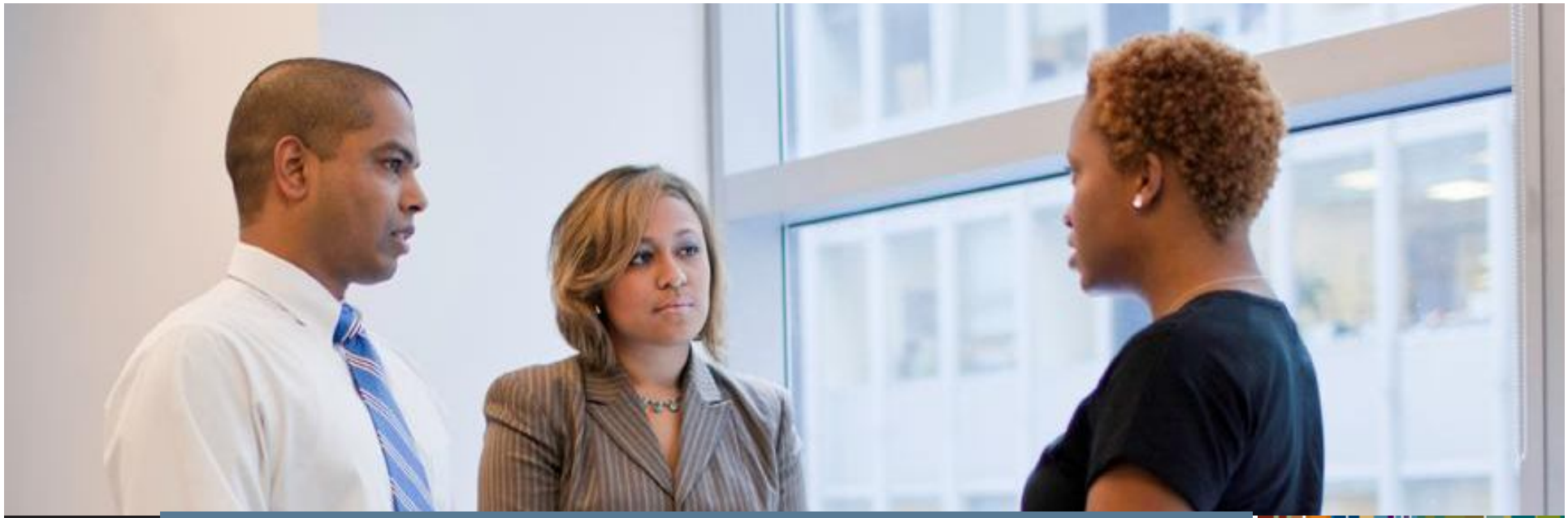


SUNGARD

Raptor: Real-time Analytics on Hadoop

Information Excellence Summit,
February 25, 2012 Bangalore
http://informationexcellence.wordpress.com/

Aditya Yadav
ATS/Research Head, SunGard India

Soundar Velu
Product Architect, Advanced Technology, SunGard

Aditya Yadav

Aditya Yadav

Head ATS/R&D

SunGard India

Aditya Yadav heads the ATS/R&D at SunGard India, An Applied Research and Consulting Group which works with Emerging Technologies like Cloud Computing, BigData/Hadoop, GPGPU, Visualization, Statistics and Analytics.

Aditya lead the team that create Raptor - Realtime Hadoop Analytics and is working now on bringing OLAP over Bigdata to the Open Source community. Aditya has earlier worked with Thoughtworks where he worked on Internet Scale Systems, Agile Coaching and Cloud Computing Evangelist.

He is an author of a dozen open source projects, 7+ books and reports. Ran a boutique consulting company and was the CTO at one of the Top 25 Indian Startups.

Soundararajan Velu

Soundararajan Velu

Product Architect
Sungard

Soundararajan Velu is a product architect with the Advanced Technology team in SunGard.

With extensive experience building enterprise applications and distributed computing products, Soundar's specialties include building large scale applications and frameworks using Apache-Hadoop and related technologies.

He consults on building low-latency applications and Service Oriented Architectures. Soundar is the creator of some of the highly reused products like SOA Accelerator, CORL Engine and Raptor.

He holds a Computer Science and Engineering degree with distinction from VTU India.

Who We Are, Who Am I, What We Do?

- Who We Are & What We Do?
  - Fortune 500 Company
  - Financial Services Firm
  - Provide software & consulting services across the industry
  - Exploring impacts of Big Data approach for last 2+ years

- Who Am I?
  - Part of Applied Research & Consulting group based out of Bangalore, India
  - We focus on latest technology trends

SUNGARD

Agenda

- Financial Services and Data Problems
- Raptor Architecture Overview
- Raptor Components
- Benchmarks
- Future Enhancements

Financial Services & Typical “data intensive” Problems

Implementation Constraints in Financial Services

Legacy Burden

Architectures, languages, tools, systems, skills

RDBMS Centricity

Constrained semantics, rigidity and specificity

Data Silos

Governance “You don’t own that data, I do”

Cost of Change

Evolution is the only answer

SUNGARD

What Did We Need?

- A generic, reliable, and cost effective analytics solution with a wide range of application areas
- Query execution and analytics at soft real-time windows (acceptable and consistent latencies)
- Minimum customization, seamless integration and ease of use
- Policy around data storage and processing
- Adaptive segmentation algorithms for optimized data search (Indexing, partitioning and filtering)

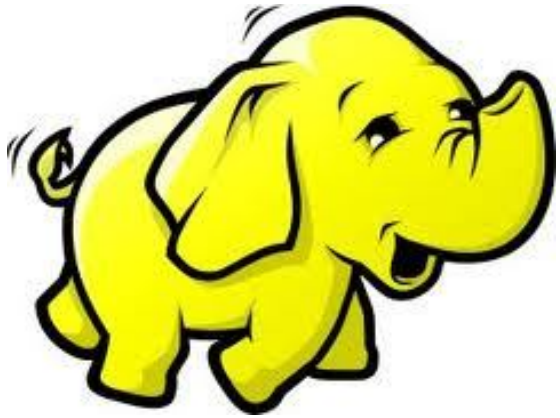
SUNGARD

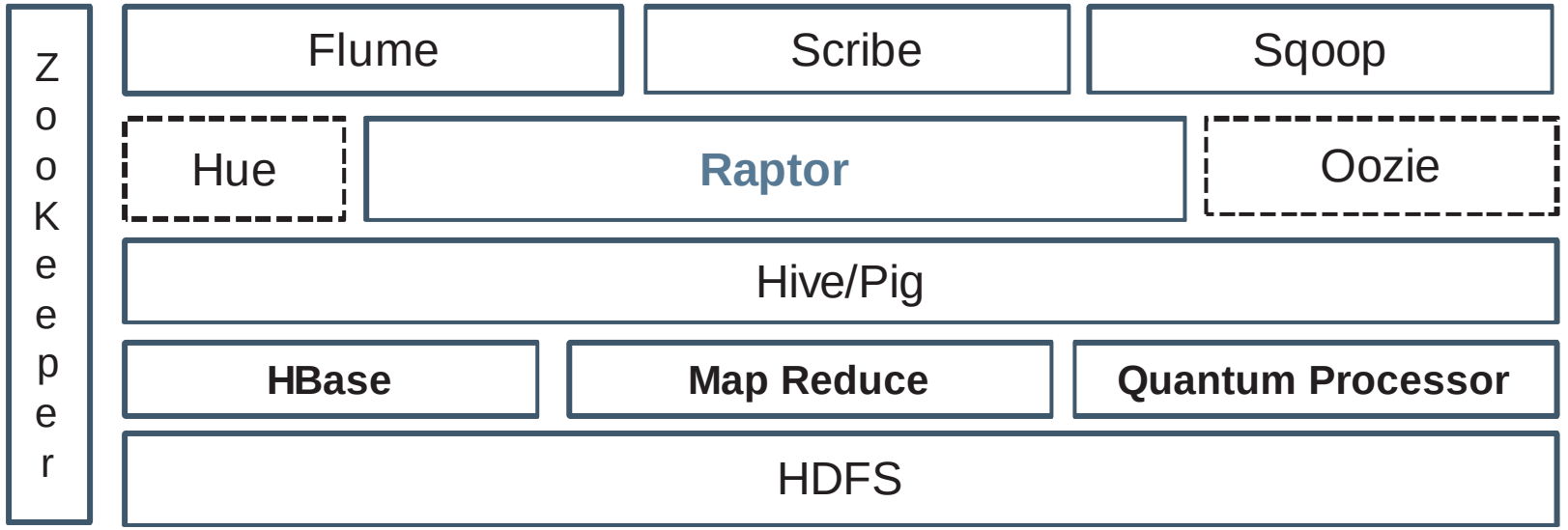
Some Limitations with Traditional Hadoop

