
14

Case Study: Proposal of Architecture for Big Data Adoption

Nelson Paiva Oliveira Leite, PhD
Luiz Eduardo Guarino de Vasconcelos, PhD Student
André Yoshimi Kusumoto, MsC Student
Cristina Moniz Araújo Lopes, PhD

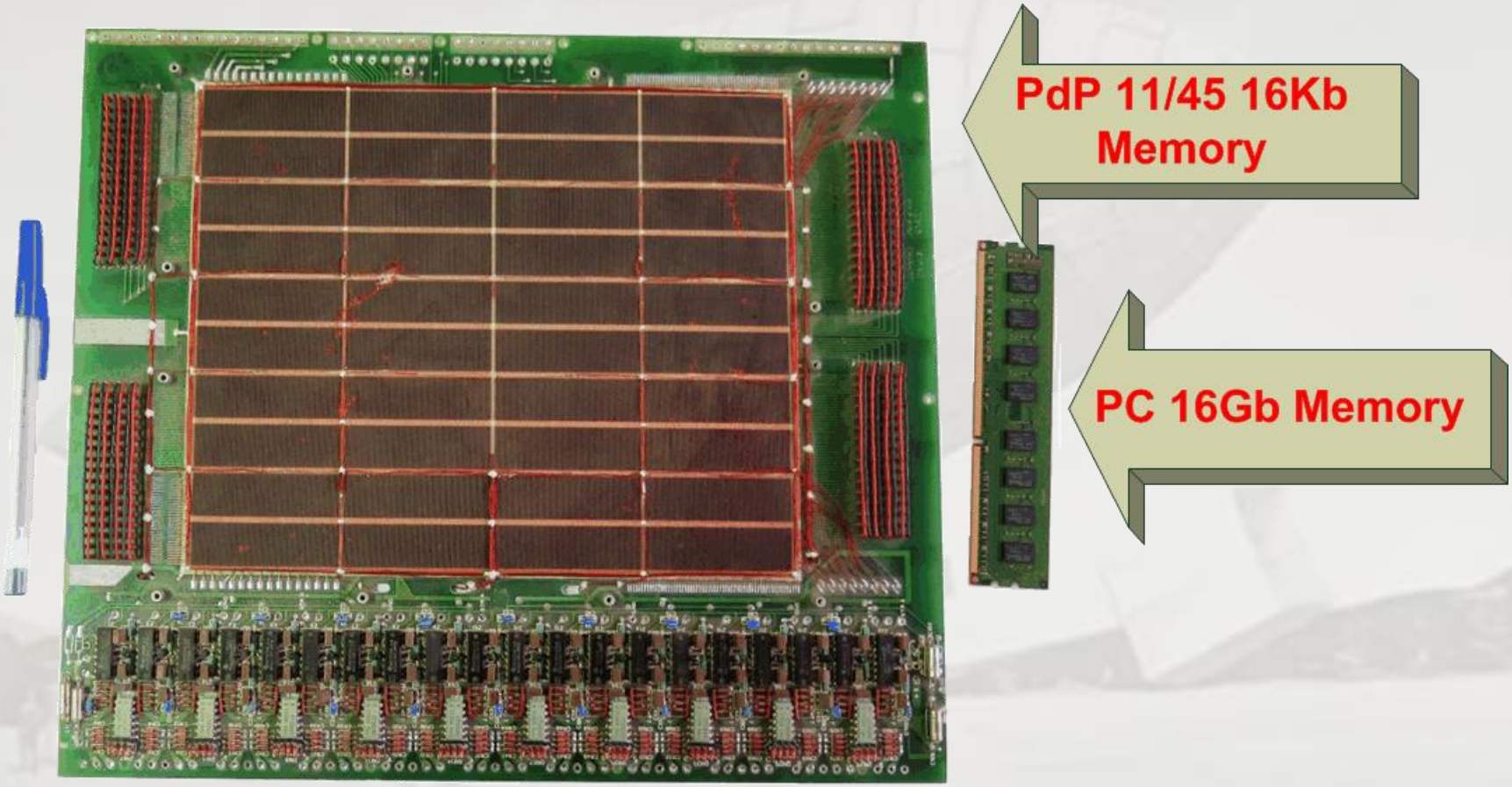
Summary

- Introduction
- Background
- The Proposed Architecture
  - ILM
  - RDMBS
  - NoSQL
  - Integration with Hadoop
  - The Architecture
- Conclusion

