MongoDB World 2022

Investor Session: Product Update

Dev Ittycheria  
CEO

Sahir Azam  
CPO

Mark Porter  
CTO

Andrew Davidson  
SVP Product
We are pursuing one of the largest & fastest growing markets in software.
Data Management Software Market, $Bn

2017: $48
2018: $54
2019: $59
2020: $64
2021: $75
2022: $85
2023: $96
2024: $109
2025: $123
2026: $138

Source: IDC
But our market is different than other software markets; it is not monolithic.
The unit of competition in many other software markets is a customer.

Competition tends to be binary - only one competitor wins a customer.
The unit of competition in our market is a *workload*.

To gain share inside an account, we need to win more workloads over time.
Our innovation will enable us to win more workloads.
Winning more workloads

- Make it easier to migrate to MongoDB
- Address even more workload types
- Support new application architectures

Relational Migrator

- Time series
- Search
- Analytics

Serverless
- Edge
First, a bit of a refresher
MongoDB’s foundational technical advantage is the document model.

**Developer Experience:** The document model is flexible and maps to how developers think and code.

**Flexibility:** Documents are a superset of all other data models, allowing us to address the vast majority of operational / transactional use cases.

**Scalable:** Documents put data together in a way that is more performant and efficient and allows almost infinite scalability.
And we kept building, creating a general purpose, mission critical database
The result is the most-loved, general purpose, mission-critical, database that addresses the requirements of modern applications...
And so people build new workloads on MongoDB every day
MongoDB Cumulative Downloads (MM)
But what about existing workloads?
The overwhelming majority of those existing workloads are still on relational technology
Relational technology, invented in the 1970s, is no longer capable of meeting the needs of today’s modern applications.

- Rigid to change + imposes unnecessary constraints for developers
- Doesn’t cope well with unstructured data
- Very difficult to scale + poor at handling distributed data
- Not appropriate for Internet scale workload scaling of 10x or 100x when needed
- Expensive hardware, punitive licensing, cloud lock-in, intrusive audits
Getting off Relational is Hard

How do I get started?
A typical enterprise has hundreds of apps. Which ones are the best candidates for modernization, and which should be done first?

What does a modern schema look like?
Relational schema design is well-understood, but doesn’t offer the best agility or performance. How should enterprise data be modelled in the modern age?

How do I get my data into the new schema?
Data needs to be transformed and migrated, taking into account any performance, security and integrity requirements.

What happens to my old code?
How can we move our applications forward, when the legacy code may no longer be well documented or understood?
We Have Been Helping Customers with Off-Relational Migrations for Years

**Training**
- MongoDB University
- Instructor-Led Courses

**Product**
- Schema Suggestions
- Query Builder
- Connectors

**Services**
- Technical consulting
- End-to-End Application Development

**Partnerships**
- Modernization Toolkit Program
- Systems Integrators
Today we are announcing a step forward in our effort to help customers get off relational - **Relational Migrator**
Relational Migrator is a new tool by MongoDB that helps you bring your relational workloads to MongoDB with confidence.

Visually analyze your relational schema and select tables relevant for migration.

Determine how your relational model should be represented as a MongoDB document model.

Replicate and transform data from Oracle, MySQL, SQL Server or PostgreSQL to MongoDB.
Relational Migrator Has Two Components

**Graphic User Interface (GUI)**
A visual and intuitive web-based experience for mapping relational schemas to the MongoDB document model

**The Data Sync Engine**
Uses the mapping definitions from the GUI to transform and replicate data, either as a snapshot or continuously
Relational Migrator will at first be used by MongoDB team (pre-sales engineers, professional services) to assist customers with migrations, while a self-serve option for customers will be available in 2023+.
If you want to learn more and see a demo, please join my keynote at 2pm
Winning more workloads

- Make it easier to migrate to MongoDB
- Address even more workload types
- Support new application architectures

Relational Migrator

Time series
- Search
- Analytics

Serverless
- Edge
Time Series Use Cases Are Everywhere

- Monitoring physical systems: equipment, machinery, connected devices, the environment, our homes, our bodies
- Asset tracking: vehicles, trucks, containers, pallets
- Financial trading systems: securities, crypto
- Eventing applications: user/customer interaction data
- Business intelligence: Tracking key metrics & health of business
Time Series data has unique requirements that are difficult to satisfy

- Massive ingestion throughput
- Enormous number of sources (for example sensors)
- New data generally most valuable, old data needs to tier out
- Queries need to perform fast on time-slice rollups
- Data can have gaps
These unique requirements have led to a proliferation of *niche* time series-oriented databases that come with all the baggage of yet another system to worry about.
A year ago, we announced *native* support for time series workloads within MongoDB.

Our customers have been using us for time series workloads for years, but we determined that a specialized data type and processing is needed to optimize performance.

