WHITE PAPER

Modern Application Development & Polyglot Persistence

ObjectRocket
a rackspace company

Table of Contents

Introduction 2

Database as a Service 3

Data and Polyglot Persistence 3

Enter ObjectRocket 4

MongoDB 4

Redis 6

Elasticsearch 6

New DBA Roles and Responsibilities 7

Moving Toward the Future of DBaaS 8

Kubernetes 8

Scaling 8

Polyglot Use Case: Untappd 9

Reaping the Benefits of DBaaS 9

Introduction

Nearly every company wants to grow in size, yet many remain unprepared for the challenges that a business in their space and position can expect to face. This problem holds especially true for technological spaces like database management, which can change so quickly that companies experience difficulty trying to keep up with various innovations.

Because most of the popular databases are easy to spin up, small companies often place a single developer or other technically savvy employee in charge of managing their database. As these companies scale up and out, however, they experience throughput bottlenecks, security concerns, and other problems that force them to take on risk.

Additionally, databases generally become more difficult to migrate, back up, and manage as the data set grows. NoSQL databases, in particular, require optimized queries and robust indexing strategies to manage the large amount of disparate data they collect.

Because each database possesses its own strengths and weaknesses, it's becoming more important to leverage different database technologies to address different database needs within an application. For example, a company might use Redis initially because of its scalability and automatic failure detection, and add MongoDB later, after secondary indexing and stricter access rights become requirements. If the company also wants to search, store, and analyze their data in real time, they can add Elasticsearch to their network and take advantage of its full-text search and analytics engine.

To understand the changing database landscape, organizations often find themselves in need of multiple database experts to stay one step ahead of the competition.

ObjectRocket

a rackspace. company

Who is ObjectRocket?

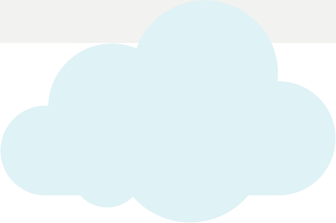
Since 2012, ObjectRocket has managed thousands of open source databases for companies of all sizes. ObjectRocket provides a database platform that's purpose-built for speed, easy scaling, high availability, and data protection along with world-class experts to solve your most challenging database problems. That way, developers can focus on developing amazing applications and features without worrying about database maintenance. When developers can focus on your core business, you can win in your competitive market.

Database as a Service

When a company encounters growth-related problems with their database, it relies on in-house expertise or, if none is available, a database-as-a-service (DBaaS) provider to help solve them.

DBaaS lifts the responsibility of managing the database off the shoulders of the company's IT staff by providing the following services:

- Managing software, including installing, configuring, upgrading, patching, and migrating to any number of servers
- Managing the infrastructure, including the hardware and cloud servers
- Backing up and recovering database information
- Maintaining physical and digital database security by restricting access and limiting permissions
- Planning and preparing for storage and capacity issues
- Maintaining, improving, and monitoring server and database performance
- Responding to sudden growth in storage and processing needs, typically by adding resources as needed to maintain uptime and improve performance
- Troubleshooting issues to ensure optimum uptime and access



Due to the growth of data, as well as the capabilities and availability of affordable cloud computing, **many companies are moving toward cloud DBaaS to reduce the pain of managing their own databases**. A 2017 report on cloud computing trends notes decreasing growth of the private cloud in which enterprises manage their own databases. Additionally, worldwide public cloud spending is predicted to reach $162 billion in 2020, having grown $95 billion during the previous five years. For companies that utilize DBaaS, their employees spend less time managing the database, and spend more time and energy directly on the applications that drive the success of the business.

Data and Polyglot Persistence

The database technology market is also a rapidly changing space that features new use cases and requirements. The ability to capture, store, and leverage data evolves as engineers develop innovations to find the value of captured data.