Introduction

- ITC 2013 Opening Ceremony
  - John M. Custer MG, Director of Federal Programs, EMC
  - "A New Paradigm: Big Data Changes Everything"
- Big DATA (ref.: Mr. Custer Presentation)
  - You Tube: 1.064 Exabytes;
  - JSF Airframe: 1Tb per sortie;
  - Google Index: 107 Million Gb.
- Flight Test Data Sets can contain gigabytes or terabytes of data, and may grow several megabytes or gigabytes per day

Introduction



Introduction

Removable Disk
512 Kb

Removable Disk
32Gb

Introduction

Main HDD 1.2 Mb

Main HDD 2Tb

Introduction

FTI Primary Storage 120 Mb

FTI Primary Storage 32Gb

Introduction

- **Motivation:**
  - **In Flight Tests, big data sources encompasses:**
    - **Streaming data from Flight Test Instrumentation (FTI) sensors;**
    - **Weather related measurement;**
    - **Video images from multiple cameras;**
    - **Test & Crew personnel voice communications;**
    - **Air safety reports;**
    - **Calibration and uncertainty data; and**
    - **Simulation estimations.**

Introduction

- **Challenges:**
  - **Big data sets may not fit into available memory space;**
  - **It may take too long for processing;**
  - **The stream could be too fast for storage;**
  - **Standard algorithms are usually not designed for big data sets processing in a reasonable amount of time or memory space.**

Introduction

- **Proposition:**
  - **IPEV experimentally defined an architecture for Big Data adoption to analyze the large volume of test flight data collected from different sources;**
  - **The proposed design uses the Apache Hadoop environment tools based on Cloudera Cloud.**

