

YarcData's uRiKA Shows Big Data Is More Than Hadoop and Data Warehouses

Published: 11 September 2012

Analyst(s): Carl Claunch

The hype about big data is mostly on Hadoop or data warehouses, but big data involves a much wider and varied set of needs, practices and technologies. We offer recommendations for IT organizations seeking a solution to "graph" problems, including use of the uRiKA graph appliance.

Impacts

- IT organizations faced with previously infeasible graph-style discovery problems may succeed using a focused solution like uRiKA.
- To address all their data requirements, IT organizations may be forced to duplicate data between systems such as uRiKA and transactional systems.

Recommendations

- Select candidates to place on uRiKA where processing is graph-oriented, the data is large-scale, and discovery of complex relationships is a core focus.
- Validate the appropriateness of specialized systems and achievability of performance targets by proof-of-concept and pilot tests.
- Carefully define the volume and performance requirements for processing data.
- If requirements can be met using fewer machine types and the economics make sense, use the fewest platform types possible.

Analysis

To listen to the hype, you might think big data is only about Hadoop, but Gartner deems big data to comprise a wider, more varied set of needs, practices and technologies. The uRiKA graph appliance from YarcData, a new company spinoff of Cray, illustrates our point and is used here to highlight just one of the classes of big data problems that are poorly addressed by traditional systems.

The Nature of Most Enterprise Data

IT groups understood the interrelationships of most of their data — orders connect to customer information, product data, inventory facts and financial figures. IT experts use this knowledge to enable efficient access. Technical staff use a variety of mechanisms in database management systems (DBMSs) and analysis software to optimize access to the data. Indexes speed up access to information expected to be needed. Techniques like aggregation records give high performance for commonly requested information. IT groups can apply such optimization methods to both operational systems and data warehouse (DW) or business intelligence (BI) systems, because they:

- Know the nature of the questions that will be asked about data
- Understand how data elements are related and connected
- Know the parts of the data where they expect most activity (for example, current-year orders versus historical data)

Experts use prior information about data access patterns to define the database, apply indexes or aggregations, spread data across storage devices, and apply the right tools to meet service-level expectations. It is obvious that product data is related to sales data, but building temperatures usually aren't related to sales or product data. Access to financial data clusters close to the current date, and the frequency of requests about past years drops steeply. Customer service representatives are likely to look at payment or charges for one customer. Metadata about access patterns, combined with predicted volumes and response time targets, is used to configure data and systems.

Most Big Data Situations Are Addressed by MapReduce or Other Traditional Technologies

Users now face volumes of data too large to work well or fit in traditional DBMSs or DW systems. To deal with the challenge, users looked to techniques invented to host global-scale public Web properties — sites like Google, Yahoo, Facebook and others — whose extreme volume of data must be partitioned and distributed across many servers. The design of most applications on these sites uses a MapReduce model,¹ in which the application will:

- Identify (map) which of a very large pool of servers should participate in running each transaction or inquiry
- Pass requests to and receive responses from potentially many servers for each transaction
- Combine responses (reduce) to produce the final result of the inquiry or transaction

MapReduce is the core of Hadoop,² an open-source project at the Apache Software Foundation. Hadoop includes an execution environment, development tools, file systems, DBMS software, BI tools and other capabilities to build large distributed systems. The effectiveness of the MapReduce model depends on the application mapping a request to just the servers with relevant data — if too many servers are involved, overall performance and capacity are impaired, the servers having the desired data are missed, and the outcome is incorrect. Success depends on the same metadata about access patterns used so effectively in traditional DBMSs and DW systems.

Knowing the related information, likely frequency, relationships and other information, technical staff can allot the huge volume to servers with a partitioning scheme optimized for performance, capacity and expected transaction types. Instead of the data on overloaded servers being serially dug through, properly distributed data can be searched in parallel by many servers, and a response produced in less time. IT groups can easily accommodate information that is too voluminous for traditional systems using a Hadoop-style system if they have solid metadata about access patterns.

Graph Problems in Big Data Are Different

Organizations have great difficulty in achieving good, consistent performance with a class of information searches called "graph problems." Information is represented by vertexes on a graph. These nodes are connected by edges, representing some relationship between the nodes. Any two nodes are connected by paths — a series of edges beginning at one vertex and ending at the other — that may pass many intermediate nodes. If one imagines many cities (vertexes) joined by roads, representing edges, many challenges of graph processing become clear. There may be many routes connecting two cities — some are longer or less direct than others. The task of discovering all paths and which are "best" can be difficult, as the number of cities and roads swells. Graph problems can represent many types of relationships as edges connecting vertexes.

