Achieving Agility and Flexibility in Big Data Analytics with the Urika®-GX Agile Analytics Platform

Analytics R&D and Product Management
Document Version 1
# Table of Contents

Introduction ................................................................................................................................. 3
Big Data and Agility ....................................................................................................................... 3
The Urika-GX Agile Big Data Analytics Platform ................................................................. 6
  Software Stack Overview ....................................................................................................... 6
Pre-Installed Analytics and HPC Frameworks ........................................................................ 7
Graph Analytics on the Urika-GX System with Cray Graph Engine .................................. 8
Building Data Pipelines and Data Movement ................................................................. 9
Multiple Workload Orchestration on the Urika-GX Platform ........................................... 9
  Resource Allocation in Mesos ............................................................................................. 10
  Hadoop/Analytics Services: Marathon ............................................................................... 12
  Hadoop on Mesos ............................................................................................................... 12
  Spark on Mesos .................................................................................................................. 12
  CGE on Mesos .................................................................................................................... 13
  Support for New Frameworks ............................................................................................ 13
Use Case Examples ................................................................................................................... 13
  De Novo Assembly in Genomics ...................................................................................... 13
  Financial Services Surveillance ......................................................................................... 14
Conclusions ................................................................................................................................. 15
Introduction

The Cray® Urika®-GX agile analytics platform is a high-performance software-hardware solution for big data analytics. It enables data-driven companies to improve operational efficiency and swiftly respond to changing needs, including supporting emerging tools, providing faster response times and increasing utilization.

It delivers the power of Cray’s proven supercomputer hardware and comes pre-installed with a variety of open source big data analytics tools such as Apache Hadoop® and Apache Spark™ and productivity tools such as Jupyter Notebooks. The Urika-GX system also supports high performance computing (HPC) applications running under global resource allocator Apache Mesos™. Additionally, the Urika-GX platform:

- Allows easy scalability for new applications
- Enables creation of complex data pipelines
- Features Cray’s proprietary Aries™ high-speed network, which offers high bandwidth and low latency in data movement across the analytics pipeline

This paper details the Urika system’s software architecture and explains how that architecture makes it an agile analytics platform.

Big Data and Agility

Big data analytics are a set of processes and methodologies used to derive insights from collected data and assist in decision making. With advances in digital technologies and mobile devices, as well as internet proliferation, data is being generated at increasing velocity. Additionally, data formats have evolved from pure text or tabular to include images, videos, audio and speech. Previously difficult to analyze due to limitations of storage and processing technologies, these new data formats can now be processed and analyzed efficiently with the emergence of big data technologies such as Apache Hadoop and Apache Spark.

Given the current volume, velocity and processing demands, a big data analytics platform must:

- Accommodate the installation of new and upcoming analytics applications
- Support a range of applications to cover multiple response time requirements
- Handle different frameworks, including HPC, interactive, batch and iterative workloads
- Minimize data movement across different components of analytics pipelines
- Provide scalable and extendable global resource management tools to enable usage by multiple teams for different applications
The answer to these varied demands is the Cray Urika-GX agile analytics platform. Taking an agile analytics approach allows an enterprise to tailor its products and services, optimize its operations and identify new business models, ensuring its competitive position.

Big data technologies are built around the concept of data locality, in which processing is done closer to the data instead of moving the data to processing. It requires this approach because while processor speed has followed Moore’s Law by doubling approximately every 18 months, data movement remains a bottleneck in large-scale data processing. That bottleneck increases the cost of building big data platforms. The data movement problem is further amplified by the multiplicity of tools and new methodologies coming into the marketplace. A competitive enterprise must adapt to these new technologies if it wants to maintain its leadership position.

Enterprises acquire big data to answer questions they believe will come up in the future. Organizations are now moving from simple descriptive analysis of data to data-driven decision making by using advanced analytics and deep-learning technologies. Advanced analytics methodologies involve the use of machine learning, predictive models, prescriptive models and graph analytics. The unknown nature of future problems means a big data infrastructure must be agile and flexible to prevent rapid obsolescence.

