31, 2014. Our operating cash flow in 2014 was $87MM which is easily covered by our cash balance. And as we have just recently become a public company, we have the ability to raise additional funds if needed. As of December 31, 2014, Hortonworks had cash and investments of $204.5 million, compared to $126.9 million as of September 30, 2014 and $38.5 million as of December 31, 2013.

C. Description of the equity ownership structure of your company: indicate if your company has a positive equity position on its balance sheet. Indicate the majority shareholders and the debt to equity ratio.

Answer: Our Shareholder Equity Position has turned positive in 2014 at $167MM. We are owned ~67% by top 6 holders including Yahoo, 2 directors and 3 other holders. Debt to equity is .53.

D. Description of any mergers, acquisitions or divestitures that occurred within the last three years, which are not public.

Answer: None

E. Description of your company's future growth potential and earnings sustainability.

Answer: The Hadoop market is estimated to sustain a compound annual growth rate of 58.2% from 2013 to 2020, and we believe that we are well positioned to take advantage of this growth. Our company has publicly guided that we will achieve profitability in Q1 2017.

F. Details on any litigation or lawsuits outstanding or pending. If no litigation or lawsuits exist please so indicate.

Answer: We are not involved in any litigation.

10.2 Insurance Requirements

A. Reference USCC's insurance requirements of the Master Agreement (Exhibit A) and

    a. Attach your company's Certificate of Insurance (COI) as requested in Exhibit A with USCC listed as an additional insured in the following manner: USCC Services, LLC, 8410 Bryn Mawr Avenue, Chicago, IL 60606. Attached