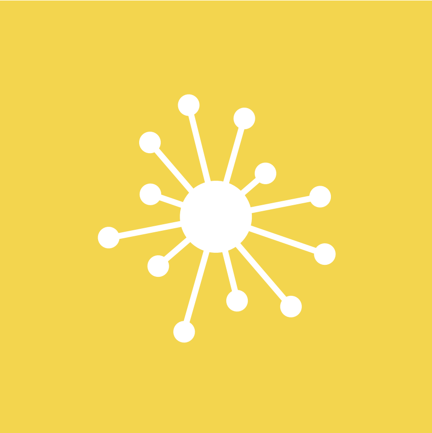
Big Data Analytics Solution Brief — Agriculture and Forestry
Intel Big Data Enablement — Open Standards
C/C++ Integration

Intel and Cloudera* Move a Company's SaaS Operation to Hadoop* Without Disrupting its C/C++ Development

Intel and Cloudera help a European agricultural SaaS company migrate to Cloudera Enterprise while safeguarding their investment in C/C++ development.



Why Intel and Cloudera

Intel and Cloudera take the guesswork out of Apache* Hadoop*. Using a unique collaborative approach, we deliver excellent performance, security, and quality distribution, built on open standards. Because we work with hundreds of vendors across the ecosystem, a solution built on Cloudera Enterprise can ensure freedom from lock-in, enabling you to build a robust Big Data solution to meet the needs of your business today and into the future.

- Uniquely aligned product roadmaps for software and hardware to drive innovation faster, providing many industry firsts with Hadoop*.
- Deep partnerships with virtually every provider in the data center, streamlining the process for building Big Data solutions.
- Proven track records of identifying the driving industry standards, so you don't run the risk of stranding yourself on an island.

A European company that creates custom software tools and SaaS for agricultural applications wants to update its environment to a Hadoop*-based Big Data platform, while maintaining its C/C++ legacy. The Company's current applications deal with yield prediction, livestock breeding best practices, and support for irrigation, fertilization, and crop protection decisions. Their product roadmap calls for adding satellite image processing capabilities to these apps, all of which have been developed—and will continue to be developed—in C/C++.

After a few unsatisfactory attempts to migrate to a Hadoop*-based system on their own, the Company asks Intel to help them integrate their existing C/C++-based operations into Cloudera Enterprise.

Results

- Intel migrates the Customer's existing C++ application to run on Cloudera Enterprise.
- Intel integrates the open source MR4C (MapReduce for C), which enables the Customer to continue using C/C++ for development work, only now in a Hadoop* environment.
- Retaining C/C++ design for the development platform protects the Company's investment in existing code and engineering mindshare.
- The Company immediately gains the framework, experience, and support benefits of belonging to the wider Hadoop* community.
- The Hadoop* platform opens the flexibility for the Company to optimize code for greater compute power (if required) through GPU work, etc.

Business drivers

Two main drivers propelled the Customer toward Hadoop*. They had been collaborating with other Hadoop*-based organizations and sharing these partners' resources for some time. Shifting to Hadoop* would provide the Company the benefits of existing infrastructure immediately and lay the groundwork for them to expand and grow in terms of compute power. They also wanted to join the open source community and focus their efforts on product development, rather than customizing operational solutions to fit their needs.

One critical requirement for the Company was the need to retain interoperability with its existing C++ applications. The Company has meticulously roadmapped its flagship applications for future development on the C++ platform, with the intention to add enhanced satellite image processing and greater data sets for analysis in near-future releases. Abandoning their C++ legacy was simply not an option.

Solution details

The Company tried a few methods to add Hadoop* to their ecosystem, including maintaining application development in C++ on an Open MPI cluster arrangement while using C++/Java integration technologies to move everything onto a Hadoop* cluster. After a few internal attempts yielded less than satisfactory results, the Company asked Intel to help integrate their C++ applications with Hadoop*.

Intel's first attempts along similar lines were either too cumbersome or too complex, requiring extra work for little gain. Finally, we evaluated an open source imaging library/framework created by Google Skybox* called MR4C (MapReduce for C). Our evaluation of MR4C, which Google Skybox recently released to the Open Source community (with Apache 2.0), was positive enough that the Company agreed to adopt it going forward.[1]

The MR4C framework was developed by Skybox Imaging to aid in large satellite image processing using common, fast, and efficient image processing C++ libraries. This further aligns the Company's existing strategy of sticking to its C/C++ development platform with its product roadmap of adding satellite imaging to future product releases.

MR4C runs atop YARN and interacts with HDFS, using JNA (Java Native Architecture) to communicate with C++ libraries (*Figure 1*). MR4C packages the C/C++ code and shares libraries in a YARN job, which it executes using a JNA API. JNA details are largely "hidden" from the C/C++ developer, making it seamless.

