A Total Cost of Ownership Comparison of MongoDB & Oracle

A 10gen White Paper

10gen | The MongoDB Company

A Total Cost of Ownership Comparison of MongoDB & Oracle

Executive Summary

MongoDB is an open source, document-oriented database designed with both scalability and developer agility in mind. MongoDB bridges the gap between key-value stores – which are fast and scalable – and relational databases – which have rich functionality. Instead of storing data in rows and columns as one would with a relational database, MongoDB stores JSON documents with dynamic schemas.

Customers should consider three primary factors when evaluating databases: technological fit, cost and topline implications. MongoDB's flexible and scalable data model, robust feature set and high-performance, high-availability architecture make it suitable for a wide range of database use cases. Given that in many cases relational databases may also be a technological fit, it is helpful to consider the relative costs of each solution when evaluating which database to adopt.

It can be faster and cheaper to develop and deploy applications on MongoDB than on Oracle Database, yielding both bottom-line benefits – lower developer and administrative costs – and topline advantages – it is easier and faster to evolve applications to meet changing business and market conditions.

This white paper seeks to illustrate the business rationale for deploying MongoDB over Oracle. We compare the total cost of ownership (TCO) of MongoDB and Oracle, accounting for upfront and ongoing costs, including software, hardware and personnel. We provide two example scenarios – one smaller and one larger enterprise project – and a model for evaluating database TCO. Customers can use this framework to assess the cost of undertaking projects of various sizes using MongoDB, Oracle or any other database. In our example scenarios, MongoDB Subscriber Edition is over 70% less expensive to build and run than Oracle Database (Enterprise Edition deployed with Oracle Real Application Clusters). Finally, we discuss how technological fit and cost can have topline implications, as well.

Table 1: Summary of Costs Associated with Building and Running a Database

	COST CATEGORIES	DESCRIPTION
UPFRONT COSTS	**Initial Developer Effort**	- Personnel cost - Developer coding required to get application and data store to work together
UPFRONT COSTS	**Initial Administrative Effort**	- Personnel cost - Admin/s to install and configure software, cluster machines, set up sharding, etc.
UPFRONT COSTS	**Software Licenses**	- All software related to the data store itself, as well as management tasks such as clustering, replication and caching
UPFRONT COSTS	**Server Hardware**	- Servers required to run database (excludes storage) - Driven primarily by the number and type of processors and RAM - Lesser costs include enclosures, network connectivity, cabling and power supplies
UPFRONT COSTS	**Storage Hardware**	- Storage required to store the data - Varies depending on whether internal or shared (SAN) storage is used, the amount of storage and whether hard disk drives (HDDs) or solid state drives (SSDs) are used
ONGOING COSTS	**Ongoing Developer Effort**	- Personnel - Coding needed to adapt data store to customer, market and business needs
ONGOING COSTS	**Ongoing Administrative Effort**	- Personnel - Administrative effort required to keep data store healthy and running (e.g., planning and responding to downtime, upgrading software and hardware)
ONGOING COSTS	**Software Maintenance and Support**	- Maintenance: upgrades and bug fixes to software - Support: on-call assistance for troubleshooting technical problems with software
ONGOING COSTS	**Hardware Maintenance and Support**	- Maintenance: upgrades and bug fixes to firmware and any software that may come with hardware - Support: on-call assistance for troubleshooting technical problems with hardware
ONGOING COSTS	**Miscellaneous Deployment Costs**	- Other costs associated with keeping database up and running - Includes cloud/hosting/colocation costs, bandwidth charges, electricity fees, etc. - Generally correlated with the number of servers in use, but vary greatly depending on a variety of factors - Because the choice of MongoDB versus a relational database is not the major driver of these costs, we don't explore these costs in this paper

Cost Categories

To compare the economics of deploying MongoDB and Oracle, we consider the total cost of ownership (TCO) of example applications using these databases. TCO captures both the upfront and ongoing costs associated with building and running a database. It includes both personnel costs (e.g., developer salaries), as well as the cost of hardware, software and support. Table 1 describes the cost categories we consider in this analysis.