Figure 1. Big data analytics performance spectrum
The performance demands on today’s analytics applications vary widely (Figure 1). At one end of the spectrum are batch analytics, which have response-time requirements in minutes, while at the other end of the spectrum real-time streaming analysis requires response times in microseconds. Additionally, as organizations adopt big data technologies, different components in this spectrum are tied together to create complex data pipelines. This leads to challenges for data science and analytics teams:

- Quickly moving data across components of the pipeline
- Quickly incorporating new tools and technologies

Traditionally, big data platforms are customized for specific types of applications, with rigid configurations for both hardware and software. An agile big data platform incorporates new applications and technologies, and it allows users to create complex analytics pipelines with multiple workloads on the same machine, minimizing data movement.

The open Urika-GX platform solves the problems of traditional analytics platforms by enabling the adoption of new technologies. At the same time its high-speed Aries network speeds communication among different workloads.
Software Stack Overview

The Urika-GX system’s software stack (Figure 2) is built on CentOS 7 with customization for Cray’s Aries high-speed network. It uses Apache Mesos as global resource manager, which dynamically partitions the platform for different types of workloads. Popular analytics frameworks such as Apache Hadoop, Apache Spark and related tools are included. The platform also comes pre-installed with the highly scalable Cray Graph Engine (CGE), which has a proven track record of scaling up to billions of triples for analyzing complex graph problems. The stack also includes R and Python interpreters, including associated data analytics libraries and packages. These interpreters and Apache Spark are available through Jupyter Notebooks, which gives users quick prototyping and collaboration capabilities.

For clear and simplified access to different Web-UIs, Cray has built an intuitive web interface for constituent application UIs, learning resources, application management, resource management and product documentation. The system management of the Urika-GX platform is based on OpenStack® technology and consists of standard and custom interfaces providing easy cloud-like management capabilities.

Figure 1. Cray Urika-GX system software stack
Pre-Installed Analytics and HPC Frameworks

The broad response time requirements for different analytics applications require a wide variety of tools and applications. The Urika-GX platform comes pre-installed with tools such as Apache Hadoop and Apache Spark that handle response time needs for batch, interactive and iterative workloads.

Apache Hadoop is an established data processing framework for batch applications. It provides a scalable, cost-effective, flexible and fault-tolerant computing solution. The Hadoop ecosystem uses YARN (Yet Another Resource Negotiator) as its default resource manager and consists of Hadoop Distributed File System (HDFS) for storage and MapReduce for processing.

Hadoop requires fast local storage, and it benefits from high-bandwidth I/O. On the Urika-GX platform, HDFS storage is set up on solid-state drives and hard disk drives on each of the compute nodes. HDFS data is accessible from other frameworks, including Spark and CGE.

Also pre-installed on the Urika-GX platform, Apache Spark is a general-purpose cluster-computing framework. It provides for parallel processing of batch workloads, as well as in-memory data analytics capabilities; iterative, incremental algorithms; ad-hoc queries; and stream processing. The commonly used components of Spark include SparkSQL for structured data processing, MLlib for machine learning, GraphX for graph processing, and Spark Streaming for real-time stream processing capabilities.

![Figure 2. Spark driver, worker communication](image)

A Spark job (Figure 3) consists of a driver — also called the master — and a set of executors, called the workers. The driver executes the user’s main function and distributes work to the executors. The driver and each of the executor processes run in a separate Java virtual machine (JVM) and communicate with each other over TCP through the process called shuffle.
Shuffles are divided into two phases: the shuffle write (send), called map tasks, and the shuffle read (receive), called reduce tasks. During the shuffle write, each map task sends data to reduce tasks by writing it to intermediate files, often cached by the OS. During the read, each reducer fetches the data that was sent to it by reading the intermediate files, fetching these over the network if they are not local. I/O bandwidth gates the performance of shuffles, and in turn the performance of Spark operations that require data movement. To mitigate the impact of shuffle I/O, the Spark shuffle intermediate file directory was moved to a node-local SSD-mounted directory on the Urika-GX platform.

