informatica

White Paper

Big Data and HParser

Game-Changing Innovation for Telecommunications Operators

This edition published May 2013

Table of Contents

Introduction

Big data opens new vistas of opportunity for the telecommunications industry—but only for telco operators able to leverage it. HParser is a simple Hadoop integration tool to help navigate this transition and transform crushing volumes of IP data into business value. Using Hadoop and HParser, telcos can shorten project timelines from months to days, shrink budgets from millions to thousands, replace costly outsourcers and consultants, and allow IT staff to be more productive.

This white paper describes the game-changing role of HParser in the telco industry and provides a real-world project example—efficient low-cost processing of call detail records (CDRs). The prescriptive example describes system design, work estimates, capacity planning, and performance benchmarks. It concludes by describing a quick, easy approach to launching big data projects.

Big Data in Telcos Today

Big data offers telco operators multiple opportunities for significant competitive advantage. As a result, the large operators are experimenting with big data initiatives in all of these areas and more:

- Customer experience management – By collecting data across all services and touch points, telcos can finally achieve the elusive goal of creating a holistic historical record of each customer's individual experience. These records in turn provide the foundation for more effective customer retention, resource allocation, and strategic planning (see Figure 1)

Figure 1. Transition to Customer Experience Management (CEM) in telecommunications

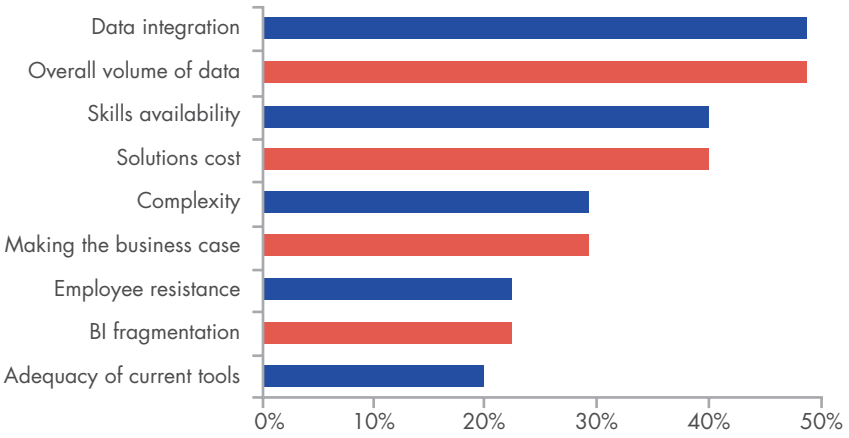
- Churn management – With vast amounts of unstructured customer interaction information, telco operators can perform more refined segmentation and build better predictive models. This greater insight into customer lifecycle patterns enables more effective initiatives to reduce churn.
- Telco-assisted services and marketing – The huge volumes of customer data that telcos can capture from diverse sources create a treasure trove. Transforming this data into actionable event-based information and services is mutually beneficial for both telcos and their customers.
- Infrastructure improvement analytics – Big data allows telco operators to no longer base infrastructure improvements on assumptions—more populated areas need more equipment, older equipment should be upgraded first—in favor of smarter, more precise investments that respond to actual business demands. Analytics supply the basis of a model that includes customer experience, service usage, travel patterns, revenue trends, social clusters, and other demographic and usage data.

- Sale of anonymized data for market research – The telco industry has an unequalled amount of information about where consumers are, where they go, and what services they use. Big data allows telcos to turn this information into a valuable commodity: aggregated, anonymous, highly segmentable data sets that help businesses and market research firms pinpoint their target markets without compromising customer privacy.

Technology Barriers

The potential of big data makes its widespread adoption throughout the telco industry inevitable. However, operators face multiple technical, organizational, and regulatory challenges as they evolve toward broader adoption of big data systems. Figure 2 presents some of these obstacles, both real and perceived; they include the following:

- Big data learning curve
- Lack of skills and tools
- Capacity/performance estimations
- Lack of definable ROI
- Complex data transformations
- Fragmented systems and data sources



Source: TM Forum, 2012

Figure 2. Challenges for big data success

HParser Breakthrough

HParser is designed to help users overcome these barriers to big data adoption by speeding and simplifying the steps to a production-ready Hadoop application. As long as HParser users know Hadoop basics, they can avoid laborious Hadoop development.