The following TCO analysis shows the expected costs of building and deploying one smaller and one larger enterprise application using MongoDB Subscriber Edition and Oracle Database (Enterprise Edition deployed with Oracle Real Application Clusters). Although there are a number of potential deployment topologies, which will vary from application to application, the deployments described in this paper paint the picture of how the economics of these two databases typically stack up relative to one another.

The TCO analysis illustrates how MongoDB's ease of deployment and administration, simpler hardware requirements and open-source licensing can make it significantly more cost-effective than Oracle. On the whole, MongoDB can be over 70% less expensive to build and run than Oracle. Moreover, customers pursuing more projects and/or higher complexity applications may find that the cost savings of deploying MongoDB vs. Oracle are even greater than those depicted here.

TCO for Example Projects

Upfront Costs

INITIAL DEVELOPER EFFORT

Initial developer effort refers to the cost of developer time required to get an application and data store to work together.

For relational databases, initial developer effort includes tasks such as defining the data model, creating an object-relational mapping (ORM) layer and writing the business logic for the application.

MongoDB is designed to be easy to use for modern developers. As a result, it is much more cost-effective to develop with MongoDB than to develop with relational databases. MongoDB derives this major productivity advantage from its document-oriented design and dynamic schemas. The way it stores application data matches both current development technology and current development practices – both of which have advanced significantly since the beginning of the relational database industry 30 years ago.

The reasons behind MongoDB's productivity advantages can be summed up as follows:

» **Ease of Use.** MongoDB supports modern development methodologies, such as the Agile Method, making it easy for developers to iterate quickly and continually over the data model. By contrast, Oracle imposes a strict set of constraints on data model development.

» **Data Model.** With MongoDB, the developer only needs to create the data model in one place: the application. With Oracle, developers need to create and maintain the data model in three places using different interfaces: the application, the database itself and the ORM layer.

» **Data Flexibility.** Unlike Oracle, MongoDB allows developers to easily store polymorphic data, as well as structured, semi-structured and unstructured data all in a single data store.

» **JSON Support.** Storing JSON – the basis of many modern applications – is seamless and requires no translation in MongoDB. With Oracle, developers need to flatten out and transform JSON in order to store it in relational tables, only to have to unflatten and rehydrate it when retrieving it from the database.

» **Cloud Architecture.** MongoDB is well suited to elastic cloud deployments given its scale-out design, whereas deploying Oracle in the cloud can be challenging given the infrastructure requirements of relational databases.

» **Ease of Licensing.** MongoDB licensing is simple; subscriptions are priced on a per server, per year basis. With Oracle, licensing is so complex that it often necessitates that developers wait for administrators to obtain and configure development environments, which can take weeks or even months.

Given the above the TCO model assumes that Oracle requires 2x the initial developer effort of MongoDB. Thus, for the smaller project we assume baseline developer effort of 24 man-months for Oracle and 12 man-months for MongoDB (a 50% reduction); for the larger project we assume of 72 man-months for Oracle and 36 man-months for MongoDB (a 50% reduction). Across both scenarios, we assume a fully-loaded developer salary of $120,000 per year.

INITIAL ADMINISTRATOR EFFORT
Installing and configuring MongoDB is inexpensive and simple. To configure a well-performing MongoDB deployment, an administrator typically needs to consider just one variable – the number of nodes in the cluster; there are only a handful of configuration settings to get up and running. By contrast, Oracle is harder to install and configure. Initial administrator effort can be an intense, multi-week process for Oracle, as an administrator must consider tuning hundreds of variables to get good performance out of the cluster. Most organizations require an Oracle-certified DBA to do this task, or they retain expensive outside consultants to do so.