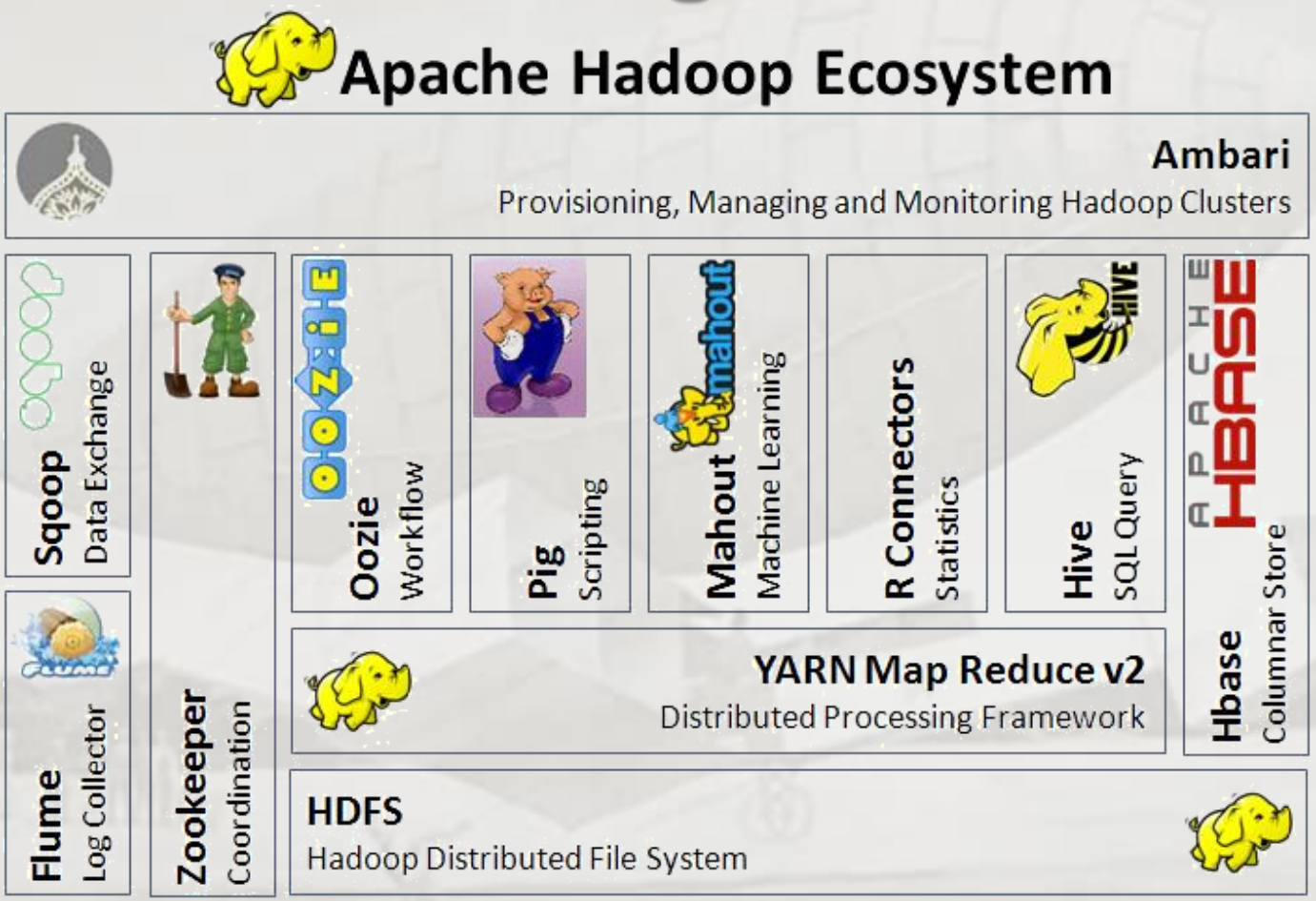
Background

- **Big Data**
  - **Big Data are large pools of data that can be captured, merged, stored, and analysed;**
  - **Data has been growing at exponential rates, but its growing rate is larger than the current evolution of read and write cycle rate for storage disks;**
  - **With legacy technologies, whereas it is required to move all data stored into a one terabyte disk, the resulting process efficiency is very poor.**

Background

- Hadoop
  - Hadoop is a open source software framework which allows distributed processing of large amounts of data sets using cluster computing.

Background



Background

- Hadoop Distributed File System
  - HDFS is a distributed file system for storage and streaming access of large data sets on a machine cluster;
  - File are divided into blocks (64 MB by Default) and distributed across the cluster;
  - In HDFS blocks of file can be replicated (by default 3 times) on other nodes. This scheme provides a fault-tolerant solution so file blocks can be retrieved from other replicas.

Background

- Hadoop Map Reduce
  - It is a distributed data processing model and execution environment that runs on a machine cluster.