HParser helps bridge the Hadoop skill gap, enabling Informatica® PowerCenter® developers and other users become productive with big data solutions.

HParser Overview

HParser architecture consists of HParser Studio for design activities and HParser Run-time for execution of transformations within the Hadoop MapReduce framework. As Figure 3 illustrates, once data transformations have been defined in Studio's Integrated Design Environment, they can be easily invoked within Hadoop MapReduce framework on a cluster of computers.

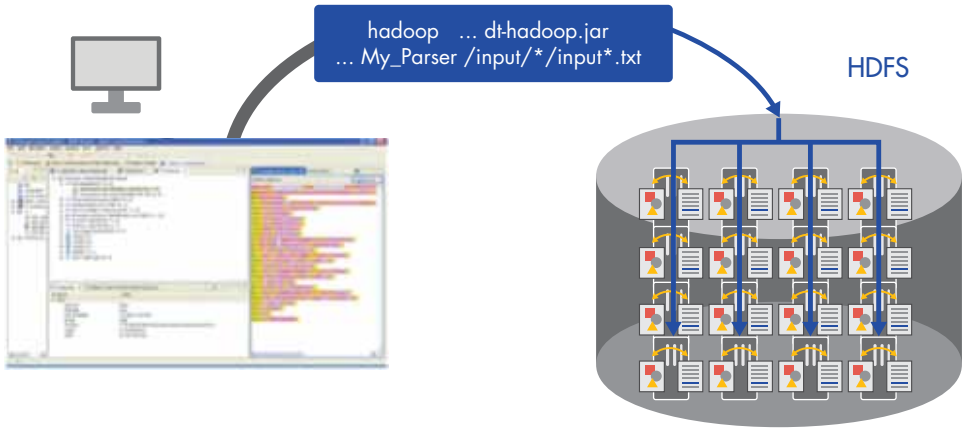


Figure 3. MapReduce paralalizes transformation processing

HParser Studio can develop data transformations from scratch or bootstrap transformation development using various forms of metadata or grammar. For example, HParser Studio understands ASN.1 grammar, commonly used in the teleco industry to describe information collected from network switches, and can automatically build data transformations from it.

Execution Patterns

HParser run-time can be configured for multiple processing patterns:

Bulk Processing Pattern

For bulk processing, the Mapper task reads each file off the Hadoop file system (HDFS) and immediately uses HParser to transform it as a single block of data. The Reducer task receives the results and writes them to the HDFS as an output file (see Figure 4).

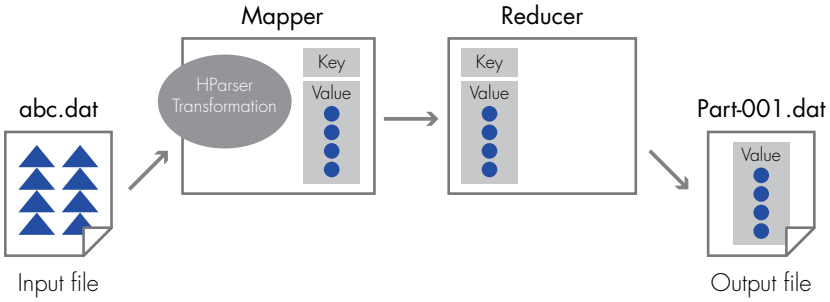


Figure 4. HParser bulk processing pattern

This is the preferred configuration for files that are relatively large but smaller than a single HDFS block.

Line Reading Pattern

In this configuration, HParser uses the native Mapper line reader to process each record individually as the Mapper task reads it. The Reducer task receives the transformation results as a set of key-value pairs, aggregates all values, then writes them to an output file (see Figure 5).