In particular, data must remain unchanged, or persistent, across time, systems, and memory. To address this challenge, ObjectRocket embraces polyglot persistence and leverages different database technologies to address the myriad database needs within an application. Because each database possesses its own strengths and weaknesses, the days of utilizing a single database per application are nearly gone. **Polyglot persistence helps to ensure that the right tool is used for the right job at the right time.**

For example, relational databases enforce existing relationships. When the data set is small, the SQL JOIN clause is used to discover relationships or to find data from different tables that belong to the same object. As the data set grows, however, this strategy becomes less effective. A graph database might solve the problem of data relationships, but it might also be ineffective regarding the transactions that the relational database management system (RDBMS) provides. In such a scenario, a NoSQL document database like MongoDB might address that issue by storing your unstructured data.

Some companies might remain solely on a SQL database for corporate or regulatory reasons. However, survey data confirms a definite trend away from traditional relational databases and toward more focused and flexible NoSQL databases like MongoDB, Redis, and Elasticsearch. Some database stacks might leverage a SQL or SQL-like database for their primary datastore, for example, but leverage MongoDB for their analytics, or Redis for their cache.

Unfortunately for many growing companies, their database administrator might be familiar with only a single technology. Even if an open-source development company provides database-management services, the following questions remain:

- What if you're using the wrong database?
- What if the needs of your applications change?
- What if another database comes along that fits your needs better? How will you find out about it?
- What if your competition starts using it first?

New datastore technologies present new challenges, and the decisions of which datastores to adopt and when to adopt them must provide a working balance for your business needs.

Another concern surrounding database persistence involves overall developer efficiency. For various reasons, projects often begin by using a single database technology for everything. This approach works as long as the database remains the right one for the job, but many companies neglect to periodically reevaluate their database options. Consequently, they might not be using the appropriate database for their workload.

Sometimes an organization lacks the operational resources to manage multiple technologies, or perhaps an application team moves so quickly that they don't have time to optimize and redesign for a more efficient technology. Still another reason might be a simple, experience-based bias toward a single technology. Regardless, developers must ultimately become creative to make the database technology work for them. They spend energy and development capacity on innovative designs that become unnecessary when a more appropriate technology is used.

For example, many companies initially leverage a single database like MongoDB to manage their user data and product metadata. As their organizations grow, however, this approach becomes difficult to manage, as the flexibility that originally made MongoDB attractive also makes it difficult to manage. Significant support and development resources are then required to resolve issues that might have been nonexistent had a relational database like Postgres been incorporated into their solution.

Enter ObjectRocket

ObjectRocket focuses on open-source data solutions, which require no enterprise licenses, and provides unparalleled expertise about architecture, indexing, and queries on the industry's best cloud database platform. Our customers aren't beholden to contracts, and their data isn't held hostage by proprietary databases. Instead, their data is theirs in the form they need it, and can be migrated to a new database without penalty.

We stand on the core premise of enabling simple and reliable scalability for databases. For example, RocketScale™ monitors the space utilization of each data node and automatically adds data nodes as you need them. After detecting a node hovering above a preset threshold for a set amount of time, the service automatically adds a data node to make certain the cluster doesn't run out of space.

ObjectRocket manages data from so many companies that we've already encountered almost every conceivable problem. Our customers can forget about the backend data source and, instead, can focus on building the specialized business applications that differentiate them from their competition.

Crimson Consulting Group released an analysis showing that contracting with a fully managed DBaaS provides more value than managing databases in-house (June 2017), with money spent on management expertise far exceeding other database costs. Crimson also noted that ObjectRocket lowers data-management costs by "orders of magnitude." Meanwhile, ComputerWeekly states that **ObjectRocket knows how to get the most out of host machines to power demanding workloads**.

> “
>
> ***We use MySQL databases for primary data store and then we supplement with MongoDB, Redis, and Elasticsearch for different pieces of our application. People often ask me why we don't just move all the way to Mongo or move all the way to Redis. The simple answer is that you don't have to put all your eggs in one basket. Use what's going to work best for you.***
>
> ***Greg Avola***
>
> *Founder and CTO, Untappd*

ObjectRocket earns the trust and loyalty of our customers by anticipating their needs, managing and scaling their data infrastructure, and consistently delivering compelling business outcomes. We are experts at optimizing object-relational mapping, and are especially well-respected for our deep knowledge of NoSQL databases like MongoDB, Redis, and Elasticsearch.