  - The input data is loaded into map tasks to be processed in parallel.
  - Its resulting output is input to reduce tasks.
  - With such scheme data is processed in a distributed form on several cluster nodes.
  - The programmer needs to specify the map and reduce functions.

Background

- HIVE
  - Hive is a distributed data warehouse for Hadoop.
  - It was built at facebook based on the requirement for a similar Structure Query Language (SQL) language to work with big data on Hadoop.
  - SQL applications are commonly used by the industry. So the use of a similar language provides an user-friendly environment for programmers and it simplifies the learning process of complex MapReduce programs.
  - Hive provides its own query language known as HiveQL which is also similar to SQL.

Background

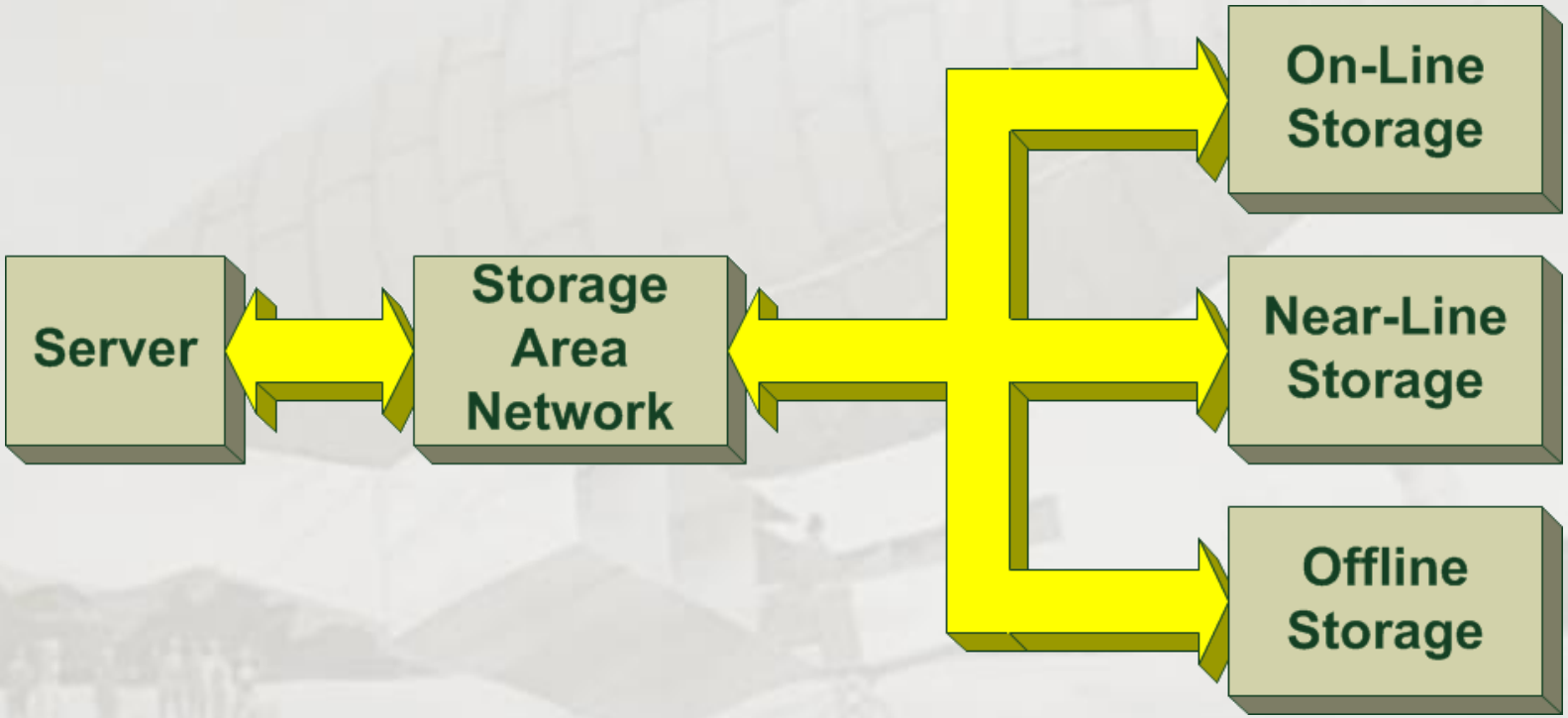
- Cloudera
  - It was also necessary to select the Hadoop distribution kit that supports the required tools and technologies presented within the initial architecture.
  - Cloudera was selected, mostly due to its well-documented site and the possibility for downloading an already pre-configured virtual machine with Hadoop environment and its tools.
  - This feature allowed the execution of several tests in a local environment.