- Jobs are executed in a brute force fashion, causing complete scans of files for every single query
- Long warm up time for jobs, performs poorly for relatively smaller data sets.
- Scheduling imbalance in HDFS operations and job execution
- Limitations around the kind of jobs that can be executed with MapReduce, (non equality joins not supported)
- Open bugs and lacking features, memory management bottlenecks
- Only for batch mode based applications, does not fit real-time analytics scenarios

SUNGARD

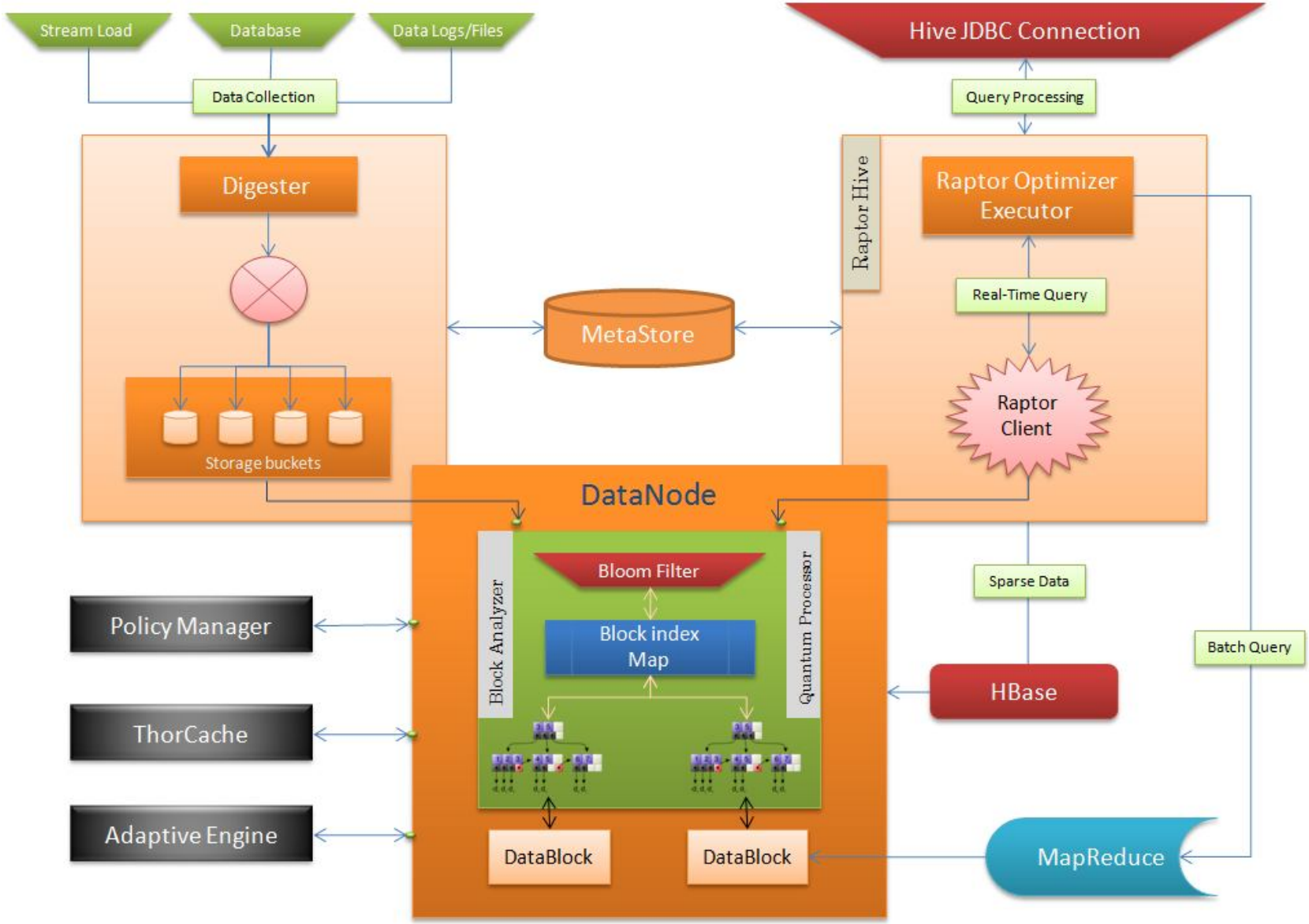
Raptor Application Stack



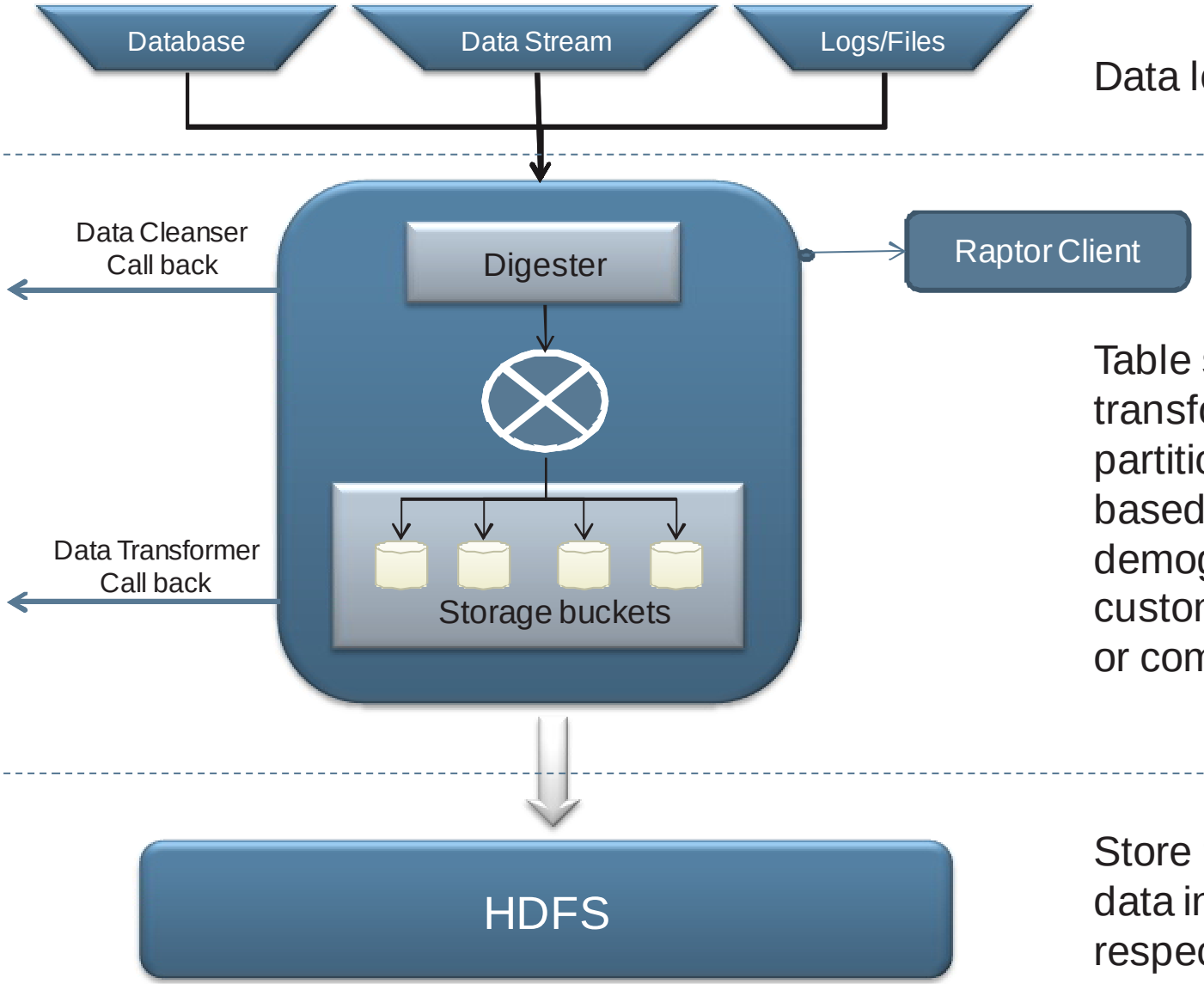


SUNGARD

Raptor Architecture



Raptor Digester - Level 1 Segmentation

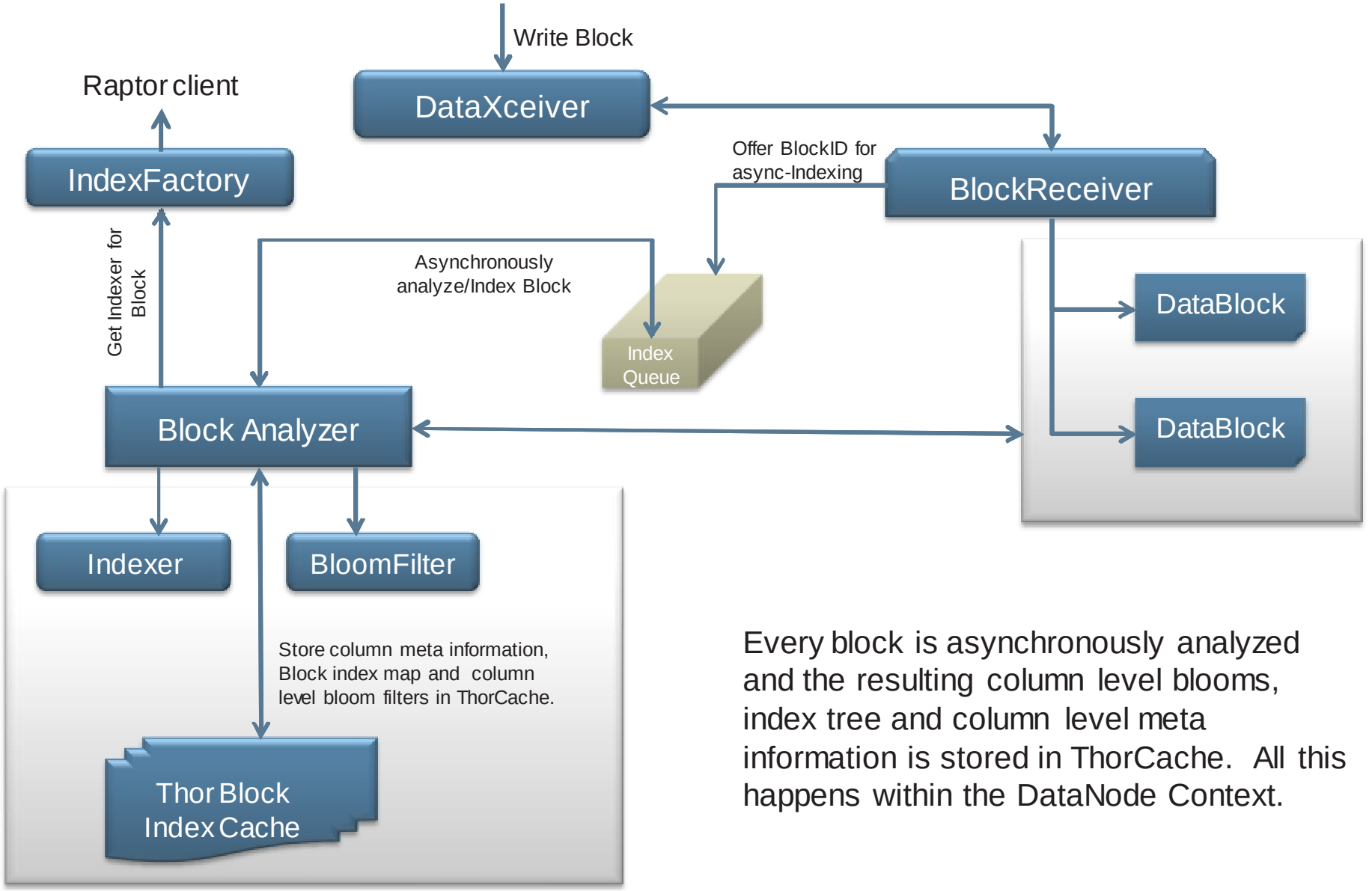


Data load adaptors

Table specific data transformer and partitioner, partition based on demographics, customers, IP address or composite columns.

Store partitioned data into respective file buckets in HDFS.

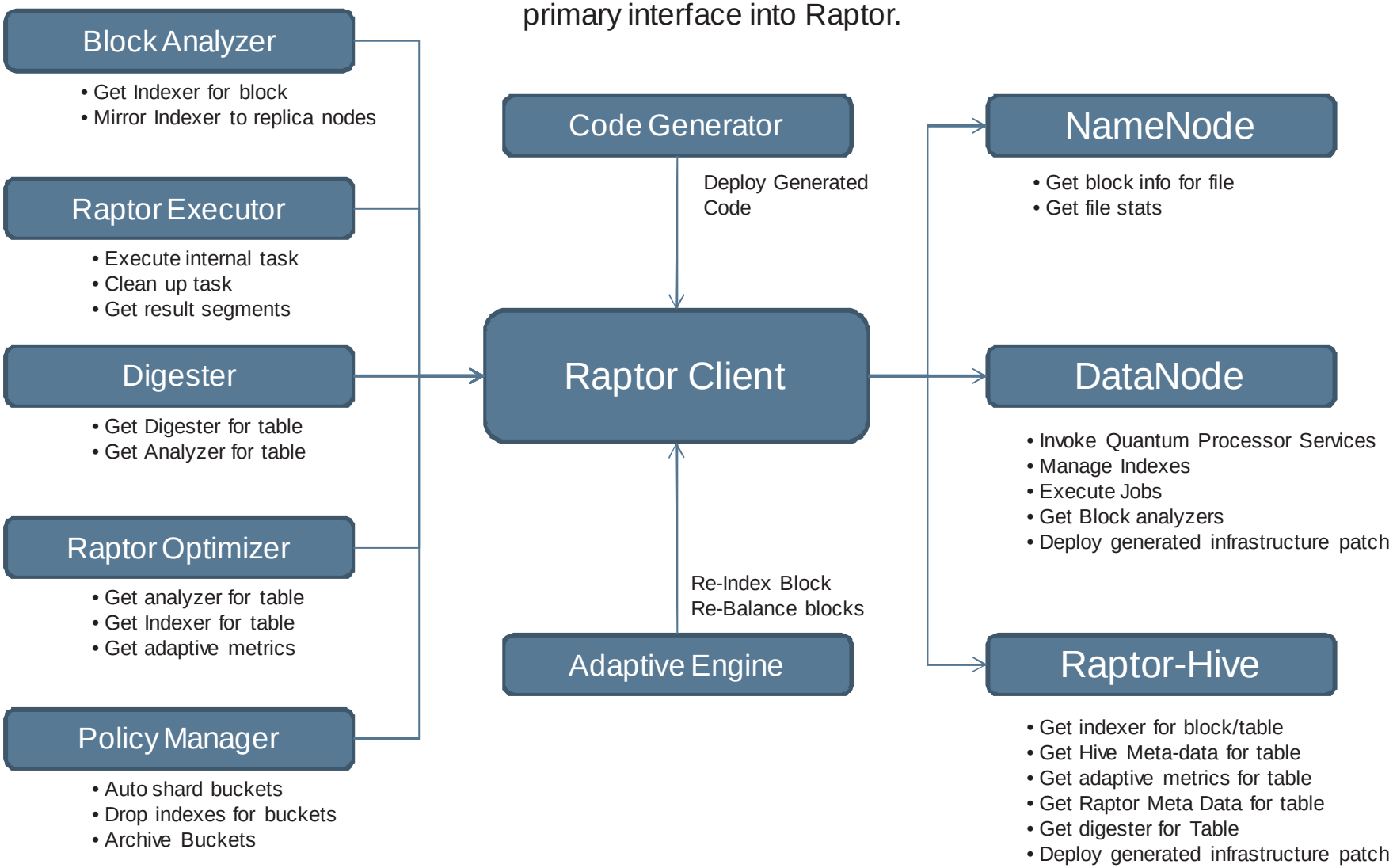
Raptor Block Analyzer – Level 2 Segmentation