MongoDB

According to a 2018 survey, MongoDB is the fourth-most popular database among professional developers. More importantly, it's the most popular NoSQL database as well as the Most Wanted database, regardless of type, among the same respondents. However, managing MongoDB doesn't fit easily into every company's business model. Locating the appropriate expertise can be a difficult chore, for example, and many companies can't afford to hire additional headcount.

ObjectRocket's dedicated MongoDB-management solution provides high security while performing millions of operations per second. Additionally, our experience with MongoDB has taught us plenty of tricks, like storing elements of a blog post in a single collection instead of in separate tables. This practice requires less time than constructing a post in a relational database like MySQL, regardless of where the relevant application is hosted.

Similarly, ObjectRocket employs various strategies for decreasing the size of a dataset, thereby reducing the amount of wasted space in the database and improving the performance. A useful tactic involves aliasing fields to shorten the known field names of everything stored in the document, like using FN to indicate First Name. Another tactic involves using a name store to tokenize any customized attribute field names that require shortening. A name store is a MongoDB document that maps values like Favorite Player to short, unique, and predictable strings, and it can be generated only by using MongoDB's atomic operators.

To reduce the sizes of field names further while continuing to use a flexible schema, ObjectRocket applies the name store token paradigm to various parts of an application. For example, a radio station application might store a customized attribute that is an array of the top 50 performing artists to whom a user listens. Instead of utilizing an array with 50 strings in it, though, we tokenize the radio station names and store an array of 50 integers on the user. As a result, the act of querying users who prefer a certain artist involves a token lookup for the field name as well as a token lookup for the value. Because the translation is cached from value to token, a multi-get is used in the cache layer to maintain a single roundtrip to the cache when translating any number of values.

The following represent some of the more common MongoDB use cases that ObjectRocket supports:

Analytics – An important challenge for many organizations involves the creation of consistently useful customer experiences, and data aggregation represents one of the keys to creating such experiences. Companies collect massive amounts of data about existing and potential customers, and aggregate it with publicly available data to discover customer demographics, preferences, and product interactions. From this disparate data, companies build customer profiles and nurture paths to encourage their customers to purchase additional products.

After a business collects such large quantities of data from different sources with different schemas, tying it all together at such a massive scale is a considerable challenge. However, the flexibility and scalability of MongoDB allows for the aggregation of this data and for the construction of analytical tools to create amazing customer experiences. Additionally, MongoDB's speed allows for dynamic experiences that evolve in real time, based on customer behavior.

Product catalog – Although product catalogs are not new to the evolving digital experience, the data that feeds their interactions is richer and more voluminous than ever. MongoDB provides a tool for storing multiple object types with varying sets of attributes, and the dynamic schema capability lets product documents contain only those attributes that are relevant to that product. As a result, MongoDB no longer requires every product record to contain every possible attribute. Instead, users can make changes to their catalogs quickly and easily, enhancing the experience for developers and customers.

Real-time data integration – To obtain a single source of data, many companies with vast amounts of data spread across their organizations are often forced to expend energy and resources on data ingestion, transformation, and schema changes. In contrast, MongoDB's flexibility and query capabilities simplify the aggregation of data and the creation of tools that increase organizational efficiency. Further, this aggregation can be achieved to provide a single view of their data in real time, and with MongoDB 3.6 or later, developers can use change streams to monitor and act quickly on specific events.

Mobility and scaling – With most mobile application development, companies deal with varying data structures from multiple sources as well as highly dynamic growth. The flexibility and scalability of MongoDB provides a superb database solution for dealing with this type of environment. By using schemas that evolve over time, mobile application developers no longer need to spend their time adjusting the database. Instead, they can focus on developing the customer experience.

Redis

ObjectRocket hosts performant, scalable, and highly available Redis instances in the cloud so that companies can focus on growing their applications. Our software stack implements a variety of features, tools, and APIs that save time, ensure uptime, and make life easier. Every aspect of the stack, from the file system to the OS kernel, has been fine-tuned to increase performance.

ObjectRocket's Redis solution also features a fully redundant infrastructure with automatic failover capabilities. All Redis instances are high-availability (HA) pairs that Sentinel manages for automated and transparent failover. To reduce the possibility of downtime, instances are multiplexed across physically separate systems.