MongoDB administrators do not need to integrate caching layers or create custom sharding logic to direct queries to the right server node. Rather, both caching and sharding are core capabilities of MongoDB. MongoDB's native support for replica sets makes site-to-site replication simple out of the box. In contrast, enabling and scaling caching, sharding and site-to-site replication often requires substantial effort and custom code with Oracle.

Based on the same logic applied to initial developer effort, the TCO model assumes that MongoDB requires half the initial administrator effort required for Oracle. We assume that Oracle requires 2 man-months of administrator time for the smaller project and 6 man-months for the larger project, whereas MongoDB requires 1 man-month for the smaller project and 3 man-months for the larger project (a 50% reduction). We assume a fully-loaded DBA salary of $120,000 per year across both scenarios.

SOFTWARE LICENSES
MongoDB is an open-source database with a free community version and a commercial subscriber edition; the latter includes support, software upgrades and bug fixes and some additional features. As the commercial edition of MongoDB is priced on an ongoing basis versus a one-time basis (i.e., per server, per year), we capture this cost under Software Support & Maintenance later in this paper.

Oracle licenses are priced on a per-core basis. Because nearly all servers produced in the last five years have between 4 and 24 cores apiece, even a low-end development or test configuration for Oracle can be expensive. Moreover, Oracle Database Standard Edition does not include a number of central capabilities required for modern applications, such as automated failover, memory caching, auto-sharding and clustering. In order to get these features, customers must buy Oracle Database Enterprise Edition (which is more expensive than the Standard Edition) plus Oracle Real Application Clusters (RAC), an add-on application that enables horizontal scaling over multiple servers.

To make the Oracle configurations as analogous as possible to the MongoDB configurations, we use Oracle Database Enterprise Edition ($47,500 per core) plus Oracle RAC pricing ($23,000 per core), for a total of $70,500 per core.[1] *Discounts on Oracle pricing can range from 0% for small deployments to 80% for large deployments. We assume a conservative 50% discount on the list price for the smaller and larger projects. Additionally, we apply an additional 50% discount on top of that to Oracle's core processor licensing factor.*[2] *This amounts to $17,625 per core for both projects.*

SERVER HARDWARE
Typically, MongoDB's server costs are significantly lower than Oracle's for similar workloads. MongoDB is designed to use commodity hardware in scale-out architectures. MongoDB deployments typically use inexpensive, commodity Linux servers that cost as little as $3,000; even a high-performance, low-power system may cost just $4,000 (excluding storage).

By contrast, an Oracle deployment typically uses a large, single server to optimize performance based on its relational architecture.[3] As an alternative to using proprietary scale-up hardware, Oracle does offer a clustering add-on, Oracle Real Application Clusters (RAC), which enables customers to deploy Oracle Database in a scale-out fashion. As previously mentioned, in this paper we model the Oracle deployment based on a RAC configuration to make the Oracle server configuration as analogous as possible to the MongoDB server configurations.

The TCO model assumes the same server hardware for MongoDB and Oracle. For the smaller project, we assume 3 servers, each with 8 cores and 32 GB RAM, at $4,000 per server. For the larger project, we assume 30 servers, each with 8 cores and 32 GB RAM, at $4,000 per server.

STORAGE HARDWARE
MongoDB's horizontal scale-out architecture can significantly reduce storage costs. MongoDB can use the inexpensive local storage used in off-the-shelf database servers, and can make efficient use of solid-state drives (SSDs).

Although Oracle can reduce its storage footprint using compression, Oracle deployments typically require much more expensive storage, as the relational architecture

1 While one can reduce software costs by using only Oracle Database Enterprise Edition and forgoing Oracle RAC, this entails adopting a scale-up (vs. scale-out) architecture, requiring the customer to purchase a more powerful and expensive server. For the sake of keeping the hardware architectures as comparable as possible across the configurations, we use Oracle RAC in this example. See Server Hardware and the associated footnote for more detail.