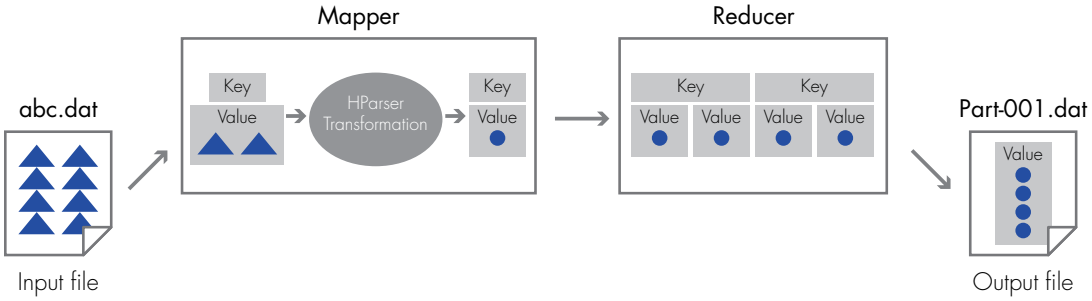


Figure 5. HParser line reading pattern

This configuration is ideal for very large files spanning multiple HDFS blocks.

Reducer Pattern

This configuration pattern invokes HParser in the Reducer task. The Mapper reads input records and passes them to the Reducer as key-value pairs, which the Map Reduce framework organizes by key to generate all values for a given key. Reducer then applies HParser transformation to all values under the key before creating the output file (see Figure 6).

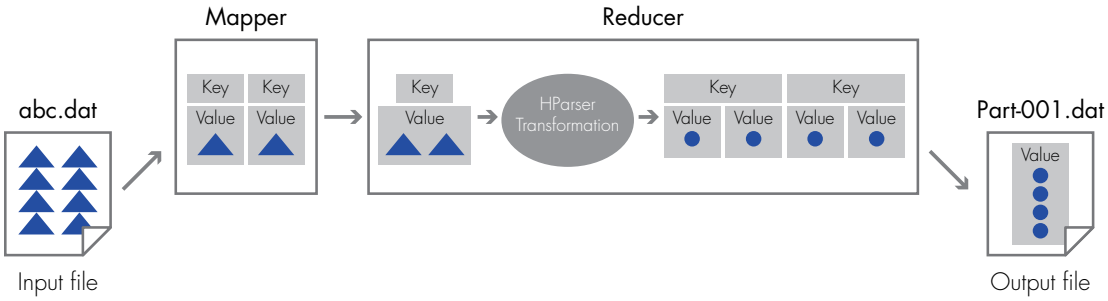


Figure 6. HParser reducer pattern

This configuration is best for data that needs to be reorganized before transformation by HParser.

Design Example: CDR Call Graph

Call detail records (CDRs) contain a wealth of data useful for churn analysis, targeted marketing, customer experience, and many other purposes. Organizing CDRs into graphs makes this data more easily accessible for customer segmentation, identification of social clusters, and historical and location views of customer behavior.

To capture this valuable historical perspective, however, the system needs to process many terabytes of data daily for years. The resulting database would eventually grow to multiple petabytes.

In order to build and continuously update this vast database, the user would be using Hadoop HDFS as a temporary storage system to collect and process CDR files before loading them to a database like HBase (see Figure 7). HBase is the system of record from which various analytical tools perform call graph analysis.

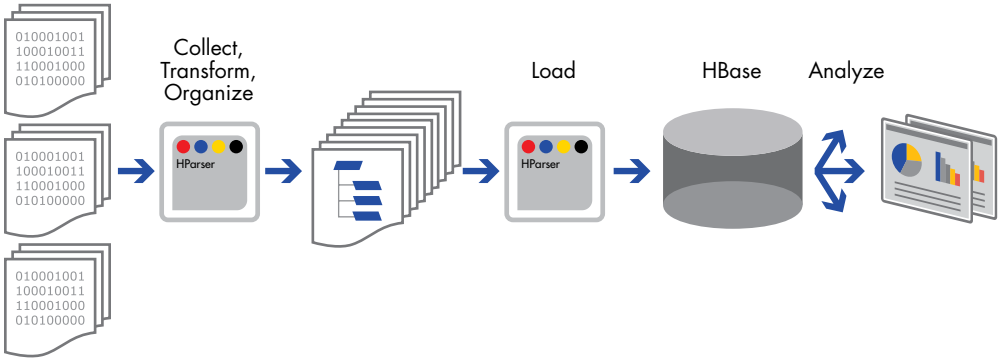


Figure 7. Call graph system data flow

Step 1 – Data Sources

The initial step transforms encoded binary CDRs into a textual representation (see Figure 8).