The Proposed Architecture

- Information Lifecycle Management
  - ILM is a lowest cost process used for managing information through its lifecycle, from conception until disposal, in a manner that optimizes storage and access.
  - ILM is not just hardware or software, it includes processes and policies to manage the information.
  - It is designed upon the recognition that different types of information can have different importance at different points in their lifecycle.

The Proposed Architecture

- **Information Lifecycle Management**

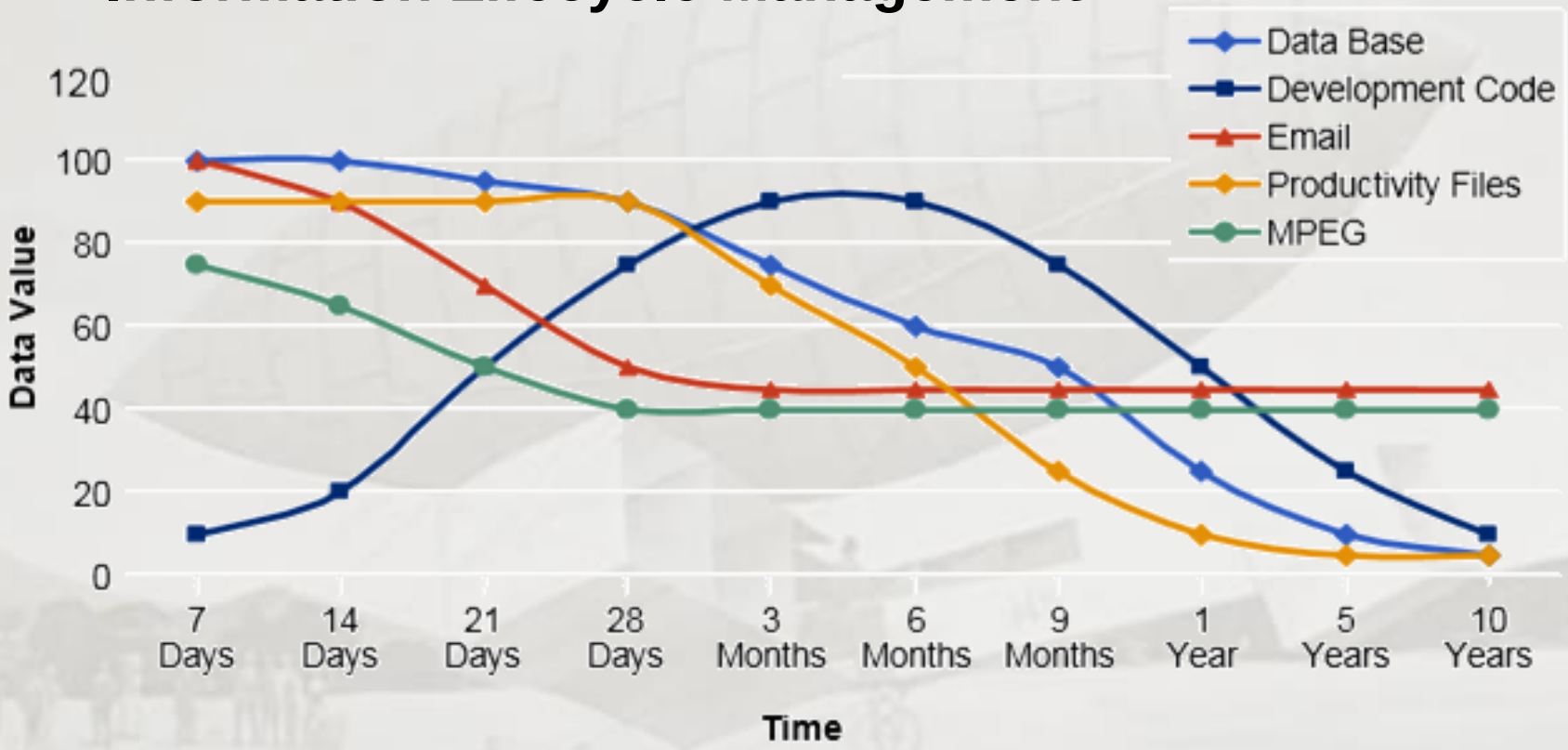


The Proposed Architecture

- Information Lifecycle Management
  - Online storage is best suitable to store frequently-used or fresh data. For IPEV, the online file usage period should not exceed 18 months.
  - Near-Line storage is adequate to store not so fresh and/or not so widely used data. For IPEV, the Near-Line file usage period should be between 18 months and 36 months.
  - Over 36 months, data should be available only in the Offline Storage.