By creating a new data collection type, we enable customers to automatically store time series data in a highly optimized format.

New data collection enables best-in-class performance at scale while allowing customers to take full advantage of benefits of our platform.
We meaningfully improved time series collections in the past year

- **5.1**
  - Shardings support
  - Multi-delete support
  - Densification
  - Atlas Online Archive
  - Cardinality handling improvements

- **5.2**
  - Columnar compression
  - Top-n accumulator

- **5.3**
  - Gap filling
  - Geo Near

- **6.0**
  - Geo indexing
  - Expanded secondary indexing
  - Read performance improvements
Winning more workloads

Make it easier to migrate to MongoDB

Address even more workload types

Support new application architectures

Relational Migrator

Time Series

Search

Analytics

Serverless

Edge
As consumers, we know that search is ubiquitous, but search technology goes beyond the classic search box - it powers personalized experiences and brings disparate data sources together.
Building and maintaining search systems is incredibly complex

- **Developer friction**: Different query APIs and drivers for database and search, coordinate schema changes
- **Pay the sync toll**: Requires its own systems and skills. Recovering sync errors can consume 10% of a developer’s time
- **Operational overhead**: More to provision, secure, upgrade, patch, back up, monitor, scale, etc.
Atlas Search is a fundamentally better experience

Avoid paying the “synchronization tax”. Data is automatically and dynamically synced from Atlas Databases to Atlas Search indexes.

Deliver new search capabilities faster. Developers work with a single, unified API across both their database and search operations.

Remove operational heavy-lifting. Automate provisioning, patching, upgrades, scaling, security, and disaster recovery.
We are seeing broad based adoption

<table>
<thead>
<tr>
<th>Grocery retailers</th>
<th>Real estate agents</th>
<th>Stock exchanges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer loyalty and ecommerce promotions</td>
<td>Property search</td>
<td>Credit risk assessment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Professional social networks</th>
<th>Sports car manufacturers</th>
<th>Insurers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Employer reviews and ratings</td>
<td>Merchandise eCommerce catalog &amp; vehicle history</td>
<td>Customer 360 single view</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grocery delivery services</th>
<th>Home fitness companies</th>
<th>Online music tuition sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer and order management</td>
<td>Class instructor content management</td>
<td>Artist and track search</td>
</tr>
</tbody>
</table>
The evolution of Atlas Search

- **JUNE 2020**: General Availability
- **MARCH 2021**: Function scores
- **JULY 2021**: Synonyms
- **NOV 2021**: Facets (preview)
- **JUNE 2022**: Facets (GA)
  - Cross collection search
  - Stored source fields
  - Embedded documents in arrays
Winning more workloads

Make it easier to migrate to MongoDB
Address even more workload types
Support new application architectures

Relational Migrator
Time Series
Serverless
Search
Analytics
Edge
Two important trends are affecting analytics

More use cases require in-app analytics
Applications are performing more complex queries on their data to drive experiences and solve more sophisticated problems.

Businesses need real-time data visibility
The most mission critical and up-to-date data lives in applications and organizations can’t always wait for it to be moved to a data warehouse.
Trend #1: more use cases require in-app analytics

- Personalization
- Fraud & Error Prevention
- Process Optimization
- Preemptive Maintenance
**Situation:** Boxed, an online wholesale retailer for bulk-sized packages, saw a 35x increase in volume as COVID-19 lockdowns spread.

**Challenges:** Boxed collects and analyzes real-time data on orders in and out, inventory, and warehouse management across the US.

**Solution:** With MongoDB, Boxed is able to make just in time adjustments to their business processes by collecting a data in real-time.

---

**Situation:** Amadeus serves hundreds of airlines and needs to provide a personalized experience to help travelers decide on a destination.

**Challenges:** Handling billions of requests each day that are processing complex queries requires instant scale.

**Solution:** The flexibility of MongoDB data model could handle the most demanding data structures and multi-region distribution for scale & high availability to serve queries helping connect travelers with destinations.
Many of MongoDB’s foundational capabilities enable in-app analytics

- Flexible data model designed for modern apps
- Aggregation Framework and Window functions
- Long-Running Snapshot Queries
- Workload Isolation

Many of MongoDB’s foundational capabilities enable in-app analytics.
Continuing to enhance our in-app analytics capabilities

Atlas Analytics Node Tiers
Isolate and scale analytical workloads from your operational workloads

Faster $lookup
Query engine optimizations to improve ‘join’ performance

Column Store Index
Improve the performance of analytical query pattern workloads
In a nutshell, our in-app analytics announcements will allow developers to build apps that leverage analytical queries in the same platform as their operational workloads
Trend #2: Businesses need real-time data visibility

MongoDB is a repository of a company’s most valuable operational data.