**Graph Analytics on the Urika-GX System with Cray Graph Engine**

Cray Graph Engine (CGE) is a highly optimized and scalable graph analytics application designed for high-speed processing of interconnected data. It enables users to search large, graph-oriented databases and query for complex relationships between data items in the database. CGE supports optimized procedures for inferencing/deduction, pattern-based queries and intuitive visualization, and is designed to provide performance and scalability on large, complex, interconnected databases. In contrast to Hadoop and Spark, which were originally developed to run on commodity clusters, CGE was developed to run in a Cray HPC environment. It is based on the industry-standard RDF graph data format and SPARQL query language for easily creating graph queries.

Because the data in an RDF database is graph structured, any node in the graph may share edges with any other set of nodes distributed across the system’s memory. Nodes may share edges with a few other nodes, or very highly connected nodes may have hundreds or thousands of immediate neighbors. This situation poses problems for distributed memory systems, since communication of information across the dataset is highly irregular, and may approach all-to-all node level communication for tightly connected graphs.

Therefore, an ideal platform for this type of graph database is a shared memory system with fast and uniform access to all memory locations. A more cost-effective alternative is to take advantage of the global address space and the synchronization features provided by Cray’s Aries network to create a scalable high-performance distributed application which can both optimize for locality and tolerate the highly irregular and all-to-all communication inherent in a semantic graph database.

With Hadoop and Spark ecosystems along with CGE, the Urika-GX system’s software stack covers the full spectrum of response time requirements for a data-driven enterprise. It enables the creation of sophisticated data pipelines on the same platform.
Building Data Pipelines and Data Movement

The aforementioned frameworks and components are the building blocks of complex data pipelines. A data pipeline consists of a data integration layer, feature extraction layer, steam and batch processing layer, storage layer, reporting layer and visualization layers. The complexity of these pipelines increases as organizations request new tools, reporting and processing.

In response, the Urika-GX platform offers Jupyter Notebooks for quick prototyping and collaboration across data science teams. Jupyter Notebooks is configured for Shell, R, Python and Scala for Spark, and Cray also provides sample codes for quick ramp-up available through the Urika-GX application interface.

As the complexity of data pipelines increases, the amount of data to be moved across different components also increases and becomes a bottleneck. A data platform must provide for low-latency communication across components, irrespective of component type. This allows the users to develop optimized workflows by selecting the framework most appropriate for each task.

A common approach for data exchange is to utilize a shared file system. Analytics frameworks on the Urika-GX platform can access both the HDFS file system and the Cray® Sonexion® Lustre®-based file system, which can be attached to Urika-GX as an option.

Maintaining data locality is highly desired for most big data applications, as performance can be significantly impacted for nonlocal access to data. The Aries network’s low-latency, high-bandwidth capabilities help mitigate a strict dependence on data locality and tolerate nonlocal access to data. This capability allows workflows that include multiple frameworks running simultaneously on separate sets of nodes to exchange data, either through HDFS or Lustre. It means organizations can create streaming workflows where pipelines span multiple frameworks.

Multiple Workload Orchestration on the Urika-GX Platform

The ability to install different types of analytics applications is one aspect of an agile analytics platform — but dynamic resource management of these applications is another important consideration. Ultimately users want to be able to run different workloads without needing IT/DevOps to configure the system for each of these applications individually.

The Urika-GX platform uses Apache Mesos, a distributed cluster resource manager, to provide efficient resource allocation and sharing across distributed applications or frameworks. Mesos supports diverse frameworks running on the same platform without static partitioning, thereby improving system utilization.
Having big data frameworks such as Hadoop and Spark, along with HPC, supported on the same platform allows the user to select the best framework for each application, and it allows the creation of complex workflows without the need for dedicated clusters. Mesos provides a common interface that allows cluster resource management for existing frameworks and supports the addition of future frameworks.