An MR4C API allows native C++ code to run in the cluster. This API provides a wide platform for designing complex interactions. A configuration controls cluster/resource requirements, HDFS inputs and outputs, input splits, and native library loading. Many jobs can run across the data nodes as per normal Hadoop* execution.

Using MR4C in a C++ application requires a simple API implementation:

```
void
executeAlgorithm(AlgorithmData&
data, AlgorithmContext& context)
```

The *AlgorithmData* object provides APIs for access to the HDFS files, parameters, and inputs. And the *AlgorithmContext* object provides simple access to logging, messaging, and other Hadoop* runtime properties. Together with these APIs, C/C++ developers are free to implement map, reduce, and simple algorithms using their preferred libraries.

The new application design uses a MapReduce style master/worker model whereby the worker nodes process the data and return results to a master coordinator, which can in turn continue the phase of data minimization to more worker processes as required.

We created two Docker* images[2] during the development phase to ease the MR4C development. These

[1] For more information on MR4C, visit *http://github.com/google/MR4C*.

[2] Contact your Intel representative if you are interested in these Docker* images.

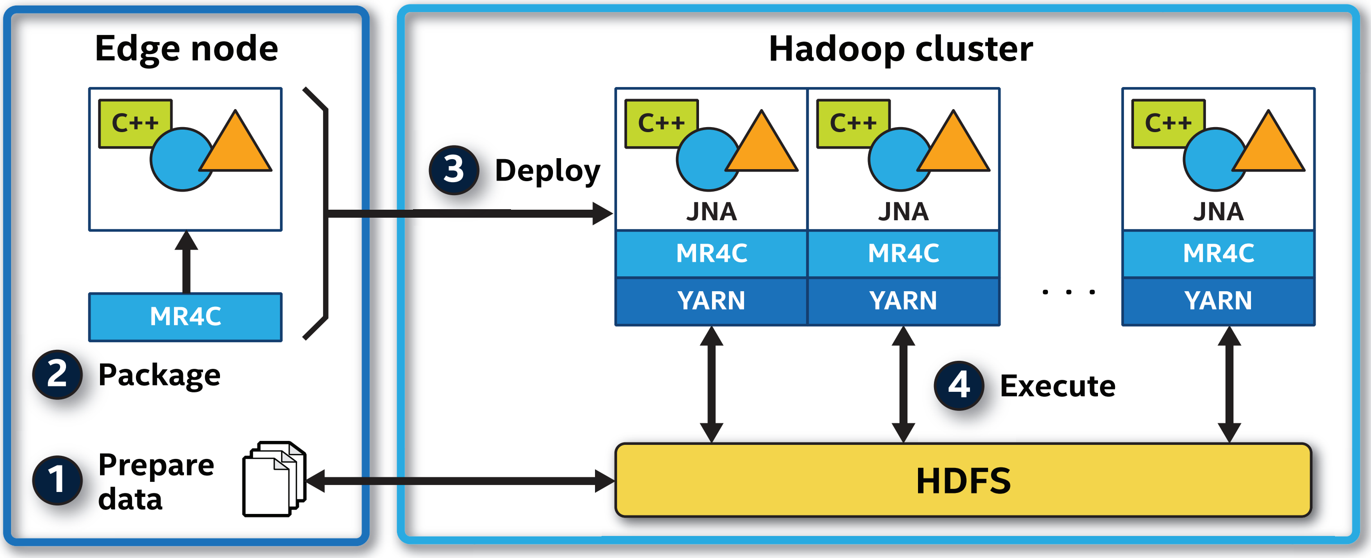
Figure 1 **Running MR4C jobs.** MR4C is both the API and the delivery mechanism for launching jobs into Hadoop*.

1 **Preparation.** Data is prepared and copied onto HDFS using the HDFS tooling.

2 **Packaging.** MR4C command line is launched, which packages your library and dependencies into a Yarn job.

3 **Deployment.** MR4C deploys the package to Hadoop* with your supplied configuration for resource requirements (nodes, configuration, data in/out, etc.).

4 **Execution.** Yarn executes the MR4C custom API, which then executes your C++ library via JNA.



Docker* images can be "taken" to the cluster and used in place without the need for complicating the cluster install. This provided us with a repeatable setup for two environments:

- Centos6 – Matching RHEL 6
- Ubuntu 15

Due to the nature of C++ libraries and JNA loading, the correct Docker* image must match the Hadoop* cluster in use. If the image and environment don't match, compiling and launching an MR4C development from a workstation causes unusual results and errors.