Deriving insights from that data is too slow, requiring transforming, loading and only then analyzing the data - this takes days, not minutes.

Constituents who need the data (analysts, data engineers, data scientists) are not MongoDB’s target customers - but they need access to the data.
Creating a faster path to insight

**Atlas Data Federation**
Seamlessly query & aggregate data across data sources

**Atlas SQL Interface**
Easily connect to Atlas from the most popular BI tools

**Atlas Data Lake Storage**
A fully managed, analytical data store
In a nutshell, our data visibility announcements will allow customers to gain *faster insight* into *MongoDB data*.
Winning more workloads

Make it easier to migrate to MongoDB

Address even more workload types

Support new application architectures

Relational Migrator

Time Series

Serverless

Search

Edge

Analytics
Developers keep moving to higher levels of abstraction

- Self-managed Hardware
- Virtualization
- Cloud instances
- Serverless
What exactly is serverless?
Serverless is simplified deployment via instantaneous, smooth scaling

Billing based on infrastructure (e.g., CPU, RAM)
Step-wise scaling
Some operational overhead (e.g., capacity planning, maintenance)

 Billing based on atomic units (e.g., reads/writes)
Instantaneous, smooth scaling
Minimal operational overhead
Here’s one way serverless helps with abstraction - much faster setup

**Dedicated**

1. Select cloud provider
2. Select region
3. Select cluster tier
4. Adjust cluster disk size, IOPS, autoscaling
5. Choose MDB version, backup configuration, advanced settings (e.g., sharding)
6. Assign a name

**Serverless**

1. Select cloud provider
2. Select region
3. Assign a name
MongoDB Atlas isn’t the first serverless database, but it is the most complete.

Traditional serverless databases...
- Suffer from limited query capabilities (typically key/value stores) servicing narrow niche use cases, requiring bolt-on additions
- Cannot seamlessly scale to zero and burst when needed
- Cost grows linearly with usage

Serverless databases in Atlas...
- Bring the full, expressive, flexible, transactional power of MongoDB
- Scale up and down from zero seamlessly
- Bring a customer-friendly usage-based cost curve
Serverless is winning us more workloads today, and will win more in the future

Today
- Application development & testing
- Sparse workloads
- Infrequent workloads (e.g., CRON jobs)
- Workloads with unpredictable traffic

5-10 Years From Now
- Mainstream Applications
Winning more workloads

- Make it easier to migrate to MongoDB
- Address even more workloads
- Support new application architectures

Relational Migrator

Time Series
- Search
- Analytics

Serverless
- Edge
Edge computing use cases are proliferating

Mobile applications

Distributed end points - sensors, retail POS system

Connected frontline workforce: healthcare workers, field technicians

Asset tracking: vehicles, trucks, containers, pallets
But building successful edge applications is much harder than it might appear.

Separate data formats are required for data storage across mobile and web platforms.

Building resilient network code that can handle all the retry and conflict resolution logic is a significant undertaking.

Development time is spent managing the local database, the cloud database, and the mechanism that keeps the two in sync.
Syncs serverless architecture lets devs focus on innovating, not managing.

Atlas Device Sync & Realm relieve developers of undifferentiated work.

Sync handles network loss and recovery for you.

Object-oriented database eliminates the need for data mapping and an ORM.

Simplified data structure

Automatic network handling

Fully-managed

Syncs serverless architecture lets devs focus on innovating, not managing.
Powering a diverse set of use cases

Real-time information
For real-time views into inventory, location, status

Real-time collaboration
For multi-user applications

Always-on / Offline-first
For performance regardless of connectivity
Used by industry leaders
New Announcements for Atlas Device Sync

Partition-based Sync
Sync data between users, devices, and the cloud based on a single field (e.g. store ID or username)

Flexible Sync
Sync data only the data you need with a fine-grained and flexible sync solution (e.g. for a healthcare app, sync data based on user’s role – doctor vs. nurse vs. patient)

Asymmetric Sync
Public Preview for one-way sync of data from devices to Atlas, highly complementary with Time Series and Online Archive

Feb 2021
Today
Today
Winning more workloads

Make it easier to migrate to MongoDB

Address even more workload types

Support new application architectures

Relational Migrator

Time Series

Search

Analytics

Serverless

Edge
But more important than any single feature or functionality is the breadth of our platform.

You can do so much with MongoDB - you don’t need “a tool for every job”
A tool for every job creates a mess
Developers are hamstrung by the **rigidity** of this mess

Operators are overwhelmed by the **fragility** of this mess

Architects are bewildered by the **complexity** of this mess

Executives are frustrated by the **cost** of this mess
That mess is pushing the market in our direction
We will keep winning because we are uniquely positioned to offer broad workload support through a modern developer experience, while enabling global application deployment.
Thank You!