SSL connections, access control lists (ACLs) with IP whitelisting, and an integrated firewall ensure that data is encrypted during transit.

The following represent some of the more common Redis use cases that ObjectRocket supports:

- Session caching – Features persistence, which provides an advantage over other session stores, like Memcached
- Full-page caching (FPC)
- Queues
- Leaderboards and counting – Increments and decrements are calculated in-memory; offers sets as well as sorted sets
- Pub/Sub – Facilitates social network connections and chat systems; triggers scripts based on Pub/Sub events

Our Redis experts also collaborate with customers to help them prepare, manage, and execute their data migrations.

Elasticsearch

Elasticsearch can be used as a datastore or to enhance the database technologies that a business is already using. To help companies focus on developing their applications, ObjectRocket shoulders the responsibility of deploying and managing production-ready Elasticsearch clusters.

ObjectRocket offers multiple versions of Elasticsearch to meet specific customer needs. Each solution includes two dedicated Kibana nodes that run on dedicated containers for performance and redundancy. Additionally, our Elasticsearch clusters include common plugins and dashboards, like Kopf, ElasticHQ, and mapper-attachments. Customers can import data from MongoDB into Elasticsearch for analysis in Kibana.

Every customer instance runs on a minimum of 11 dedicated containers, each of which runs its own Elasticsearch or Kibana process. Further, all clusters feature SSL encryption, ACLs with IP whitelisting, user authentication, and container-based isolation. ObjectRocket also offers encryption at rest as an optional setting for customers who want an extra layer of security.

Every plan is highly available with a redundancy throughout the cluster and hardware environment.

The following represent some of the more common Elasticsearch use cases that ObjectRocket supports:

Logging and log analysis – The ecosystem that has grown around Elasticsearch generates a logging solution that is easy to implement and scale. While many of our customers take advantage of this feature purely for logging, others use it to add logging capabilities to their main use case. From Beats and Logstash to Ingest Nodes, Elasticsearch provides multiple options for collecting and indexing data, regardless of the location in which it resides. Other tools, like Kibana, let you create rich dashboards and analysis, while Curator allows you to set the retention period to Autopilot.

Scraping and combining public data – As with logging data, the Elastic stack features tools to simplify the tasks of collecting and indexing remote data. Also, like most document stores, the lack of a strict schema gives Elasticsearch the flexibility to take in multiple sources of data yet still keep it manageable and searchable. For example, our Twitter connector lets you set up hashtags that you can monitor, and facilitates the collection of tweets with those hashtags, all of which can be analyzed in Kibana.

Full-text search – The application of this feature extends beyond the traditional enterprise or E-commerce search. From security and fraud detection to collaboration and beyond, our customers have demonstrated the power and flexibility of Elasticsearch's search capabilities, including multiple tools to make searching easier. Additionally, Elasticsearch provides its own query domain-specific language (DSL) as well as built-in capabilities for auto-complete, "Did you mean…?" responses, and other features.

Event data and metrics – Elasticsearch operates effectively and efficiently on time-series data like metrics and application events. This area is another one in which the Beats ecosystem lets you collect data for common applications. Regardless of the technologies that you use, Elasticsearch most likely features the components necessary to collect metrics and events out of the box. In the rare event that the appropriate components aren't included, adding the right capability is easy.

Visualizing data – With multiple charting options, a tile service for geo-data, and TimeLion for time-series data, Kibana is a simple yet powerful visualization tool.

ObjectRocket's Elasticsearch clusters are built to scale easily and without downtime. When a customer needs more capacity, adding a data node is literally a click away. When scaling to a larger plan size is required, our dedicated support team quickly and safely migrates the appropriate data.

New DBA Roles and Responsibilities

Regardless of your database preference, ObjectRocket's DBaaS offerings eliminate the need for database administrators (DBAs) to maintain database servers in a "hands-on" fashion. Just as hiring a subcontractor to run wires and install plumbing fixtures frees a general contractor to address higher-level problems, hiring ObjectRocket to manage your database frees your DBA to address the following issues and become a true value-add for the business:

Optimizing performance

Technological companies that want to make things better, faster, and more efficient can task their DBAs with optimizing the database instead of updating servers and troubleshooting user errors. Activities like query-performance tuning, data modeling, and improving indexing strategy dramatically improve the efficiency and performance of a database and the applications that run on it. When your database works faster and accomplishes more work with fewer resources, the service cost to your company is reduced, and you can spend more time contemplating strategies for reducing resource consumption and for streamlining your database.