In many graph applications, the user wants to discover patterns and connections between a myriad of facts, measure "distances" and paths between nodes, and choose the edge to traverse based on what has been learned up to that moment. The course of the search may leap large distances, when a relationship is discovered between distant nodes. The region of the graph in which the search spends most time could concentrate in unknown spots or spread across the full graph. A DBMS designed for known relationships and anticipated requests runs badly if the relationships actually discovered are different, and if requests are continually adapted to what is learned.

Examples of graph problems:

- Spotting unrealized connections between actions and people — none suspicious in itself — that represent a coordinated threat to public safety, enabling plots to be blocked
- Finding patterns and correlations in treatments and results to enable health organizations to personalize treatment, improving outcomes for patients and institutions
- Learning unrecognized factors that shift market demand, drive swings in investment prices and alter portfolio risks, fast enough to take corrective action

Figure 1. Impacts and Top Recommendations for Running Graph-Style Discovery Problems

Impacts	Top Recommendations
Organizations faced with previously infeasible graph-style discovery problems may succeed using uRiKa.	<ul style="list-style-type: none">• Select candidates where processing is graph-oriented, the data is large-scale, and discovery of relationships is the focus.• Test the appropriateness of specialized systems and achievability of performance targets.
To address all their data requirements, IT organizations may be forced to duplicate data between systems such as uRiKa and transactional systems.	<ul style="list-style-type: none">• Carefully define the volume and performance requirements for processing data.• Use the fewest platform types possible.

Source: Gartner (September 2012)

Impacts and Recommendations

IT organizations faced with previously infeasible graph-style discovery problems may succeed using a focused solution like uRiKA

Why Graph-Oriented Problems Behave Pathologically on Traditional Systems

"Big data" is a broad term, but many of these are graph problems, discovering connections by roaming across a melange of data sources. When the purpose of the system is discovery of relationships, not extracting information from already known interrelations, achieving satisfactory performance is difficult. The key to managing MapReduce workloads is the scheme by which the data is partitioned and assigned to specific servers in the cluster. Every inquiry or other transaction must be mapped to the servers that are relevant to the task, leveraging the scheme by which the data was placed. When the relationships among data are mysterious, and the nature of the inquiries unknown, no meaningful scheme for partitioning the data is possible. The inquiries themselves are likely to have to run on all the servers, and when the trail from one bit of data to a related one takes several hops, each to a different portion of the data housed in another server, the time spent in each server can vary dramatically. The most overloaded server then determines the response time to the original inquiry or transaction. As a result, it is challenging to pool multiple inquiries together in a way that spreads out the processing, if the pattern of processing is not predictable in advance.

Other challenges arise from graph-type processing due to irregular and unpredictable leaps along a route. The performance of almost all modern processors is dependent on locality of reference, to exploit very fast but expensive cache memories. When a series of requests to memory are near to

one another, the cache may return all but the first of the requests. The first request is slow, as RAM memory is glacially slow in comparison with cache memory. Similarly, when a program asks again for data it recently accessed, that data is likely to still be in the cache, available with little delay. Thus, locality of reference in time and in area allows a large majority of memory requests to occur at cache speed, masking the impact of much slower RAM chips. Caches in disk drives, storage systems and server memory achieve a similar effective speedup over the access times of disk drives.

When graph problems are processed, the irregular pattern means a lessened locality of reference in time. The large leaps in unpredictable directions along the graph mean lessened locality of reference in area. Both cause the effective performance of the processor and the storage systems to decline, sometimes substantially, as requests are now more likely to require the full delay of RAM or disk access times, since they are not in the caches.

A Design Accommodating Graph Problem Peculiarities Works Where Classical Approaches Fail

YarcData has designed uRiKA with three technologies to minimize or eliminate the costs of the irregular, unpredictable leaps in graph processing. A unique approach to processor design, YarcData's Threadstorm chip, shows no slowdown under the characteristic zigs and zags of graph-oriented processing. Second, the data is held in-memory in very large system memory configurations, slashing the rate of file accesses. Finally, a global shared memory architecture provides every server in the uRiKA system access to all data.

The Threadstorm processor runs 128 threads simultaneously, so that individual threads may wait a long time for memory access from RAM, but enough threads are active so at least one completes an instruction in each cycle. All 128 are active simultaneously, unlike other chips where only a few threads are active in one cycle; the threads all move at a moderate rate that is insensitive to locality of reference.

Employing a single systemwide memory space means data does not need to be partitioned, as it must be on MapReduce-based systems like Hadoop. Any thread can dart to any location, following its path through the graph, since all threads can see all data elements. This greatly reduces the imbalances in time that plague graph-oriented processing on Hadoop clusters.