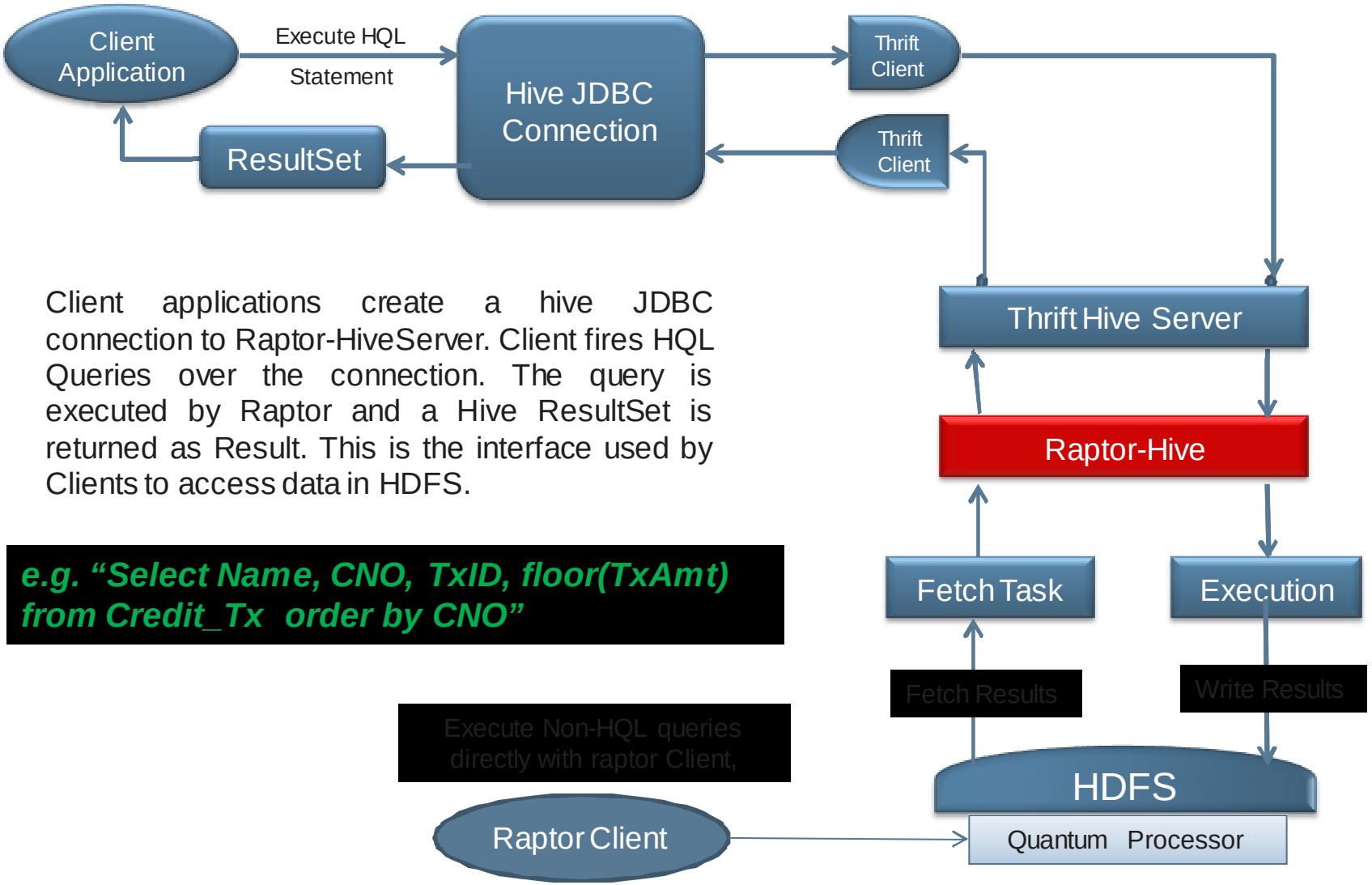
Every block is asynchronously analyzed and the resulting column level blooms, index tree and column level meta information is stored in ThorCache. All this happens within the DataNode Context.

Raptor Client Framework

Raptor Client is similar to dfsclient or jobclient, it stands as a primary interface into Raptor.



Getting Results Out – End to End Flow

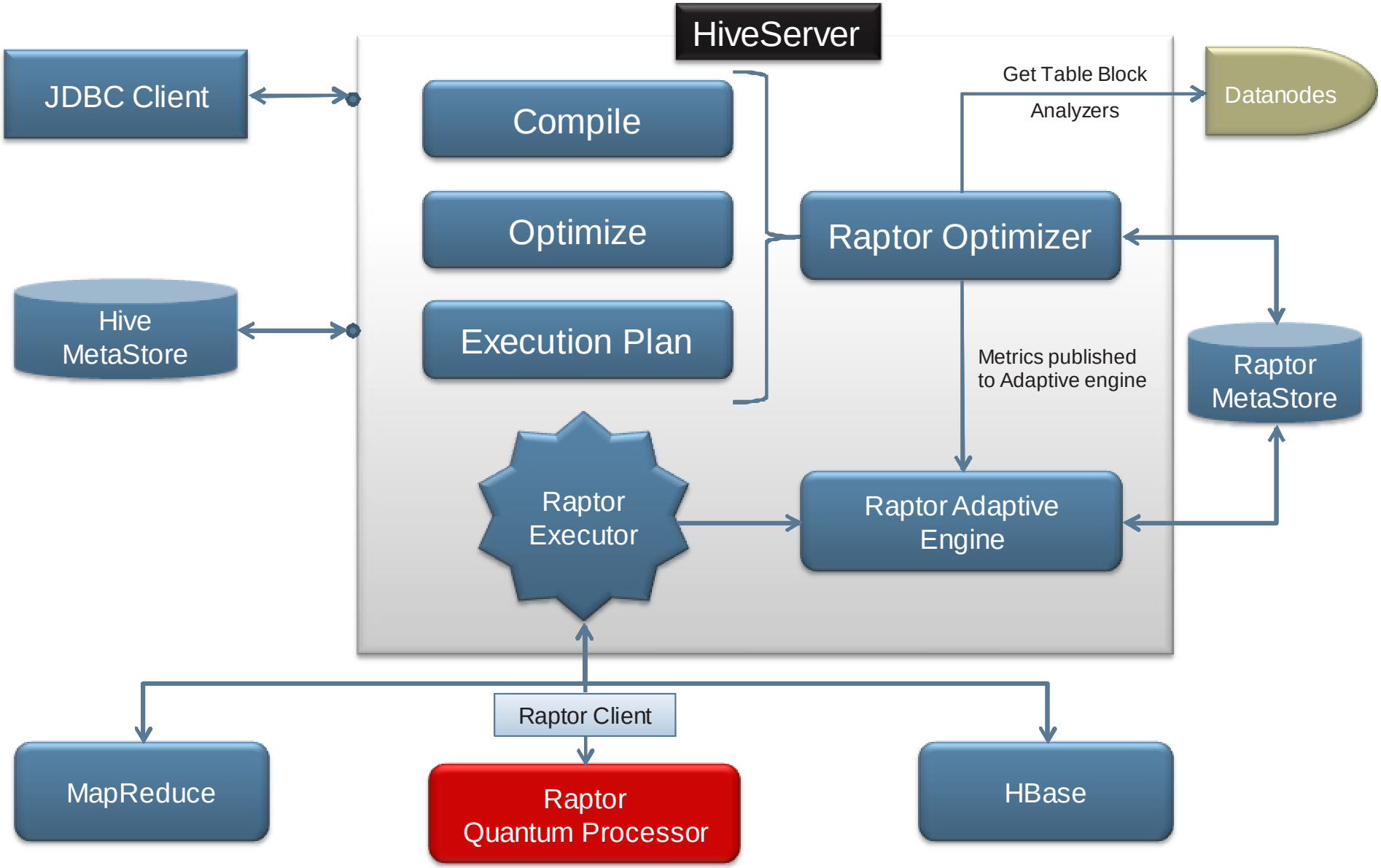


Client applications create a hive JDBC connection to Raptor-HiveServer. Client fires HQL Queries over the connection. The query is executed by Raptor and a Hive ResultSet is returned as Result. This is the interface used by Clients to access data in HDFS.

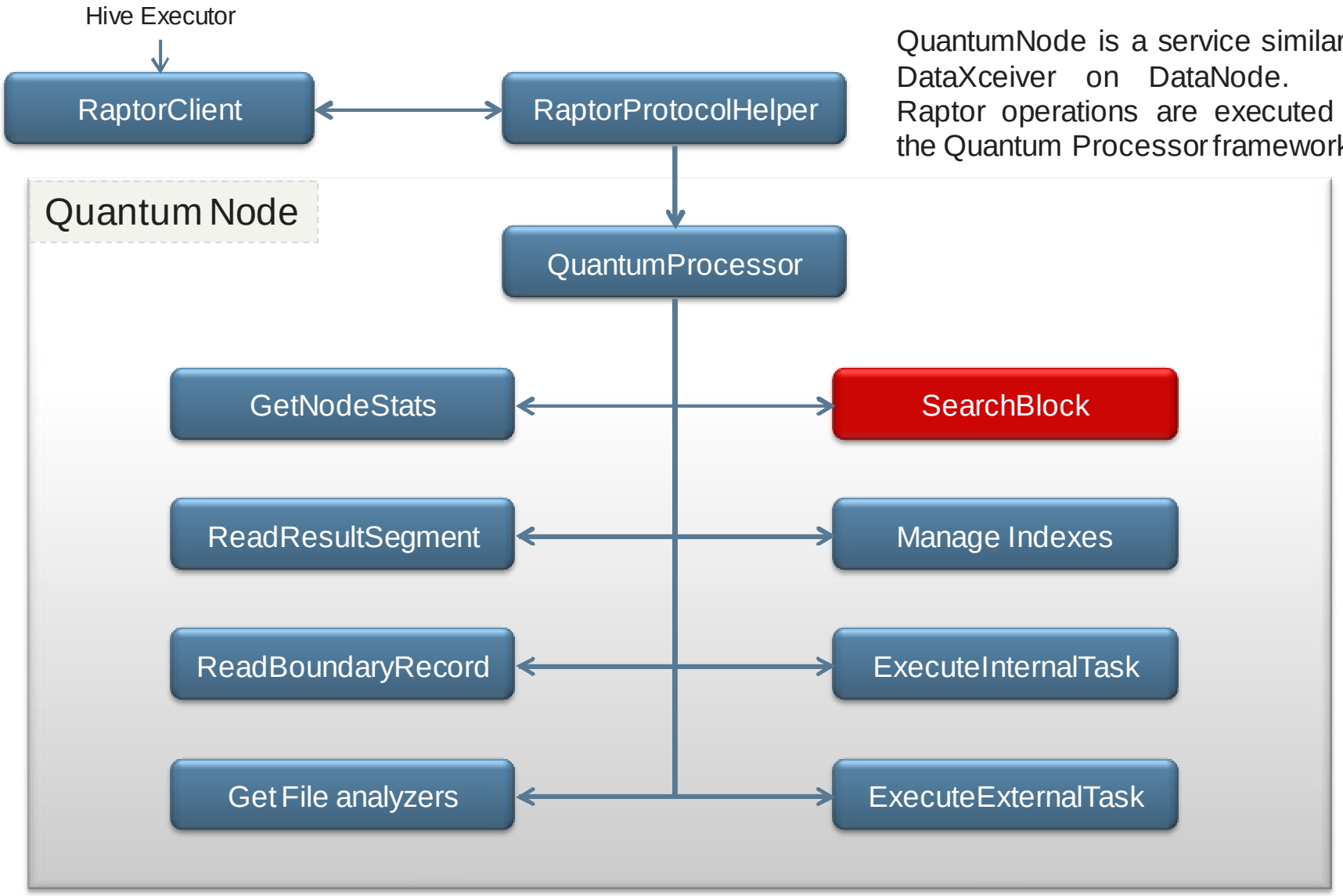
e.g. "Select Name, CNO, TxID, floor(TxAmt) from Credit_Tx order by CNO"