The Proposed Architecture

- **Information Lifecycle Management**



The Proposed Architecture

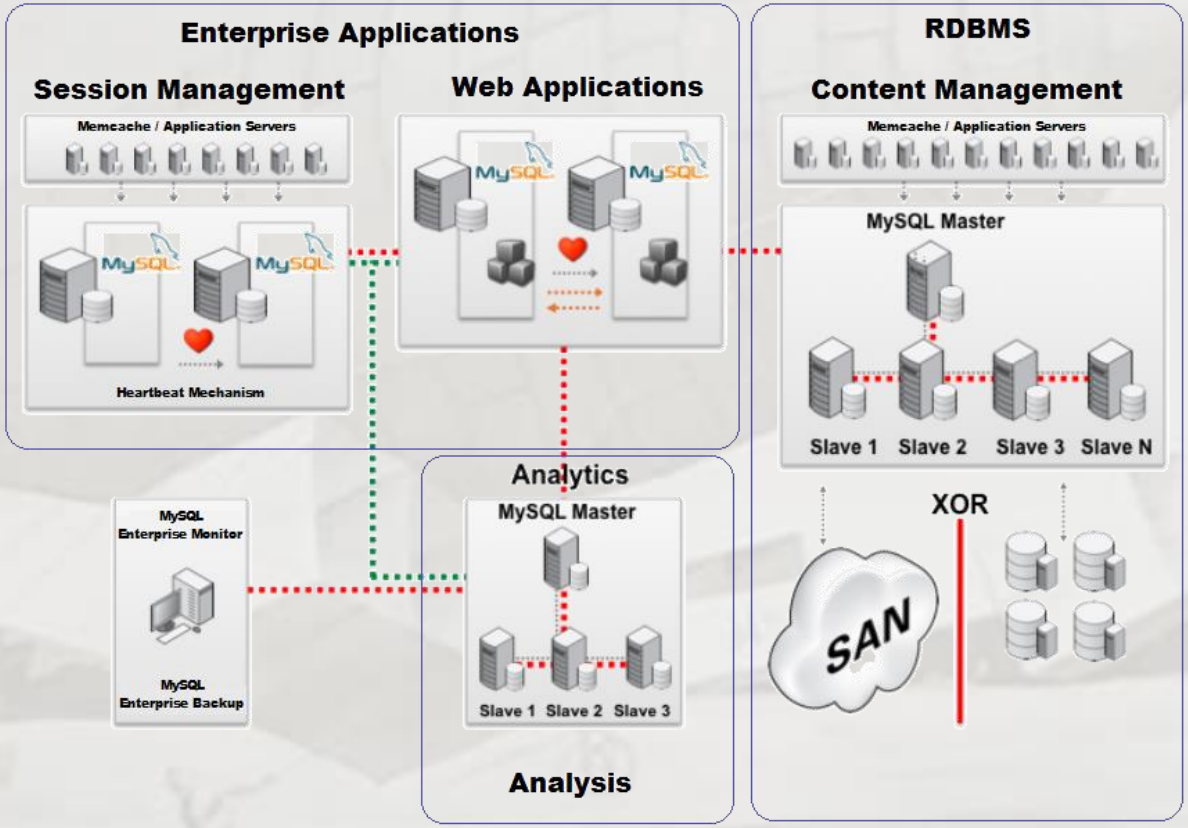
- **Information Lifecycle Management**

The Proposed Architecture

- Relational Database Management System
  - In this application the MySQL is used for storage software as relational database management system (RDBMS).
  - MySQL is deployed in 9 of the top 10 most busiest sites on the web including Facebook, Twitter, eBay and YouTube

The Proposed Architecture

- **Relational Database Management System**



The Proposed Architecture

- **Relational Database Management System**
  - **Enterprise Applications workload uses the default InnoDB storage engine to provide transactional support and crash recovery.**
  - **There are two ways for delivering high availability:**
    - **Use of Linux Heartbeat with semi synchronous MySQL replication; or**
    - **Use of OS-based solutions like Distributed Replicated Block Device (DRBD), along with MySQL Enterprise Backup.**
  - **For data mining and business intelligence applications, session and web data are captured in an analytics database that executes off-line report generation.**

The Proposed Architecture

- **Relational Database Management System**
  - **As a real-time database designed for 99.999% availability.**
  - **MySQL Replication is used to deliver read scalability to each MySQL master that is typically attached to 20 – 30 slaves.**
  - **In a regular content management workload, each slave should be able to support up to 3,000 concurrent applications. Such performance exceeds a typical test range requirement such as IPEV's.**
  - **For physical storage, the content assets can be stored either on a Storage Area Network (SAN).**

The Proposed Architecture

- NoSQL
  - MySQL shall not be used for everything.
  - Apache Cassandra is an open source system designed to manage large volumes of data in real-time, enabling immediate response and providing a fail-safe solution.
  - The Cassandra system acts as a distributed database and it could be used with non-relational database, such as NoSQL (not only SQL).

The Proposed Architecture

- NoSQL
  - Cassandra is used for high rate writes, and low rate reads.
  - The main advantages are:
    - The fact that Cassandra can run on cheaper hardware than MySQL,
    - Simple expandability; and
    - Its schema less design.
  - In this particular application, all flight test data, such as images, videos, tests reports, and post-mission results are stored in Cassandra.