Pick the Right Tools Based on the Nature of Your Big Data Problem

Thus, uRiKA is a system designed for the characteristics of graph problems. This example underscores one of the ways that big data can be more than just huge quantities, and that IT organizations may need unfamiliar or novel technologies when they face unique big data situations. uRiKA is not the general solution to all big data challenges, nor is it the only technology that might work adequately with a graph-oriented application, but it does prove that Hadoop-style systems are not the universal tool for big data.

Recommendations:

- Survey opportunities across the business for leveraging discoveries from graph-oriented processing into meaningful business advantages.
- Select candidates to place on uRiKA where processing is graph-oriented, the scale of the data is large, and discovery of relationships is a core focus of the work.
- Validate the appropriateness of specialized systems and the achievability of performance targets with proof-of-concept and pilot tests.

To address all their data requirements, IT organizations may be forced to duplicate data between systems such as uRiKA and transactional systems

Data sources may be used for multiple purposes, and the system that best addresses each purpose is different. This can cause organizations to have to duplicate data across these islands; this can multiply the costs of storing big data volumes in one location. Some portion of the data that is searched to discover new relationships may be best placed on a system optimized for graph-oriented processing, such as uRiKA, but that data may also be used in operational, transactional systems whose needs are far better addressed by traditional DBMSs and analytics software. Other parts of the enterprise's data may be cost-effective to host and search only on large Hadoop-style clusters, searching for information using well-understood relationships, yet this may also be a source for the discovery tasks that are graph-oriented in nature.

The same information might be transactionally processed on operational systems, with a copy placed in data warehouses to extract BI with the mature and powerful tools available for those purposes, another copy pumped into a Hadoop-style cluster for very large scale inquiries, and yet a third copy streamed into a graph-processing-optimized system. However, the inflated costs that come from all that duplication, the increased complexities of managing multiple technology islands and the downsides of establishing isolated islands are serious disincentives.

For many, the optimization from discrete approaches may not be worth the ramped-up costs and other impacts. IT departments may select a few technology types to handle all requirements — accepting performance or processing rate limitations to avoid the costs of too much diversity. IT departments that can meet their raw scale or performance levels only by adopting the platform right for the task will have to bear the increased costs that ensue.

Recommendations:

- Carefully define the volume and performance requirements for all types of processing required against data.
- Calculate the impacts of duplication, complication and isolation for each potential additional technology platform to be implemented.
- When the requirements cannot be met without diversification, build plans around the optimized platform type.
- When the requirements can be achieved on a compromise employing fewer machine types, as long as the economics make sense, use the smallest number of platform types possible.

Recommended Reading

Some documents may not be available as part of your current Gartner subscription.

"From Data to Decision: Delivering Value From 'Big Data'"

"Big Data Challenges for the IT Infrastructure Team"

"'Big Data' and Content Will Challenge IT Across the Board"

Evidence

¹ Paper describing the invention and use of MapReduce within Google — J. Dean and S. Ghemawat, "[MapReduce: Simplified Data Processing on Large Clusters.](#)" Communications of the ACM, January 2008.

² The [Hadoop project](#) at the Apache Software Foundation.

Regional Headquarters

Corporate Headquarters

56 Top Gallant Road
Stamford, CT 06902-7700
USA
+1 203 964 0096

European Headquarters

Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Asia/Pacific Headquarters

Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600

Japan Headquarters

Gartner Japan Ltd.
Atago Green Hills MORI Tower 5F
2-5-1 Atago, Minato-ku
Tokyo 105-6205
JAPAN
+ 81 3 6430 1800

Latin America Headquarters

Gartner do Brazil
Av. das Nações Unidas, 12551
9° andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509

© 2012 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner is a registered trademark of Gartner, Inc. or its affiliates. This publication may not be reproduced or distributed in any form without Gartner's prior written permission. The information contained in this publication has been obtained from sources believed to be reliable. Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information and shall have no liability for errors, omissions or inadequacies in such information. This publication consists of the opinions of Gartner's research organization and should not be construed as statements of fact. The opinions expressed herein are subject to change without notice. Although Gartner research may include a discussion of related legal issues, Gartner does not provide legal advice or services and its research should not be construed or used as such. Gartner is a public company, and its shareholders may include firms and funds that have financial interests in entities covered in Gartner research. Gartner's Board of Directors may include senior managers of these firms or funds. Gartner research is produced independently by its research organization without input or influence from these firms, funds or their managers. For further information on the independence and integrity of Gartner research, see "Guiding Principles on Independence and Objectivity" on its website, http://www.gartner.com/technology/about/ombudsman/omb_guide2.jsp.