SUNGARD

Raptor-Hive Processing Engine

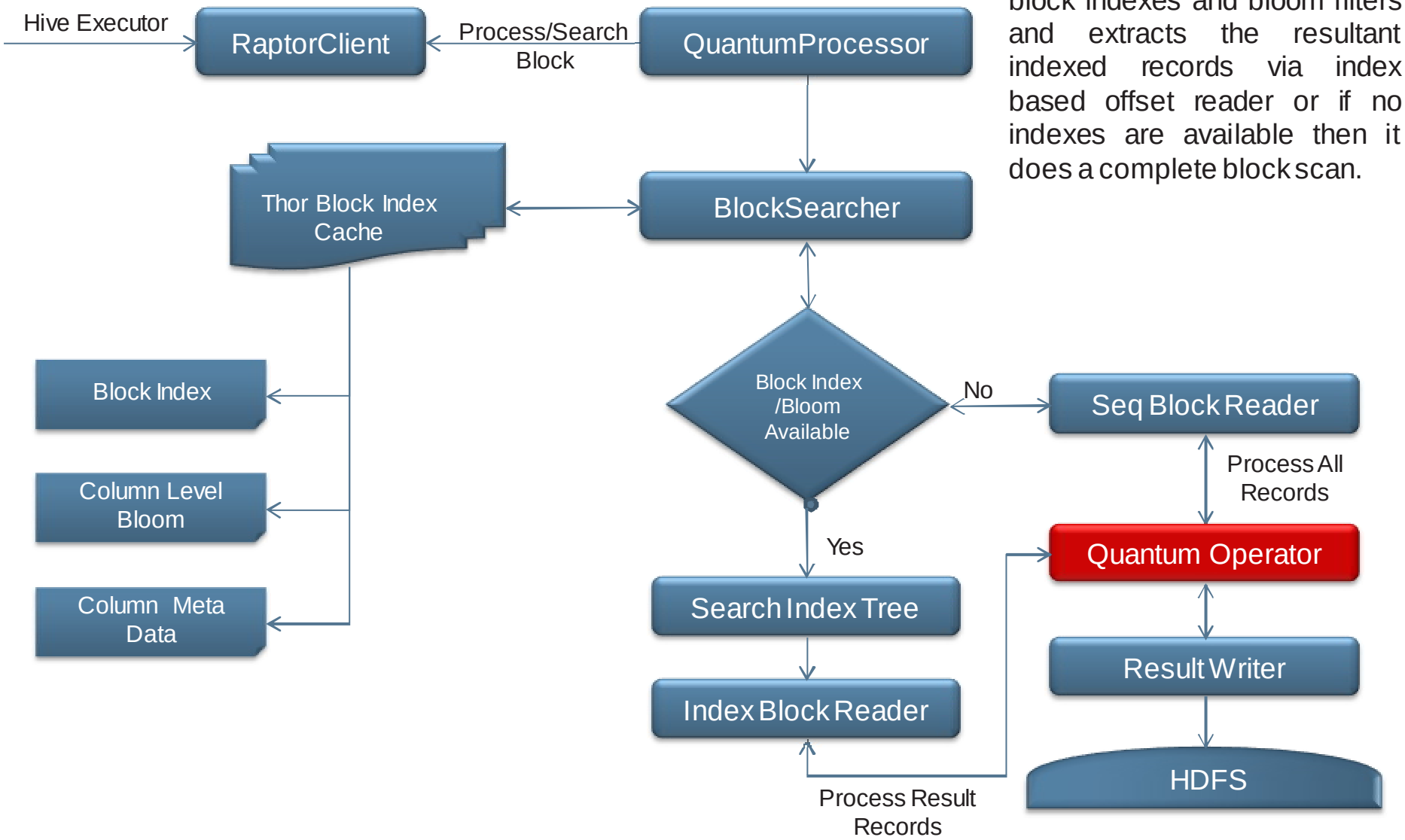


Quantum Processor Framework



QuantumNode is a service similar to DataXceiver on DataNode. All Raptor operations are executed by the Quantum Processor framework.

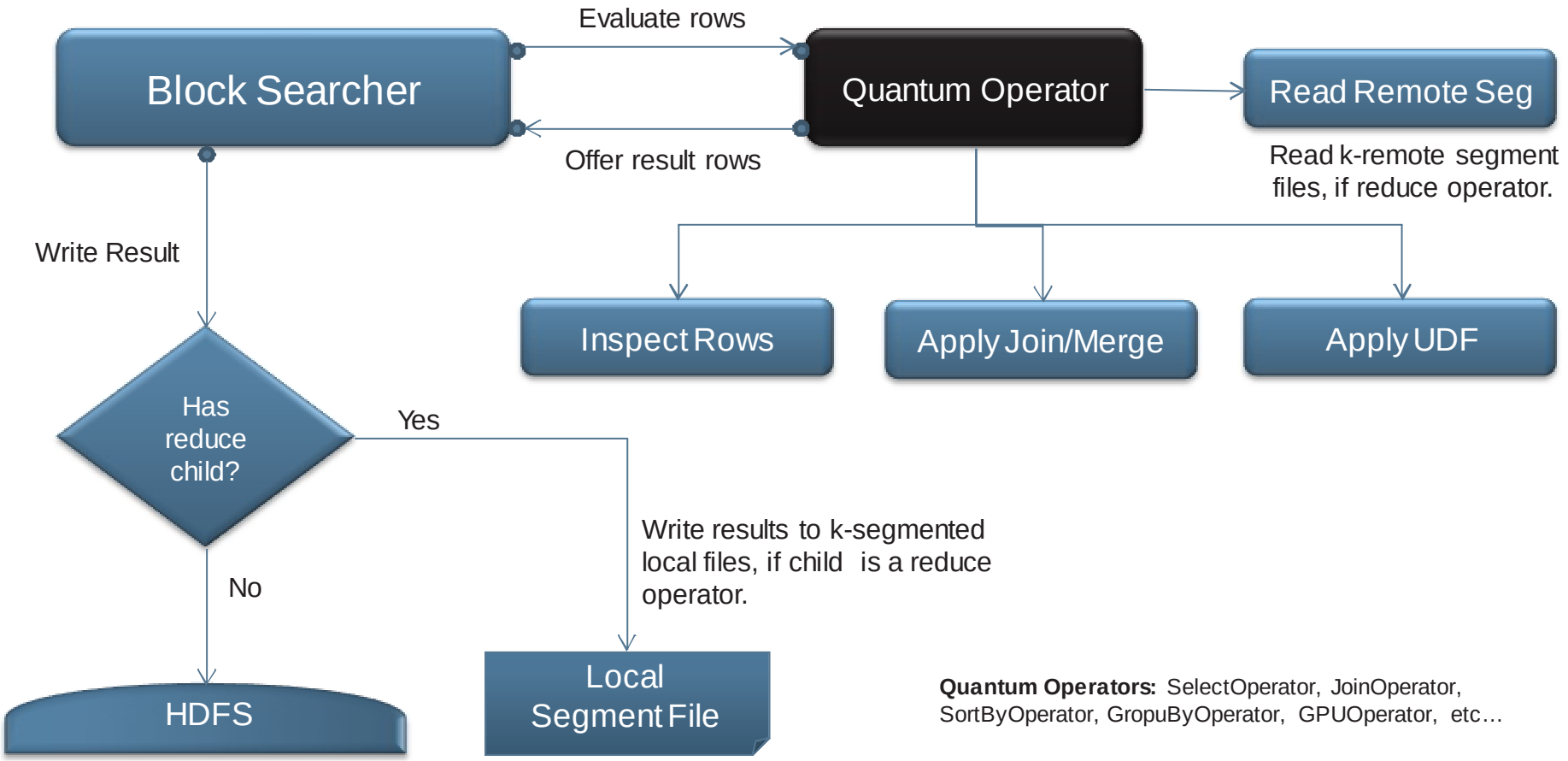
Raptor Block Searcher/Processor



Block Searcher analyzes the block indexes and bloom filters and extracts the resultant indexed records via index based offset reader or if no indexes are available then it does a complete block scan.