Managing risk and responding to emergencies

Even the most carefully structured system with the most robust redundancies can fail. When your DBA isn't keeping the system running, he or she can spend additional time focusing on and preparing for potential disasters. Familiarizing oneself with DBaaS systems and their interactions with internal systems helps prepare your DBAs for dealing with challenges as they arise. This approach allows your business to continue operating even when there's a problem, regardless of whether it's related to the data, database, or communication between the DBaaS servers and the clients.

Cybersecurity

Managed services typically provide their own physical and digital security to protect a database from direct incursion. When your business and the managed service have access to the database, though, both entities are vulnerable if the client's security isn't addressed effectively. With ObjectRocket as your DBaaS provider, your DBAs have time to protect data from internal and external threats alike by limiting permissions, establishing rigorous password practices, closing software vulnerabilities, and teaching anti-malware and anti-phishing tactics to the organization.

Planning the architecture and managing the budget

Along with optimizing database performance, a DBA who's working with a DBaaS or managed service can plan how much architecture is hosted in-house and how much is outsourced. DBAs add considerable value by helping to which functions are better handled by in-house teams and which features are better handed off to vendors. Because a company saves money as it organizes and refines the physical and digital architecture of its database, fine-tuning of the system is a continuous process

Data guru

When DBAs no longer need to focus entirely on a database system, they can increase their attention on the data and applications that use it. By familiarizing themselves with the data sources, their life cycles, and their value to the organization, DBAs can keep their databases performing as effectively and efficiently possible. Additionally, this approach yields extra time for DBAs to pursue additional database technologies or to focus on other areas, like retention, reporting, business intelligence, analytics, machine-learning areas, and the latest data science.

Given the large number of companies that offer business-intelligence solutions to help make sense of data, the field of Analytics represents an especially valuable knowledge set. By focusing on the data itself, DBAs capitalize on the knowledge gap between those who use the data and those who understand the manner in which it's collected and organized.

Moving Toward the Future of DBaaS

Because modern applications require multiple, modern databases to support their features and microservices, ObjectRocket is developing a new, Kubernetes-based platform to host and manage any open-source database. Ultimately, this platform will expand our database support beyond MongoDB, Redis, and Elasticsearch to include any database technology on any cloud in any region.

Kubernetes

Kubernetes is a set of open source software that facilitates the deployment, scaling, and management of applications and containers consistently across a collection of hardware. It handles hardware failover automatically, lets you conduct deployments in ways that are easy to manage and construct, and is compatible with a range of tools.

As an abstraction layer, Kubernetes uses rules to automate responses to known, expected events, like when an AWS machine to which ObjectRocket has connected VMs becomes unavailable. Additionally, Kubernetes lets us deploy, provision, and manage different databases on different clouds. If Alibaba opens a data center in South Africa, for example, and our customers want to utilize their services, we can quickly satisfy their expectations.

> ***Note*** Memory, latency, performance, IP limitations, and other factors are established by the Cloud provider.

Kubernetes also lets us assist our customers in the following ways:

- Provide support on multiple clouds, and increasing the number of hosting and connection choices
- Decrease the amount of time required to start up a database
- Provide support for customers who need specific hardware that's not part of a cloud
- Ensure a consistent experience, regardless of the cloud to which they're deployed
- Standardize our support so that the same level of assistance can be expected, regardless of where they're hosted
- Standardize deployments, regardless of the cloud or database type. Because Kubernetes eliminates the need to write custom tooling for each cloud, the ObjectRocket product team is required to learn how to deploy each scenario only one time.
- Address uncommon or unusual implementations. For example, a customer might want a three-node replica set in US-East 1, but they might also conduct business in Europe, southeast Asia, and Australia. Understandably, they might want to set up data nodes for reads where their customers live.