2 To account for differences in CPU core architecture, Oracle multiplies the number of cores by a core processor licensing factor. This ranges from 0.25 for older Sun SPARC processors, to 0.5 for most AMD and Intel processors, to 1.0 for IBM Power and others.

3 These servers, generally manufactured by Sun/Oracle, IBM, HP, or Fujitsu, scale by adding more processors to a single box, can contain dozens of CPUs or cores apiece, and range from $25,000 to over $200,000 each.

typically requires a single storage model, like a Storage Area Network (SAN), to ensure base levels of availability and performance. SANs range from $25,000 to over $500,000, depending on their capabilities, and thus can significantly increase configuration costs.

For MongoDB, the TCO model accounts for two 1 TB SSDs (1 TB mirrored) per server ($8,000), which translates to $24,000 in the smaller project and $240,000 in the larger project. For Oracle, we assume a 3 TB SAN ($125,000 for 3 TB usable) for the smaller project and a 30 TB SAN for the larger project ($500,000 for 30 TB usable).

Ongoing Costs

ONGOING DEVELOPER EFFORT

The dynamics of ongoing developer effort are similar to those of initial developer effort. With Oracle, the cost of making schema changes is even higher for an in-production database than for a database that has not yet been released. As a result, many companies strictly prohibit changes to databases or limit them to once or twice a year.

With MongoDB, however, it is easy for developers to add database fields and change schemas, resulting in significantly lower costs and allowing developers to adapt applications as business demands evolve.

In the smaller scenario, we assume that MongoDB requires 50% less ongoing developer effort (6 man-months) than Oracle (12 man-months). In the larger scenario, we apply the same logic, assuming 18 man-months of ongoing developer effort for MongoDB and 36 man-months for Oracle. We assume a fully-loaded developer salary of $120,000 per year.

ONGOING ADMINISTRATOR EFFORT

Ongoing administrator effort includes activities that keep the system healthy and running (e.g., upgrading software and hardware, taking backups and recovering from unexpected downtime).

It takes significantly less time and effort to administer MongoDB than it takes to administer Oracle. Administering a MongoDB deployment mainly involves managing Linux settings and the hardware itself; there are only a couple dozen MongoDB settings to understand and manage. MongoDB's native replica set capability makes it easy to perform common administrative tasks like switching out failed hardware and upgrading a server's OS. 10gen customers report that their Linux system administrator groups have no trouble picking up the task of managing MongoDB, because no special skills are required.

The complexity involved in developing for Oracle databases extends to the administrative arena and translates to increased overhead. As data schemas and custom code evolve, the configuration of the database must evolve, too. Moreover, Oracle has thousands of settings, and administering Oracle requires deep technical skills and training. Customers moving from Oracle to MongoDB report that they can slash their administrator costs significantly. One company had a full-time internal Oracle DBA and retained an outside consulting firm. Once they moved to MongoDB, one of the developers easily took on half-time administration of the MongoDB cluster, and the outside consulting firm was no longer needed.

For the smaller project, we assume that Oracle demands 50% of one DBA's time and MongoDB requires 25% of one DBA's time (a 50% reduction). Similarly, for the larger project, we assume that Oracle demands 1.5 full-time DBAs, while MongoDB requires half that (3/4 of a DBA's time). We assume a fully-loaded DBA salary of $120,000 per year.

MAINTENANCE AND SUPPORT

MongoDB subscriptions are priced on a per server, per year basis. MongoDB Subscriber Edition costs $4,000 per server per year (regardless of server size). This includes access to MongoDB support, software upgrades and bug fixes, as well as certain features only available in the for-pay edition.[4]

For Oracle, annual software maintenance and support includes customer support as well as upgrades to the software. It is typically 22% percent of the software license cost and thus is driven by the number of cores, not the number of CPUs or servers. As a result, even for small configurations, the cost of Oracle support dwarfs the cost of MongoDB support, often by orders of magnitude.