Quantum Operator

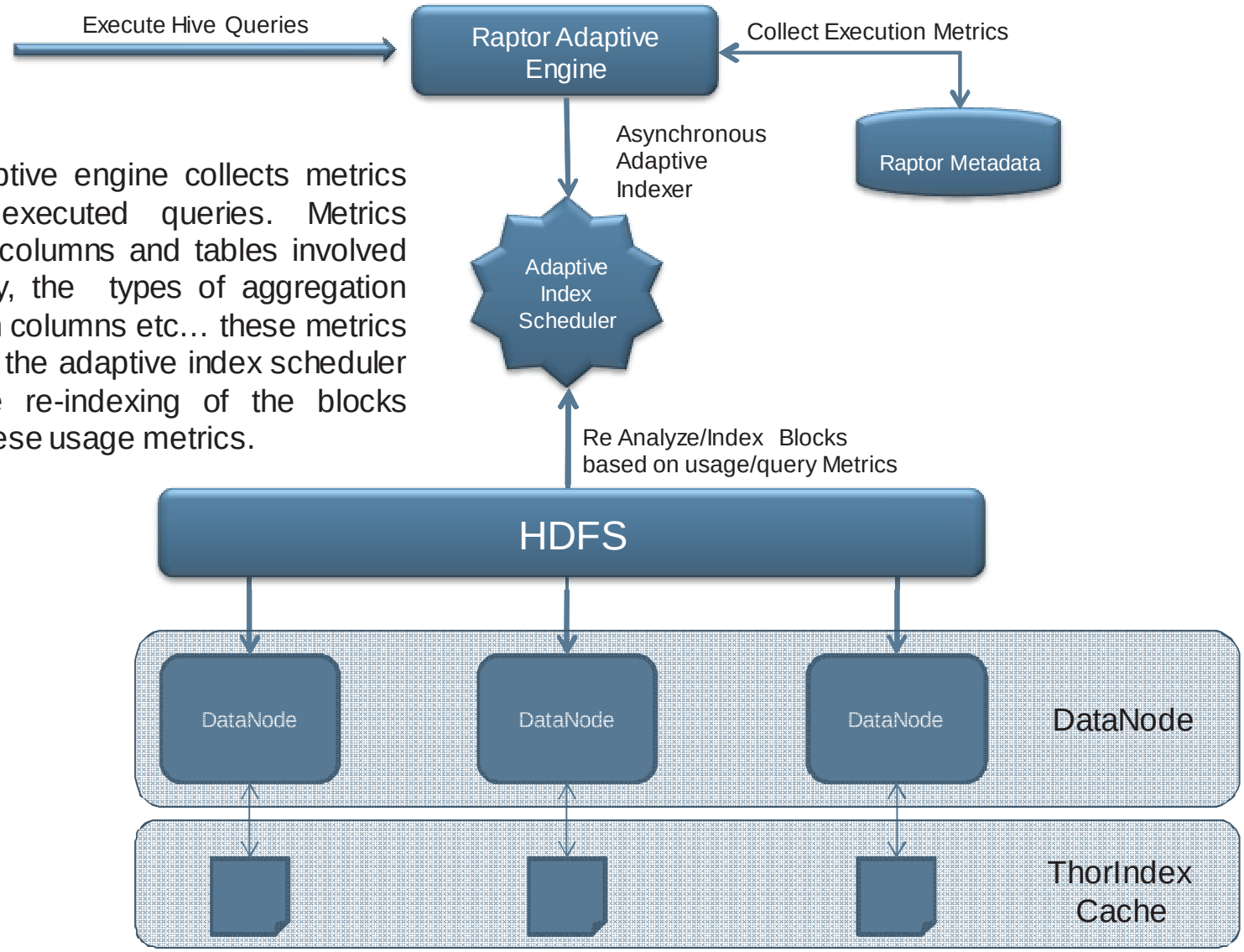
Quantum Operator takes block records as input and performs the required field inspections, and applies UDFs and aggregation. The resultant records are either written to local segment files or to HDFS directly based on the operator type.



Quantum Operators: SelectOperator, JoinOperator, SortByOperator, GropuByOperator, GPUOperator, etc…

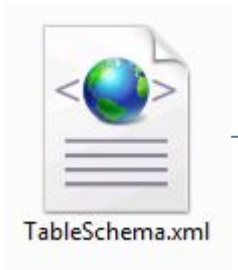
Adaptive Indexing

Raptor adaptive engine collects metrics from the executed queries. Metrics include the columns and tables involved in the query, the types of aggregation executed on columns etc… these metrics are used by the adaptive index scheduler to schedule re-indexing of the blocks based on these usage metrics.

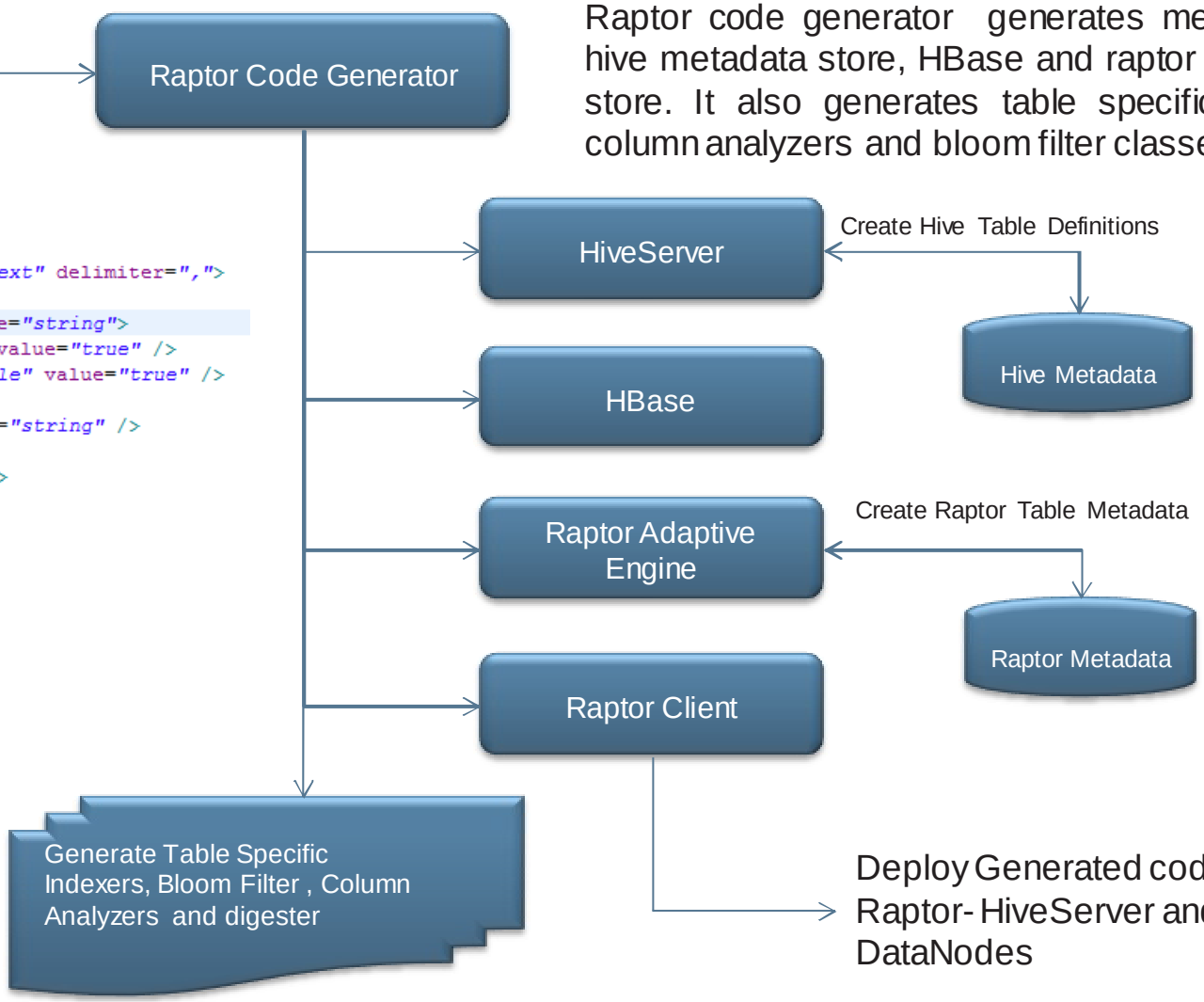


Code Generation and Data Definition Framework

User supplies table schema in an XML format. Based on the Xml table schema and hints, Raptor code generator generates metadata in hive metadata store, HBase and raptor metadata store. It also generates table specific indexer column analyzers and bloom filter classes.