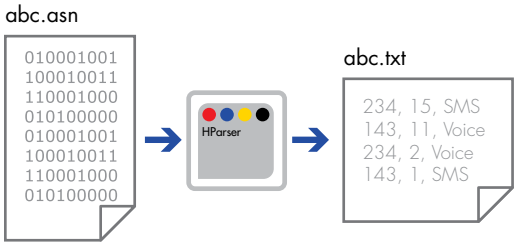


Figure 8. Converting binary data to CSV

In this example, we assume that an ASN.1 grammar file describes the binary CDR format. HParser Studio automatically generates a CDR parser that converts binary data representation into an intermediate XML format. The only manual work necessary is to extract from XML into CSV (Comma Separated Values) output the important elements that the transformation needs to produce.

Once completed and tested in HParser Studio, the transformation definition moves into a Hadoop cluster. Because CDRs generally come in files of 10 to 50 MB—a size that does not require archiving and does not exceed an HDFS block—HParser configuration is straightforward in this case: the bulk processing pattern. With this configuration, HParser reads binary CDR files from HDFS and creates equivalent files containing textual representation of the calling records.

Step 2 – Conversion to Canonical Format

As Figure 9 illustrates, the next step begins by sorting the records on a subscriber key, then transforming them into a hierarchical format. This example uses XML; a more compact file representation would probably use the JSON, Protobuf or Avro format.

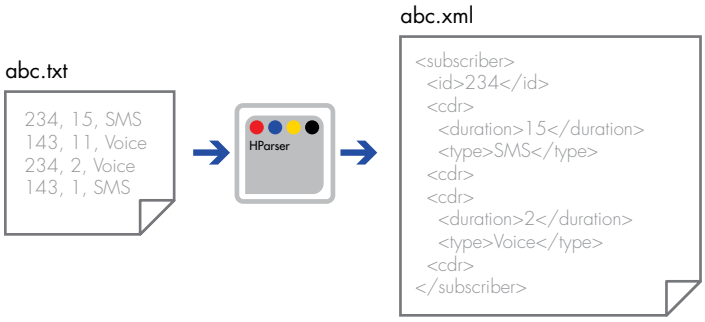


Figure 9. Converting CSV into Canonical Form

This step has two parts:

1. Design a CSV to XML transformation in HParser Studio. This step is simple, assuming that the CSV records will be sorted.
2. Move the transformation definition into the Hadoop cluster and configure HParser run-time. This example uses the Reducer pattern, passing call records through the Mapper Task. The records are sorted between the Mapper and the Reducer. Therefore, the Reducer would accumulate all records for each subscriber key. At the point of completion, HParser transformation would convert all call records for each subscriber key into an XML fragment

Step 3 – Load

As in previous steps, the first task is to design and test the transformation in HParser Studio. HParser represents the HBase database structure as XML Schema Definition (XSD). Therefore, the last transformation in this system is XML to XML.

Once the transformation is ready, it is also moved into the Hadoop cluster (see Figure 10). Depending on input XML file size, we can use one of several HParser configuration patterns. XML files smaller than 100 MB can use the bulk processing pattern. Alternative approach is generating very large XML files that utilize HDFS storage more efficiently. HParser would be configured for the line reading processing pattern, using XML tags as delimiters to find logical data boundaries.

The last step is to load hierarchical call records into HBase.

