SOLVING THE SCALABILITY PROBLEM WITH BIG DATA
The amount and variety of data collected is continuing to expand and the need to mine this data is imperative. Those factors – in combination with the expansion of client devices, demand for highly componentized systems, and more – are making centralized computing a thing of the past. This expert e-guide outlines how embracing a more distributed architecture addresses those issues, especially in terms of big data. Learn how the advent of NoSQL options can provide an embarrassment of riches once standardization is the norm.
HOW BIG DATA AND DISTRIBUTED SYSTEMS SOLVE TRADITIONAL SCALABILITY PROBLEMS

It’s rare to see an enterprise that relies solely on centralized computing. But there are nevertheless still many organizations that do keep a tight grip on their internal data center and eschew any more distribution than is absolutely necessary. Sometimes, this is due to investments in existing infrastructure. At other times, it is due to security concerns that arise from a risk averse culture. However, centralization is becoming less and less feasible due to a number of unavoidable factors:

- The number and variety of client devices increases year over year, creating an increasingly complex array of end points to serve

- The amount and variety of data collected continues to expand exponentially with the use of social, mobile, and embedded technology

- The need to mine this data for business insights becomes imperative in
a competitive marketplace

- The requirements of continuous development and deployment create a demand for systems that are highly componentized for greater agility and flexibility (SOA)

- The cost of scaling internally to provide the computing resources to keep up with demand while maintaining acceptable performance levels becomes too high to handle from both an administrative and infrastructure standpoint

- Having a potential single point of failure is unacceptable in an era when decisions are made in real time, loss of access to business data is a disaster, and end users don’t tolerate “downtime”

So how does embracing a more distributed architecture address the aforementioned issues? Different aspects of the distributed computing paradigm resolve different types of performance issues. Here are just a few examples:
PEER PRESSURE IS A GOOD THING

The peer-to-peer distributed computing model ensures uninterrupted uptime and access to applications and data even in the event of partial system failure. Some vendor SLAs guarantee high availability with 99% uptime or higher, a feat which few enterprises can match using centralized computing. Automated failover mechanisms mean end users are often unaware that there is even a problem since communication with the servers is not compromised. What about latency issues? SLAs may also be customized with specific performance metrics for response time and other factors that align with business objectives.

THE SKY IS THE LIMIT

The “virtually” unlimited scalability of cloud computing provides the ability to increase or decrease usage of infrastructure resources on demand. Instant and automated provisioning and de-provisioning of servers and other resources allow enterprises to perform better by ensuring that end user access to applications keeps up with simultaneous, resource intensive demand – even when traffic spikes unexpectedly.
DATA IS A BIG DEAL
The use of distributed systems also has implications for “Big Data”. The advent of NoSQL options provides an opportunity for enterprises to bifurcate their data stream to accept and fully utilize both relational data via SQL DBs and non-relational data with DB options such as MarkLogic and MongoDB. Arnon Rotem-Gal-Oz, Architecture Director at Nice Systems, points out that SQL still has the edge when it comes to reporting functionality, security and manageability. On the other hand, he admits, “If you have scale problems that are hard or expensive to solve with traditional technologies, NoSQL fills these needs in ways that you didn’t have before.”

Implementing native applications that run on thick clients relieves servers of some of their workload and can deliver a faster and more user-friendly experience (assuming there isn’t a need to update data frequently between the client and the server). Using a tiered structure that divides responsibilities among web, application and data servers can permit organizations to outsource any of these processes or layers that can be handled most effectively by a third party vendor. This type of multi-tiered distributed computing can also be used to lessen the burden on internal servers even when deploying applications for thin clients such as mobile devices.
BARGAIN BASEMENT PRICING

Large scale distributed virtualization technology has reached the point where third party data center and cloud providers can squeeze every last drop of processing power out of their CPUs to drive costs down further than ever before. Even an enterprise-class private cloud may reduce overall costs if it is implemented appropriately. The number of vendors in the cloud arena is still growing, leading to more and more competitive pricing arrangements. In addition to lowering costs, relieving the administrative burden from internal IT personnel may free up resources for developing applications that improve performance in other ways.