Hardware support costs are typically 10%-12% of the hardware purchase price. Given that Oracle typically requires more expensive hardware (e.g., SANs), maintenance and support for Oracle deployments is higher than it is for MongoDB.

For MongoDB, the TCO model assumes software maintenance and support costs of $4,000 per server, per year for the smaller project, and $3,600 per server, per year for the larger project (a 10% discount). The TCO model assumes 22% of license costs for Oracle. The model also assumes hardware maintenance and support costs of 10% of the hardware purchase price for both MongoDB and Oracle.

4 The Subscriber Edition of MongoDB includes SNMP support and a commercial license.

Table 2: Database TCO Analysis Summary

		SMALLER ENTERPRISE PROJECT		LARGER ENTERPRISE PROJECT		COMMENTS
		MongoDB	Oracle	MongoDB	Oracle	
	Configuration Description	**Software:** MongoDB Subscriber Edition **Server Hardware:** 3 servers (8 cores/server) **Storage Hardware:** 3 TB SSD (mirrored)	**Software:** Oracle Database Subscriber Edition & Oracle Real Application Cluster (RAC) **Server Hardware:** 3 servers (8 cores/server) **Storage Hardware:** 3 TB SAN (usable)	**Software:** MongoDB Subscriber Edition **Server Hardware:** 30 servers (8 cores/server) **Storage Hardware:** 30 TB SSD (mirrored)	**Software:** Oracle Database Subscriber Edition & Oracle Real Application Cluster (RAC) **Server Hardware:** 30 servers (8 cores/server) **Storage Hardware:** 30 TB SAN (usable)	
UPFRONT COSTS	Initial Developer Effort	$120,000	$240,000	$360,000	$720,000	**MongoDB:** Assume that MongoDB's ease of use and increased developer agility decreases development time by 2x (see body of paper for more detailed explanation) **Oracle:** Assume baseline of 24 man-months of application development for smaller project and 72 man-months for larger project Assume fully-loaded developer salary of $120,000/yr.
	Initial Administrative Effort	$10,000	$20,000	$30,000	$60,000	**MongoDB:** Using same logic as above, assume MongoDB decreases admin. time by 2x vs. Oracle **Oracle:** Assume baseline of 2-man months of admin. effort for smaller project and 6 man-months for larger project Assume fully-loaded DBA salary of $120,000/yr.
	Software Licenses	$0	$423,000	$0	$4,230,000	**MongoDB:** Cost for MongoDB Subscriber Edition is captured under Software Support & Maintenance (below) **Oracle:** $70,500/RAC core ($47,500 for Oracle DB Enterprise Edition + $23,000 for Oracle RAC), 0.5 Xeon Core License Factor, 50% discount off list price
	Server Hardware	$12,000	$12,000	$120,000	$120,000	**MongoDB and Oracle:** 8-core servers with 32 GB RAM ($4,000/server). 3 servers for smaller project; 30 servers for larger project
	Storage Hardware	$24,000	$125,000	$240,000	$500,000	**MongoDB:** 2, 1TB SSDs (mirrored) per server ($4,000/SSD). 6 SSDs for smaller project; 60 SSDs for larger project **Oracle:** 3 TB SAN (usable) for smaller project ($125,000); 30 TB SAN (usable) for larger project ($500,000)for larger
	Total Upfront Costs	**$166,000**	**$820,000**	**$750,000**	**$5,630,000**	
ANNUAL ONGOING COSTS	Development Effort	$60,000	$120,000	$180,000	$360,000	**MongoDB:** Assume that MongoDB's ease of use and increased developer agility decreases development time by 2x **Oracle:** Assume baseline of 12 man-months of application development for smaller project and 36 man-months for larger project Assume fully-loaded developer salary of $120,000/yr.
	Administration Effort	$30,000	$60,000	$90,000	$180,000	**MongoDB:** Using same logic as above, assume MongoDB decreases admin. time by 2x **Oracle:** Assume smaller project requires 50% of one DBA's time and larger project requires 1.5 full-time DBAs Assume fully-loaded DBA salary of $120,000/yr.
	Software Maintenance and Support	$12,000	$93,060	$108,000	$930,600	**MongoDB:** MongoDB: $4,000/server/yr. for smaller project; $3,600/server/yr. for larger project (a 10% discount) **Oracle:** 22% of license fees
	Server Maintenance and Support	$1,200	$1,200	$12,000	$12,000	**MongoDB and Oracle:** 10% of hardware purchase price
	Storage Maintenance and Support	$2,400	$12,500	$24,000	$50,000	**MongoDB and Oracle:** 10% of hardware purchase price
	Miscellaneous Deployment Costs	Varies Greatly	Varies Greatly	Varies Greatly	Varies Greatly	Not considered -- varies significantly and is assumed to be comparable for MongoDB and Oracle
	Total Ongoing Costs	**$105,600**	**$286,760**	**$414,000**	**$1,532,600**	
	Total Ongoing Costs over 3 Years	**$316,800**	**$860,280**	**$1,242,000**	**$4,597,800**	
	3-Year Nominal TCO	**$482,800**	**$1,680,280**	**$1,992,000**	**$10,227,800**	
	Savings vs. Oracle	**71%**		**81%**		MongoDB enables savings of over 70% vs. Oracle