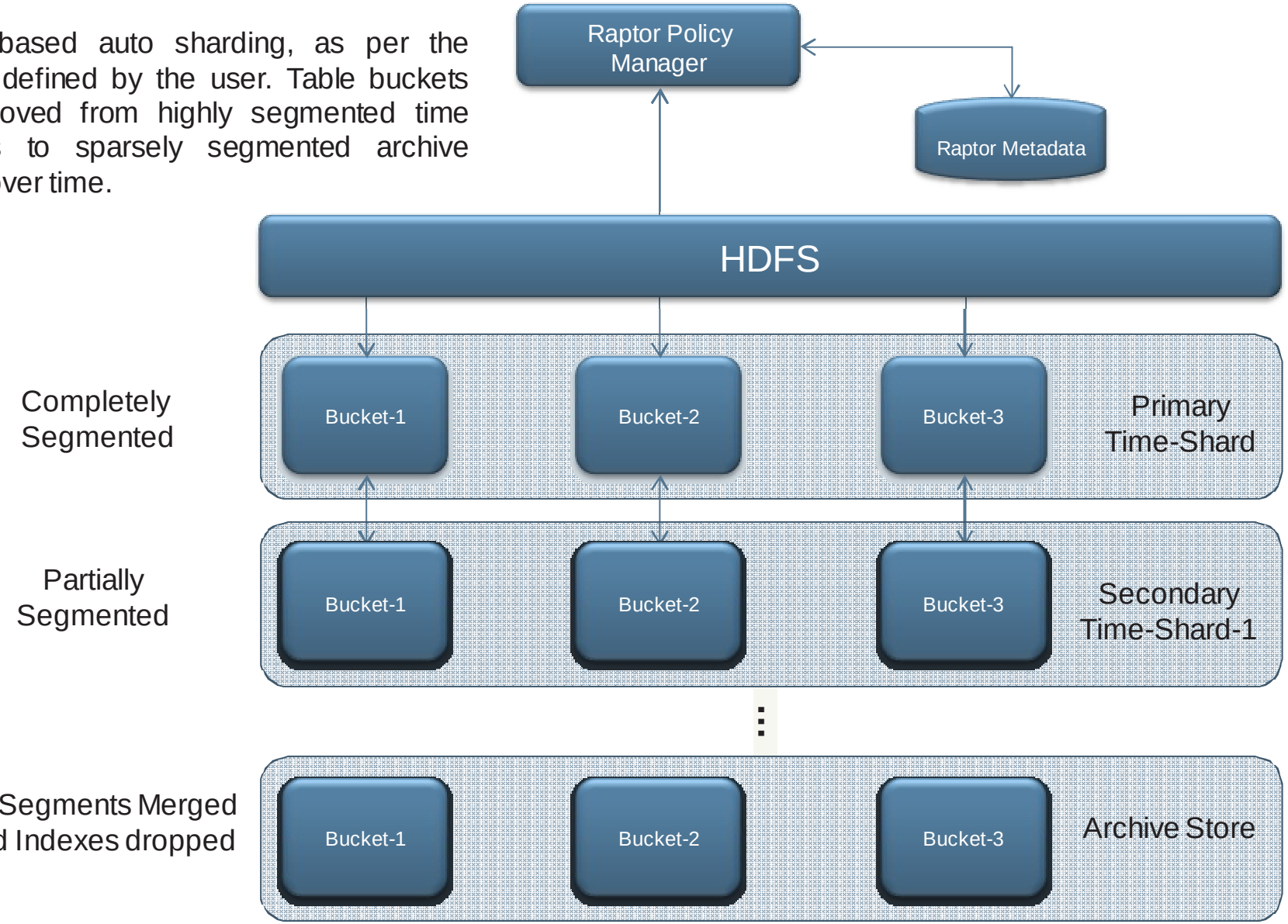



```
<tables>
    <table name="confold" format="text" delimiter=",">
        <columns>
            <column name="Name" type="string">
                <hint name="bloom" value="true" />
                <hint name="queryable" value="true" />
            </column>
            <column name="SSN" type="string" />
        </columns>
        <partitionKey name="Region">
            <key name="UID" />
            <key name="Loc" />
        </partitionKey>
        <index name="index1">
            <key name="Name" />
            <key name="UID" />
            <key name="Loc" />
        </index>
```



Raptor Policy Manager – Time Based Shard

Time based auto sharding, as per the policy defined by the user. Table buckets are moved from highly segmented time shards to sparsely segmented archive store over time.



SUNGARD

Infrastructure Enhancements

- Adaptive compression based on network congestion statistics
- Scheduling of jobs via raptor client (via Hive) in conjunction with enhanced NameNode block policy
- Computation intensive jobs scheduled on GPU enabled nodes
- Object Pools across the Hadoop-Raptor ecosystem
- Hand-shake mechanism between clients and DataNodes to avoid imminent operation failures
- Interactive user console for managing the cluster, tasks, data policy, filters etc…