The Proposed Architecture

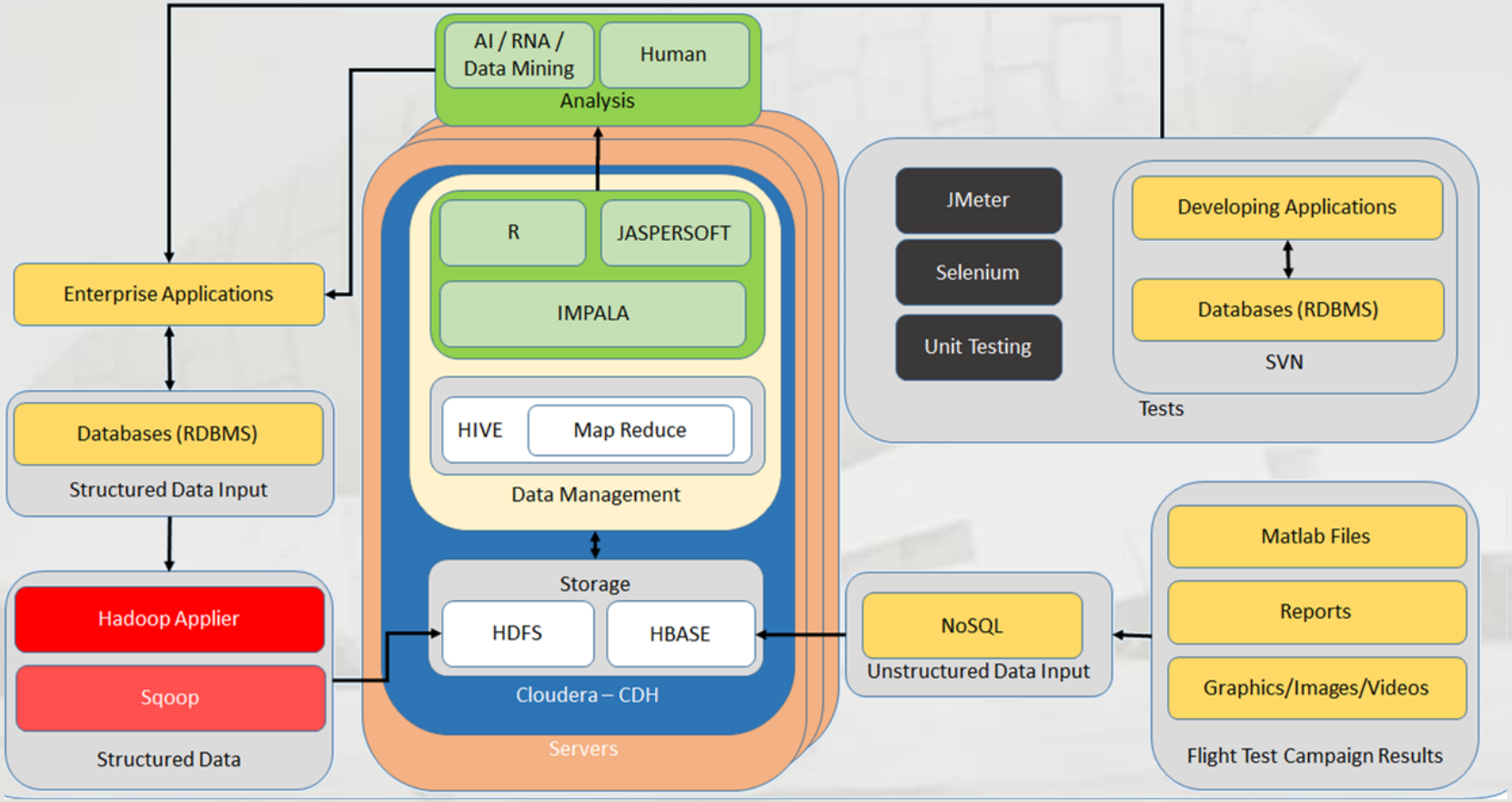
- **Integration RDBMS, NoSQL and Hadoop**
  - **The challenge was explore the possibility of automatic data replication, in real-time, from relational architecture (MySQL) to the unstructured data, represented by HDFS.**
  - **This challenge was addressed with MySQL Applier for Hadoop (Happlier).**
  - **The tool reads MySQL binary log through its Binary Log Application Program Interface (API) and creates corresponding entries for databases and tables as Comma Separated Values (CSV) files in HDFS.**
  - **This process is transparent to the user application.**

The Proposed Architecture

- **Integration RDBMS, NoSQL and Hadoop**
  - **Considering NoSQL, the Cassandra supports executions of Hadoop MapReduce tasks.**
  - **The tasks of MapReduce can search data inside of Cassandra and return data to Cassandra or in file system.**
  - **Cassandra's Hadoop implements the same interface as HDFS to achieve input data locality.**

The Architecture

Big Data Architecture - Flight Test Research Institute (IPEV)



The Architecture

- All enterprise applications store data in MySQL.
- The transactions were first maintained on a relational database (e.g. MySQL) and then instantly replicated to the HDFS in CSV format via MySQL Hadoop Applier (i.e. Happlier).
- For further analysis, the HIVE was used and the CSV is remapped as databases and tables equivalent to the relational model. This allows data analysis using Hadoop.

The Architecture

- The Enterprise Applications are developed under Java with Spring Framework or Microsoft ASP.NET in C#.
- In case of Java, Maven is used to manage the application lifecycle and to control its dependencies.
- All applications use a persistent framework for data storage (e.g. Hibernate or Entity Frameworks).
- The Model View Controller (MVC) architecture and RESTful API are also used to allow an easy integration process among all applications