SUMMARY

Given the assumptions used in this TCO analysis, MongoDB Subscriber Edition is over 70% less expensive to build and run than Oracle (Enterprise Edition deployed with Oracle RAC).

As previously mentioned, although we believe this analysis is representative of the economics of MongoDB vs. Oracle, applications, topologies and costs will vary from use case to use case. The TCO analysis presented here represents two example projects. Customers deploying more applications and/or more complex applications could see even greater cost savings than those shown in this paper; in some cases, the cost disparities may be smaller. We encourage those evaluating different database solutions to use our framework as a starting point for conducting this analysis for themselves.

Topline Implications of Using MongoDB

Beyond tangible cost savings, MongoDB's document-oriented model and flexible schema also afford businesses increased agility and flexibility – providing topline benefit. A business that spins its wheels trying to modify a rigid relational schema to change its application not only wastes money on extra development time, but also suffers the opportunity cost of a slower time-to-market.

Many of the technical and cost-related benefits discussed previously translate to increased time-to-value and time-to-market – topline benefits. For instance, schema flexibility and alignment with the Agile development method enable businesses to adapt their products quickly if customers demand change. The ability to deploy in elastic cloud environments means that businesses can scale technology in line with revenue and customers.

While these benefits can be substantial and far-reaching, they are far more subjective and situation-dependent than any of the costs discussed in this paper. As such, we do not provide even sample quantifications of those benefits here, but we encourage customers to think about what their businesses could achieve if database development and deployment were simpler and more flexible.

Conclusion

The TCO analysis presented in this paper attempts to outline the financial benefits that businesses can realize by adopting MongoDB. Although cost disparities could be smaller or greater depending on several factors, such as the number and complexity of applications being deployed, MongoDB is over 70% less expensive to build and run than Oracle for the example projects shown here. The cost disparity is driven by MongoDB's increased ease of use and developer flexibility, which decreases personnel costs; by MongoDB's use of commodity hardware (storage, in these examples); and by Oracle's substantially higher fees for licensing and support. Furthermore, MongoDB's technical and cost-related benefits translate to topline advantages as well, such as faster time-to-market.

We hope that customers find this framework helpful in evaluating TCO for whatever projects and databases they may be considering.

To learn more about MongoDB and its TCO advantages, or to speak to a sales representative, please email **info@10gen.com.**

10gen | The MongoDB Company

Published by 10gen, Inc.
October 2012