SUNGARD

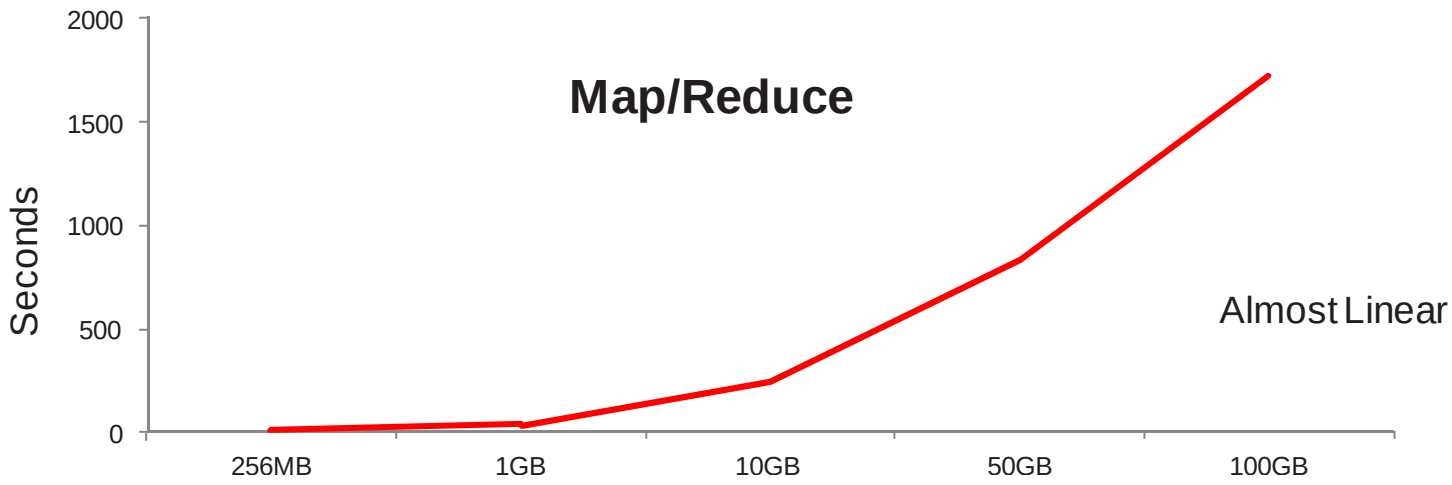
Benchmark Raptor

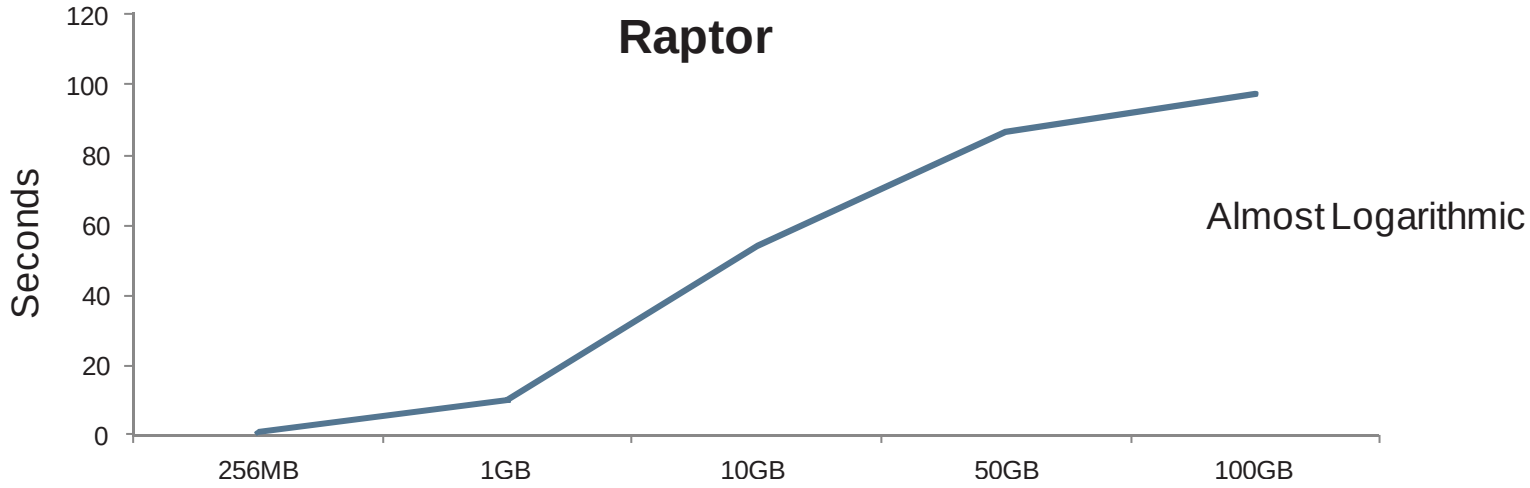
Benchmark carried out on a 5 node cluster with commodity hardware { Core2-Duo, 4GB RAM, 1TB storage, 100Mbps NIC, 64MB block size, Replication factor 3, Network Compression enabled}

Table with 13 columns and mixed types. Sample data generated with node.js with moderate entropy. (table with 1 group index{3 columns} and 0 column bloom filters, 1 bucket}

5 Node Cluster	Raptor			Map/Reduce		
Operation\Table Size	**256MB**	**1GB**	**10GB**	**256MB**	**1GB**	**10GB**
Load Time	4.8s	22s	**4.2m**	54.6s	3.5m	**36.0m**
Simple select without predicate	6.9s	22s	**3.2m**	21.6s	50.5s	**9.5m**
Select with complex Predicate	0.8s	1.7s	**0.9m**	9.2s	32.2s	**4.1m**
Select with Order By	1.6s	6.5s	**1.1m**	41.0s	3.2m	**5.6m**
Select with Group By	0.6s	1.2s	**0.8m**	61.3s	1.5m	**4.0m**
Simple Join	2.9s	5.7s	**1.7m**	1.2m	3.4m	**9.2m**
User Defined row level Function	0.6s	1.1s	**0.9m**	13.2s	49s	**7.3m**

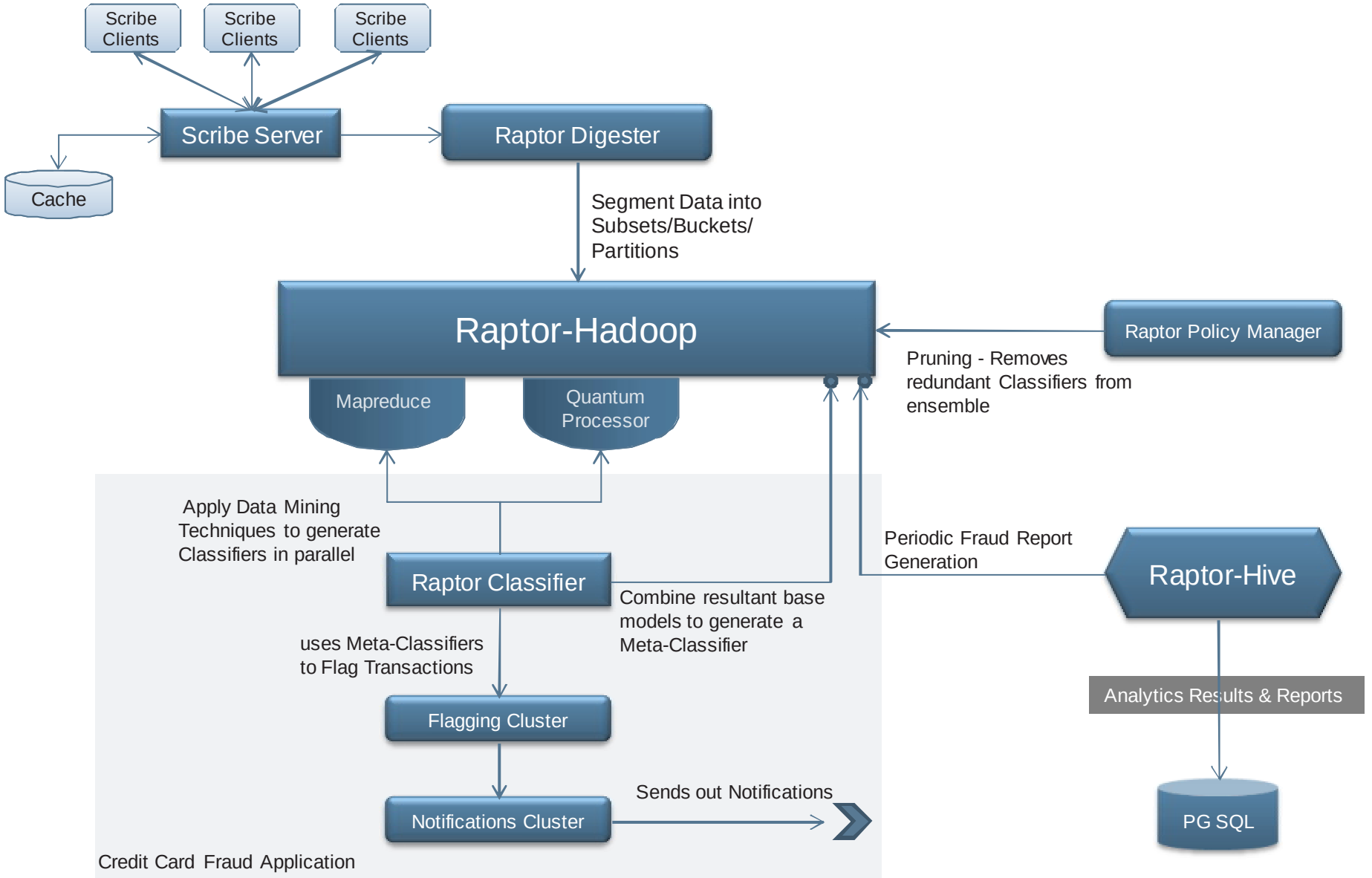
Response-Time Trends





SUNGARD

Case Study: Credit Card Fraud Detection



SUNGARD

Other Raptor Use Cases

- Smart customer care solution
- Financial fraud analytics
- Media usage log analytics
- Computation intensive jobs using GPUs
- Predictive trading

SUNGARD

Roadmap

- Merge Raptor into Next Generation MapReduce
- Dimensions and Cubes (MRCubes)
- Cloud ready Raptor
- Roles and security
- Job failover management
- Rules and triggers
- Planning to open source
- Starter Kits/Examples of various Use Cases

SUNGARD

Benefits

- Query responses at soft real-time windows.
- Code generation framework, for table specific raptor infrastructure code.
- Zero down time, with hot deployment of generated infrastructure code.
- Seamless integration into existing infrastructure with multiple ingress options.
- Automatic time based sharding and data archival.
- Distribute & execute non-MR jobs on the Hadoop Cluster

SUNGARD

Contacts

Soundar Velu

Product Architect

Soundararajan.velu@sungard.com

Twitter: @greyquest

Aditya Yadav

ATS/Research Head

Aditya.Yadav@sungard.com

SUNGARD

References

- Optimizing Distributed Joins with Bloom Filters
  www.l3s.de/web/upload/documents/1/analysis.pdf
- Apache Hadoop Goes Realtime at Facebook
  borthakur.com/ftp/RealtimeHadoopSigmod2011.pdf
- Data Mining with MAPREDUCE: Graph and Tensor Algorithms with MR
  www.ml.cmu.edu/research/dap-papers/tsourakakisdap.pdf
- Distributed Cube Materialization on Holistic Measures*
  www.eecs.umich.edu/~congy/work/icde11a.pdf
- HadoopDB in Action: Building Real World Applications
  www.cs.yale.edu/homes/dna/papers/hadoopdb-demo.pdf
- TeraByte Sort on Apache Hadoop
  sortbenchmark.org/YahooHadoop.pdf
- Mahouth in Action
  www.amazon.com/Mahout-Action-Sean-Owen/dp/1935182684
- Web-Scale K-Means Clustering
  www.eecs.tufts.edu/~dsculley/papers/fastkmeans.pdf
- Hive – A Petabyte Scale Data Warehouse Using Hadoop
  infolab.stanford.edu/~ragho/hive-icde2010.pdf

CONFIDENTIALITY STATEMENT

SUNGARD

Raptor: Real-time Analytics on Hadoop