VERSATILITY IN TECHNOLOGY CHOICES

A distributed architecture is able to serve as an umbrella for many different systems. Hadoop is just one example of a framework that can bring together a broad array of tools such as (according to Apache.org):

- Hadoop Distributed File System that provides high-throughput access to application data
Hadoop YARN for job scheduling and cluster resource management

Hadoop MapReduce for parallel processing of big data

Pig high-level data-flow language for parallel computation

ZooKeeper high performance coordination service for distributed applications

This framework may be of special interest to enterprises since some very bright minds are working on commercialization projects at Yale University right now in concert with Hadapt. Dr. Daniel Abadi believes that “Hadoop is going to make it to the next level. We saw a lot of adoption in 2012. Now it’s about trying to figure out the ‘perfect’ Hadoop use cases. So, building some vertical-specific applications is going to be a pretty big trend for 2013.” Those use cases that increase distributed computing and business performance will be trailblazers to watch.
LACKING NOSQL STANDARDS MORE DANGEROUS THAN PROPRIETARY VENDOR LOCK-IN?

For enterprise architects and software developers interested in exploring a vast array of options for managing application data, the choice in data persistence solutions is no longer limited to the relational world, as the range and depth of NoSQL options has become an embarrassment of riches. MongoDB, Cassandra, CouchBase, Neo4J, VoltDB, ElasticSearch and Redis make up only a short list of the various NoSQL options available. Almost anything you want to do can be done using the various NoSQL database products on the market today. But as any software architect knows, it is not always possible to get everything that is needed from a single database product. As a result, seasoned professionals aren’t afraid to Hadoop together a combination of database solutions to approximate what is really needed. Of course, since there are so many different factors involved with NoSQL, this can rapidly become a challenge. Some of the options and features an application architect needs to consider when choosing a solution for handling persistence include:
How big data and distributed systems solve traditional scalability problems

Lacking NoSQL standards more dangerous than proprietary vendor lock-in?
NOSQL, NO STANDARDS, NO END IN SIGHT

Many enterprises are simply trying to decide what’s most important and are thusly implementing the NoSQL solution that comes closest to delivering what they need right now. Then, they cross their fingers and hope that they don’t have to switch vendors anytime soon. They know deep down that even in an age of less contractual and licensing red tape, portability would be a huge problem. The lock in is a byproduct of the fact that there are no standards.

Ann Kelly and Dan McCreary, coauthors of Making Sense of NoSQL, have plenty to say about this dearth of standardization. That is after all, one of the things that seems to make NoSense in the NoSQL world. Other authors have claimed that the lack of standardization is simply an unavoidable feature of having such a wide variety of solutions. But that’s kind of a cop-out. Kelly points out that all you have to do to uncover the real reason is to follow the money. It’s not in the short term financial interest of venture capitalist firms to support standards. “They’re goal up front is to get a return on their investment. You don’t get that by investing in standards. You can only get a return on your investment if your solution is unique enough for people to adopt it.”

The splintered nature of the current NoSQL market creates an ideal environment to maximize short term profits. Of course, in the long term, it will pay
off to have standards. The few niche players that survive the blood bath and become the standard setters will attract lots of third party developers, helping grow demand for their products.

**THE PUSH TOWARDS STANDARDIZATION**

The idea that there’s simply nothing that can be done isn’t accurate either. McCrea revealed that useful and relevant standards are already available. It’s just that efforts to make them relevant industry-wide have been spotty. “There are standards that NoSQL developers could implement. In fact, there are third parties that are grafting them on. There are two separate efforts now to put XQuery interfaces on MongoDB. That means I could port my data between MarkLogic which is a native XML system and MongoDB and maybe 3 other native XML databases that have good scale characteristics.”

The present may look muddled, but there is hope for the future. As McCrea says, “Standards are beautiful, wonderful things, but you cannot predict when they’re going to be adopted. There are uncontrollable events that suddenly allow them to take off.” As the Sam Cooke song says, change is gonna come. We just don’t know when or from what direction. All it took was Steve Jobs saying, “No Flash on Apple products, ever!” to get the ball rolling on SVG
standards. So, there may be a corresponding flashpoint for NoSQL in the commercial field as well. That will be a glad day. Software architects are waiting with baited breath for a champion to appear, and for standards to be the norm when it comes to working with NoSQL solutions.
How big data and distributed systems solve traditional scalability problems

Lacking NoSQL standards more dangerous than proprietary vendor lock-in?
How big data and distributed systems solve traditional scalability problems

Lacking NoSQL standards more dangerous than proprietary vendor lock-in?