Although the overall solution met the Customer's requirements, Intel identified a few caveats:

- Data files and results are read from and written to HDFS. Because I/O to and from HDFS needs to comply with POSIX streams, per the MR4C framework, we do not recommend direct file access, to align to the data locality approach with HDFS.
- Some third-party libraries may be too complex to use with Hadoop*/MR4C. We discovered that the complex CERN ROOT library was incompatible, and we needed to rewrite the custom data tree storage to use Apache Avro serialization away from ROOT CERN.

Cloudera Enterprise

Intel experts showed the Company how to integrate Cloudera seamlessly with their existing C/C++ platform. Cloudera Enterprise lets businesses handle rapidly increasing volumes of data and a variety of workloads from existing systems while optimizing the efficiency of legacy infrastructure.

Other factors that played into the Company's decision to select Cloudera include:

- **Scalability.** Cloudera uses Apache HBase*, a distributed, scalable database that runs on top of the Hadoop Distributed File System (HDFS), a fault-tolerant and self-healing distributed file system. HDFS accepts data in any format, optimizes for high bandwidth streaming, and scales to proven deployments of 100 PB and beyond. It stores data on commodity machines, which makes it affordable and easy to add new servers or hotswap failed drives on the fly as the volume of data increases and the need for more space grows. In short, a Hadoop-based solution offers lower costs and faster performance.
- **Enterprise features.** Cloudera offers several key enterprise-level features needed for IT compliance, including encryption at rest and in motion, Simple Network Management Protocol (SNMP) support and alerts, rolling updates, AD/Kerberos integration, and automatic backup and disaster recovery (BDR).
- **Standardization.** Because Cloudera Enterprise is based on open source components, the Company can integrate Cloudera Enterprise with other open source tools. This further opens the Company's C/C++ development to the Hadoop* ecosystem for future development. The Company also now has access to a larger number of "off-the-shelf cluster" personnel than they had with a customized system.
- **Support.** Excellent maintenance, stability, and support from Cloudera and the wider Hadoop* community on cluster capabilities.

Summary

The new framework integration sets a good footing for future development, widening technical options and support possibilities for the Company's development. Areas of future R&D may include GPU processing, satellite image processing, and additional, more complex C++ libraries.

Intel designed a system that helped the Company transparently integrate its existing C/C++ development platform with a scalable Hadoop*-based Big Data platform.

Let us help your business too.

Spotlight on Cloudera

Cloudera is revolutionizing enterprise data management by offering a unified platform for Big Data, an enterprise data hub built on Apache* Hadoop*. Cloudera offers enterprises one place to store, access, process, secure, and analyze all their data, empowering them to extend the value of existing investments while enabling fundamental new ways to derive value from their data.

Cloudera's open source Big Data platform is the most widely adopted in the world, and Cloudera is the most prolific contributor to the open source Hadoop ecosystem. As the leading educator of Hadoop professionals, Cloudera has trained over 40,000 individuals worldwide. Over 1,900 partners and a seasoned professional services team help deliver greater time to value. Finally, only Cloudera provides proactive and predictive support to run an enterprise data hub with confidence. Leading organizations in every industry plus top public sector organizations globally run Cloudera in production.

For more information, visit **www.cloudera.com**.

cloudera®

Meeting your needs

We look forward to meeting with you to define your requirements and meet your objectives.

- **Accelerate time to value:** Achieve real-time cost savings, respond to market trends, and drive innovation.
- **Secure Big Data:** Deploy a sustainable Big Data program that doesn't put your organization, or you, at risk.
- **Maintain control:** Work with a partner who educates your team so you become self-sufficient.
- **Increase business potential**: Create and execute a plan that helps you adapt now, and in the future.

Hadoop sizing guide

		Cluster size		
		Small	Medium	Large
CPU		Intel® Xeon® Processor E5 v3		
Storage (TB)		<72 TB	72 to 570 TB	>570 TB
Node count	Master	2 to 3	4 to 7	≥8
	Slaves	<12	12 to 95	≥ 96
Memory (GB)	Master	64 GB	128 GB	≥256 GB
	Slaves	48 GB	96 GB	≥128 GB
Network		1 Gbps	10 Gbps	10 Gbps

Hardware configuration is highly dependent on workload. A high storage density cluster may be configured with a 4 TB JBOD hard disk, while a compute intensive cluster may be configured with a higher memory configuration.

Contact us

Contact your sales rep or e-mail us at: Hadoop-services@intel.com.

Intel.com/bigdata/services

The results cited in this document are based on research and testing conducted by Intel and Intel's customer and are for informational purposes only.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit *http://www.intel.com/performance*.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at *www.intel.com*.

Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725 or by visiting Intel's website at *http://www.intel.com/design/literature.htm*.

Please Recycle