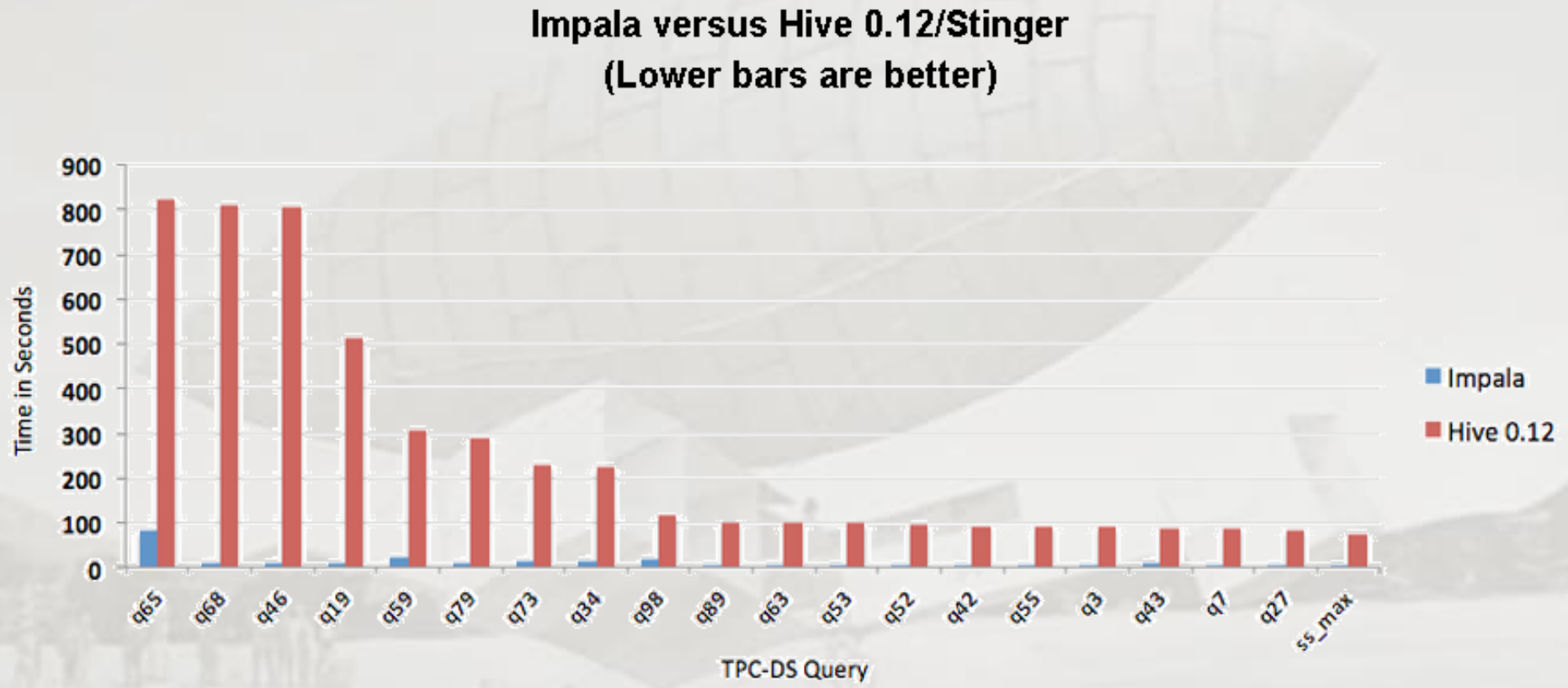
The Architecture

- During the flight test campaigns, many applications are developed (e.g. Matlab Script Applications), reports are produced and several images and videos are generated.
- Such information and related files are unstructured and thus stored in Cassandra, in NoSQL environment.
- After that, the data is sent to the Hadoop.
- With all information in bigdata environment, the next step is to do data analysis.
- For this, the application also allows the execution of user-transparent data analysis in the big data environment.

The Architecture

- For statistical analysis, the R tool is a good option; however, it works as standalone tool.
- The intention was to add the analysis with R scripts within the web application (Java/C#).
- To make this possible, the web application executes Scripts R accessing data via IMPALA application. Then it presents graphical data and outlines to the Technical Staff.
- Impala enables access to data in HDFS with better performance than Hive. The improvement in performance occurs because Impala doesn't requires the execution of a job map reduce operation through Hive queries.

The Architecture



Summary

- This paper proposes the architecture for big data adoption in Flight Test business. It also describes various resources that should be used in the project.
- The proposed solution provides a robust and scalable architecture to be used by IPEV for software development and information storage.
- Moreover, it is based on the use of open source tools that is already proved an efficient solution for large-scale systems.

Summary

- The main benefit of the architecture is the gain of competitive advantage.
- The big data architecture allows the combination of different data types for the execution of a more comprehensive Flight Test data analysis. Such system can be operated and managed at scale.

Summary

- The proposed architecture deals with different technologies.

- Future works should:
  - Verify the integration of Pig and Jaql technologies to the scope of the project;
  - Explore the use of specific tools that automate testing for the Big Data environment (e.g MRunit for Map Reduce Tests).
  - Explore and verify new tools that could improve the application safety

Acknowledgement

- We wish to thank the unconditional support given by the Instituto de Pesquisas e Ensaios em Voo (IPEV) and Instituto Tecnológico de Aeronáutica (ITA).
- Also we like to thank FINEP under agreement 01.12.0518.00 that funded the development of this proposed architecture and the presentation trip