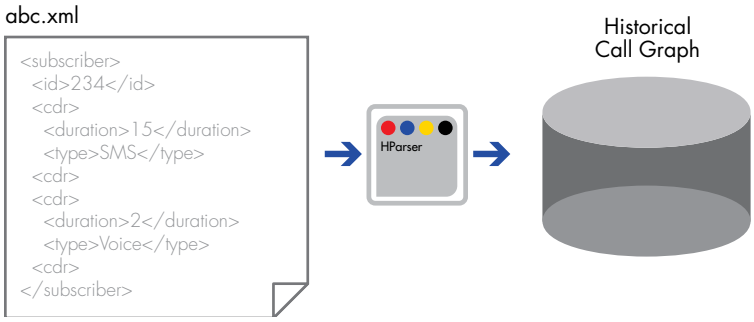


Figure 10. Loading XML data to database

Finally, HParser is configured with connectivity information for HBase in order to move the contents of XML files into HBase.

Step 4 – Planning

At this point, we can estimate development effort and required system capacity (see Figure 11).

STEP	TIME	PERFORMED
1	2-5 days	once per data source
2	4 days	once
3	4 days	once

Figure 11. Sample timeline estimates

Step 1 requires the most development time because it has to be repeated for every new data source. On-boarding a new data source can take as little as a few days for a simple format and as much as five days for a more complex one.

Steps 2 and 3 only need to be developed once. These are simple transformations that should each take no more than four days each.

Estimated system capacity encompasses both processing capacity for collecting and organizing data and storage capacity for staging CDRs in HDFS and building out HBase over time.

Processing capacity is a function of both the amount of data that needs to be processed and the time window in which processing needs to complete.

Amazon's cloud hosted Hadoop platform, EMR (Elastic Map Reduce), facilitates quick and inexpensive path for estimating system processing capacity. EMR can easily access S3 (Simple Storage Service) storage hosting CDR test data sets (see Figure 12).

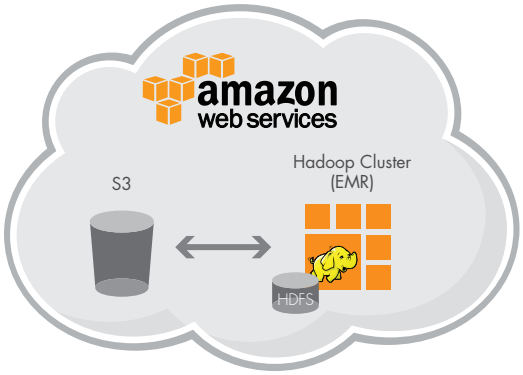


Figure 12. Amazon's cloud-based Hadoop platfrom

Running steps 1-3 on EMR's cloud-based Hadoop should provide the number of available nodes required to complete the task in the required amount of time. The benchmarking section below provides an example of this calculation.

Storage capacity for HBase, both currently and for anticipated volumes over time, can also be determined by running initial loads to HBase on EMR and then projecting future loads.

First Steps and Rapid Experimentation

Taking the first steps toward a plunge into big data is often a difficult decision. It should not be.

Amazon's dynamically scalable EMR environment supplies the perfect opportunity for initial testing of HParser applications. It also promotes rapid experimentation and allows running the system under a full load.

Configuring an on-premises Hadoop installation is an important step in Big Data adoption process. However, it often takes many months of justifying investments for thousands of dollars, procuring hardware and software, and configuring the cluster. Initial clusters are generally small and don't reflect the real power of big data processing.

With EMR users can start running jobs in days at minimal cost—a few hundred dollars will go a long way. Once the system is functioning, it can be easily scaled with a press of a button. Resulting benchmarks in turn accelerate putting together a compelling justification for a right-sized on-premises Hadoop system.

However, once the CDR processing system is designed, planned, and tested EMR offers unparalleled ROI and agility as a production platform.

HParser is certified for all major Hadoop distributions including Amazon EMR. You can find a detailed tutorial for using HParser with EMR at http://aws.amazon.com/articles/0124533548208923 (or by searching on keywords "hparser emr tutorial").

Benchmarks

The following two simple HParser tests are designed to estimate required capacity and provide a sense of raw, unoptimized performance.

Both tests ingest a 16 MB binary file with CDRs described by ASN.1 grammar. These are real production files generated by switches belonging to a telco operator. HParser transforms binary CDRs to textual call records represented in CSV format. Twenty elements are extracted from each binary record and appear in CSV.