Resource Allocation in Mesos

The ability to install different types of analytics applications is one aspect of an agile analytics platform — but dynamic resource management of these applications is another important consideration. Ultimately users want to be able to run different workloads without needing IT/DevOps to configure the system for each of these applications individually.

The Urika-GX platform uses Apache Mesos, a distributed cluster resource manager, to provide efficient resource allocation and sharing across distributed applications or frameworks. Mesos supports diverse frameworks running on the same platform without static partitioning, thereby improving system utilization.

Having big data frameworks such as Hadoop and Spark, along with HPC, supported on the same platform allows the user to select the best framework for each application, and it allows the creation of complex workflows without the need for dedicated clusters. Mesos provides a common interface that allows cluster resource management for existing frameworks and supports the addition of future frameworks.

The Mesos master decides how many resources to offer to each framework (Figure 4). The framework scheduler then selects which of the offered resources to use. The framework chooses to either accept or decline the resources. When a framework accepts offered resources, it passes to Mesos a description of the tasks it wants to launch (Figure 5). The master allocates requested resources and sends tasks to the agent. A framework can reject the resources that do not satisfy its requirements or constraints (Figure 6).

Figure 4. Resource offers
This allows frameworks to ensure constraints such as data locality are met. Mesos shares resources in a fine-grained manner. As part of the Mesos ecosystem, the Urika-GX system features the Cray-built mrun application launcher command. With mrun users can run parallel jobs such as CGE and HPC applications. By leveraging different resource managers to interoperate with Mesos, the Urika-GX platform supports mixed workloads without the need to statically partition platform resources. The frameworks are configured to interact with Mesos for acquiring resources to launch their jobs. For frameworks not native to Mesos, the Urika-GX platform provides interfaces to allow these resource managers, such as mrun and YARN, to dynamically acquire and release resources from Mesos (Figure 7).
Hadoop/Analytics Services: Marathon

On the Urika-GX system, Hadoop/YARN and CGE are launched through Marathon using Cray-built extensions to the Marathon framework, which is a registered Mesos framework. Marathon provides RESTful APIs for starting, stopping and scaling long-running services.

Hadoop on Mesos

The Urika-GX platform uses Marathon to configure a dynamic YARN sub-cluster. Cray-built flex-up command dynamically starts YARN node managers as Mesos agent tasks using Marathon, which negotiates with Mesos for resources for setting up YARN cluster. Once the YARN cluster is flexed, users can submit Hadoop jobs. Flex-down scripts tear down the dynamic YARN cluster and return the resources to Mesos.

Spark on Mesos

Spark runs as a native Mesos framework, and it interacts directly with Mesos for resource allocation. On the Urika-GX platform, Spark uses Mesos as its resource scheduler. The coarse-grained mode launches only one long-running Spark executor on each Mesos slave. The Spark executor manages the parallel tasks across all the cores on that node.
Within a Spark application, the Spark scheduler schedules tasks across executors running on Mesos slaves. By default, Spark uses coarse-grained mode, which acquires and holds on to all requested cores. Fine-grained mode launches each Spark task as a separate Mesos task, enabling more fine-grained and dynamic sharing of the cluster, but at the cost of significantly higher startup overheads.

CGE on Mesos

Applications that can leverage native Aries communications protocols require special network initialization prior to launch. This is different from big data frameworks like Spark and Hadoop that communicate exclusively via the ipogif interface for TCP over Aries, where they do not require this initialization. CGE is one example on the Urika-GX platform. The network initialization is traditionally done by a scheduler during node allocation. Urika-GX uses the mrun job launcher to run CGE jobs using Mesos. The mrun launcher launches HPC-style workloads (including CGE) as Marathon jobs, and handles all of the required Aries network initialization.