ObjectRocket uses Kubernetes as an abstraction layer for our product team, which doesn't know – and doesn't need to know – how to deploy MongoDB on Alibaba Cloud. Instead, the team requires only a single group of resources dedicated to deploying Kubernetes on clouds. This approach lets our MongoDB experts focus on helping customers develop software that uses MongoDB, and on thinking about the operational concerns that surround MongoDB. Our experts no longer split their time between MongoDB and Alibaba Cloud.

Scaling

The task of scaling a data center is affected by constraints on hardware, networks, and storage devices. Everything exists in ratios, for example, and high-availability clusters are divided into groups of three. Such patterns are easily and often broken, like when a customer with three data nodes attempts to replicate two sets in another location. Deploying new hardware into a data center can take as long as three months.

Kubernetes lets us specify the information to collect – like a specific number of nodes, or the different sizes of disk volumes – and establish remote connections. It also features deployment tools and scalable configurations, both of which can be automated. By using a single, common set of such tools, ObjectRocket Support no longer needs to familiarize themselves with multiple backend tools. Instead, they can directly collect the type of database and instance information that helps us provide our customers with consistent and exceptional support.

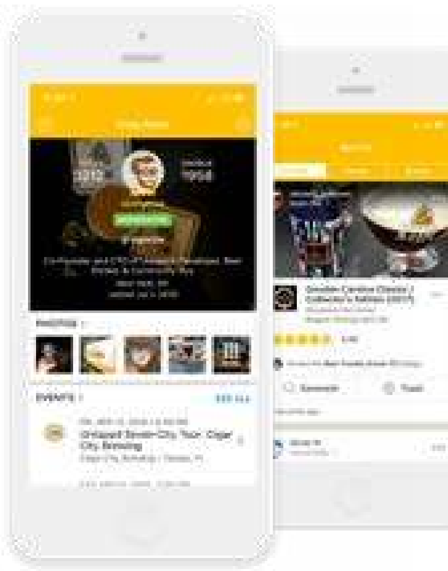
Polyglot Use Case: Untappd

The story of Untappd provides an example use case. Untappd is a social network that lets beer lovers rate, review, and track their favorite beer with friends. Beer enthusiasts receive recommendations while discovering what and where their friends are drinking, and vendors add venue details and current beer menus to make it even easier to explore the world of beer. The rapid transmission of information to beer enthusiasts and vendors is critical, especially given the increasing customer expectations of speed associated with social networks.

Initially, Untappd leveraged MySQL as the primary data store for their queries. This strategy worked in the beginning, but as the company began scaling, the higher traffic volumes resulted in query times that exceeded acceptable durations. By leveraging MongoDB for their heavier queries, they reduced their query times to less than a second, even during periods of high traffic.

As Untappd grew their base even further, they required a more cost-effective solution than MongoDB. Elasticsearch played a vital role in this scenario because it costs approximately one-fourth as much as MongoDB. Elasticsearch also allows users to keep 30 days of activity instead of 10 because thousands of documents are no longer inserted for all of a user's friends or contacts. Instead, Elasticsearch requires a single document per check-in.

Untappd still uses MySQL, but now they possess the flexibility to leverage different technologies for different purposes.



Reaping the Benefits of DBaaS

In keeping with ObjectRocket's commitment to polyglot persistence, our Kubernetes-based platform is poised to host and manage the many modern databases that our customers use to support their features and microservices. For growing companies that are looking for open source data solutions and unparalleled database expertise on the industry's best cloud database platform, ObjectRocket offers the most comprehensive and satisfying solutions.

ObjectRocket
a rackspace company

About ObjectRocket

ObjectRocket's technology and expertise helps businesses build better apps, faster so developers can concentrate on creating applications and features without having to worry about managing databases. We'll migrate your data at no cost and with little-to-no downtime. Our DBAs do all the heavy lifting for you so you can focus on your builds. We provide 24x7x365 expert support and architecture services for MongoDB, Elasticsearch, Redis, and Hadoop instances in data centers across the globe.

Contact ObjectRocket for a more information about our Database as a Service (DBaaS) offerings. We're always happy to provide a free, no-hassle consultation of the leading rocket-fueled platform that is purpose-built for running production database workloads at scale.

LET'S TALK DATA