

# MammothDB Checklist

Features	MammothDB
Storage type (Commodity or SAN/NAS)?	Block storage, locally attached storage (LAS) or MapR FS recommended. NAS is not recommended but is supported.
Storage Architecture (Shared everything/shared nothing)?	Shared Nothing
Columnar or not?	Yes
Is scalable beyond single node ?	Yes
Has to support stored procedure?	Not currently, as we are trying to be generic. With Stored procedures you have to support a specific language
Can work on commodity hardware/software?	Yes
Fault tolerance @ Query, session level?	Not supported as focus is on interactive queries (see also below). A failed query needs to be re-run.(note, this is typically more important/applicable to typical "big data" Hadoop-MR queries, rather than Mammoths interactive queries, where a re-run is more efficient.)
Fault tolerance @ Disk, Partition & Node level ?	Replication of partitioned tables is being currently built but is not available in the currently shipped production version. It is expected to be in the product in the beginning of 2016.
ANSI SQL 2003-2008 compliance?	MySQL 5.1.x dialect is fully supported. We plan to extend the supported subset with ordered analytical functions in 2016.
Built-in Query optimizer (PPD, Join reordering....etc)?	Relies on underlying relational engines.
Concurrency (high/low)?	Depends on the workload and the available CPU power. We do one worker thread per core on the nodes, this can probably be increased. There is a limitation in MySQL on the number of concurrent SQL sessions ( <a href="https://dev.mysql.com/doc/refman/5.1/en/too-many-connections.html">https://dev.mysql.com/doc/refman/5.1/en/too-many-connections.html</a> ) which is in the order of 250. We can serve 5k client sessions through connection pooling in the presentation layer.
Supports DML (Insert/Update/Delete/Merge)?	INSERT/DELETE supported now, UPDATE support scheduled for Q3 2015. We recommend data warehouse designs that minimize DELETE/UPDATE, as for DWH/BI INSERT is the preferred architecture.
Supports Join/Union?	Yes both. Joins between large tables require careful design.
OLAP (Can create Star/Snowflake schema) ready ?	Yes. MammothDB design is optimized for star/snowflake schemas and OLAP workloads.
NOSQL support ?	MammothDB supports only tabular storage, however, unstructured data and NOSQL jobs could be run on the same cluster that runs MammothDB using classical Hadoop tools.

ETL Capabilities (Inbuilt or relies on external) ?	Supported through external integrations, using a tool that does pushdown optimization like Oracle Data Integrator is recommended.
Unstructured analytics capability (Can we store unstructured data as well)?	Not natively in MammothDB. Unstructured data can be processed on the same cluster using classical Hadoop tools or third-party SQL-on-Hadoop tools. Results can be stored in MammothDB for interactive analytics.
Core level & node level parallelism ?	One worker thread per core.
Support for transactional (atomicity) ?	In development. Planned for Q1 2016. Transactional throughput is low.
Authorization & Authentication governance (AD/LDAP/Kerberos) ?	No as MySQL 5.1 does not support it. Can relatively easily migrate to a MySQL/MariaDB version that provides pluggable authentication.
ODBC/JDBC connectivity support?	Any client that supports MySQL binary protocol.
Supports .NET, Java, Perl & Shell scripting ?	See above.
Management console ?	There are a number of tools available, e.g. MySQL Workbench, Toad, HeidiSQL.
Backup/Recovery mechanism?	CLI available for backup/restore to NFS/MapR FS. Our recommendation is to use fault tolerant HA block storage.
Licensing type (Proprietary/GNU/GPL) ?	Private
Support during India Business hour?	Yes
Estimated pricing?	3k/node/year – includes basic support. 20% volume discount for more than 10 nodes.
Appliance or not?	Can also be packaged and shipped as an appliance on the SuperMicro Microblade architecture. Currently undergoing testing in our lab.

Additional notes (as raised on the call):

- Q: Use of zookeeper, and asks if our zookeeper will be in conflict with the mapr zookeeper.  
We'll run our own instance of ZK. Planning to remove ZK from our architecture in the future.
- Q: Storing the data in MapR FS itself.  
Yes, through a local POSIX client on the nodes.
- Q: Dremmel or spark based query engines – if a node fails, the query should continue to run (but be slowed)  
That is not an architectural goal as we target interactive workloads. Long-running batch jobs should be offloaded to some SQL-on-Hadoop tool.
- Q: Data centre replication.  
Currently we don't have support for replicating across data centers. Since our storage is append-only, this can be done relatively easily in the software, but we need to discuss the requirement further. Is replication of block storage an option?
- Q: SSRS and Tableau integration.  
Yes both are supported.