Test 1

The first test demonstrates how increasing the number of nodes in a Hadoop cluster improves the processing of a fixed-size set of data—in this case, one directory containing multiple 16 MB binary files totaling 10 GB and a second directory containing 50 GB of data. Figure 13 depicts the results.

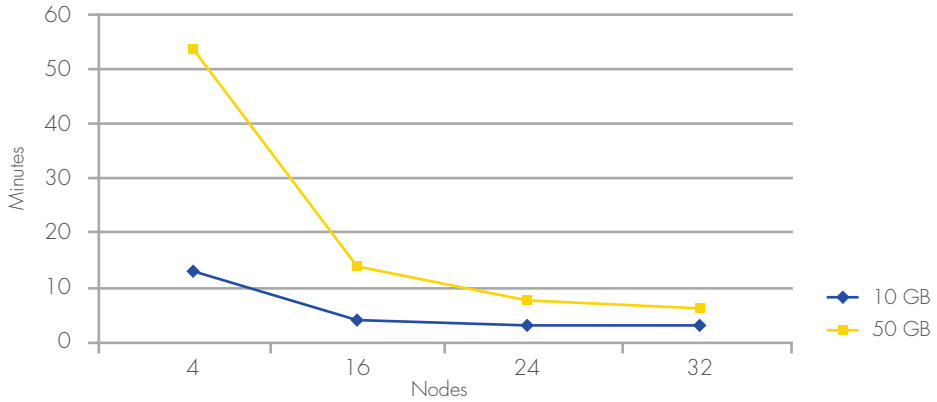


Figure 13. Processing ASN.1 data with HParser with increasing Hadoop cluster capacity

Test 2

The second test measures how quickly a Hadoop cluster of fixed size (in this case, 72 nodes) processes loads of varying data sizes up to 1 terabyte. Figure 14 presents the results:

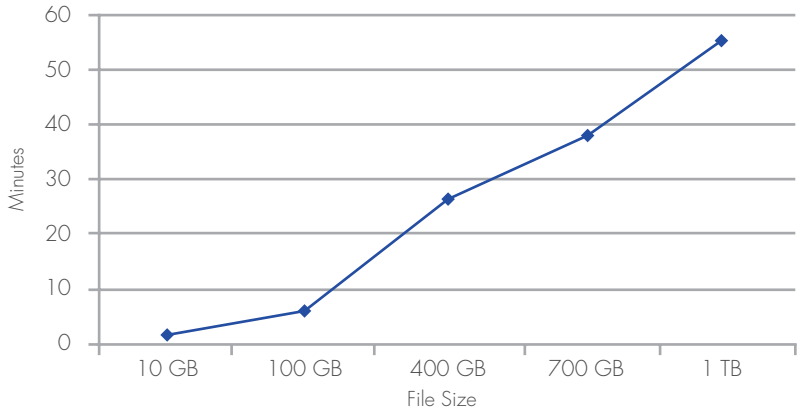


Figure 14. Processing ASN.1 with HParser on 72-node Hadoop cluster

Conclusion

The amount of data generated by telecom operators is only going to continue to grow. As the industry embraces big data, operators that are able to manage and analyze it faster, simpler, and better will have a significant competitive advantage.

HParser offers a way to achieve that competitive advantage. With HParser, telecom operators can quickly build affordable big data systems while leveraging their existing technical staff. They can also easily experiment in a cloud sandbox.

As a result, the operators can use Hadoop to derive the maximum value from big data and achieve a clear ROI that's measurable in direct business value. At the same time, the big data approach with HParser also fosters innovation, experimentation, and continuous improvement.

About Informatica

Informatica Corporation (NASDAQ: INFA) is the world's number one independent provider of data integration software. Organizations around the world rely on Informatica for maximizing return on data to drive their top business imperatives. Worldwide, over 4,630 enterprises depend on Informatica to fully leverage their information assets residing on-premise, in the Cloud and across social networks.

Put potential to work.™

Worldwide Headquarters, 100 Cardinal Way, Redwood City, CA 94063, USA Phone: 650.385.5000 Fax: 650.385.5500
Toll-free in the US: 1.800.653.3871 informatica.com linkedin.com/company/informatica twitter.com/InformaticaCorp

IN09_0313_02373_0513