Support for New Frameworks

Mesos provides high-level API to port new application frameworks to run on the Urika-GX platform, providing the flexibility to use the platform for future workloads. Cray-developed mrun is an excellent example of the platform’s expandability of resource management infrastructure and its ability to adapt to new frameworks.

Use Case Examples

De Novo Assembly in Genomics

De novo assembly is the method of determining the nucleotide sequence of a contiguous strand of DNA without the use of a reference genome. It is a computationally challenging and memory-intensive process that is critical to exploring novel genomes and highly varied portions of the human genome. Many agricultural research projects depend on this method since a good reference is often absent. Downstream genomic interpretation requires large-scale big data integration with a wide variety of sources, both structured and unstructured. With recent software advances, the Urika-GX agile analytics platform allows de novo assembly to scale to 1,700 cores, providing an excellent big data platform for data prep and downstream interpretation/integration.

The Aries interconnect is critical to the extreme parallelism required for high-speed assembly. Additionally, the large memory configuration is a critical factor during the assembly process, as is the Spark-optimized platform for downstream genomic interpretation. CGE provides a highly differentiated analytical capability. The higher throughput provides researchers a practical method to leverage de novo assembly more broadly and to scale to higher coverage depths, leading to a higher-quality assembly.
Higher throughput translates directly to lower cost. Being able to prepare the data, perform the assembly and perform the advanced analytics required to interpret the results all on the same platform simplifies the compute infrastructure and eliminates expensive and time-consuming data movement.

**Financial Services Surveillance**

Today’s financial service businesses require monitoring of systems, trades and employee interactions for unusual activity that could be related to systems failure or fraud. Large volumes of systems log, database and unstructured data need to be merged, connections and patterns found, and the findings operationalized. To establish unusual patterns, data scientists want all the tools to work on all of the data on a single productive platform — workloads that might include machine learning for advanced correlation and graphing for links and causation.

In Figure 9, multiple data sources such as plain text files, compliance rules, social media feeds and email transcripts are streamed using Apache Kafka™ into the application. The data is further saved in an Apache Cassandra™ database for future usage. The data is also fed to CGE- and Spark-based machine learning applications for anomaly detection, pattern matching and advanced correlation. The results of the machine learning and graph applications are made available to analysts through web interfaces for interactive analysis. Urika-GX allows for the creation of this complex pipeline on the same platform and uses HPC and big data frameworks on the same platform, minimizing the data movement typically encountered in traditional platforms.

The Urika-GX platform enables all these forms of analytics, including nonpartitioned graphing on a single platform, and the Aries interconnect enables improved machine learning and Spark performance. The Cray Sonexion-based storage helps in loading and managing massive data volumes. Its integrated parallel file system enables faster simulation load times, resulting in higher productivity, faster job turnaround, faster time to insight and lower TCO.

---

*Figure 9. Financial services surveillance — data pipeline*
Conclusions

While enterprises embrace big data technologies to solve their most complex business problems, their efforts are marred by the problems of efficiently implementing these technologies. At the core of these problems is the increasing sophistication of data pipelines requiring fast communication across pipeline components, as well as new applications coming onto the market. This situation requires a future-ready big data platform with the agility to run multiple and varied workloads with different response time requirements. Efficiently and dynamically running multiple workloads on the same machine not only minimizes data movement, which is a major bottleneck, but also allows higher utilization of the infrastructure by opening it up for multiple departments and teams.

The Cray Urika-GX agile analytics platform is designed to solve those issues. It provides an open and flexible architecture that enables data-driven enterprises to harness the power of big data for quick decision-making and business growth. Combining Cray’s supercomputing technologies, state-of-the-art big data software, and the Aries high-speed network, Urika-GX is the ideal platform for agile big data analytics.

© 2017 Cray Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owners.

Cray, the Cray logo, Urika and Sonexion are registered trademarks of Cray Inc. Other product and service names mentioned herein are the trademarks of their respective owners.