Statabase

Overview

2024.Q3

Statabase

Introduction

Companies that set out to architect a large system tend to choose an enterprise-level database to serve as the core repository of information storage and retrieval. It makes little sense to attempt to build a proprietary database from scratch. They work well for information-centric systems like shopping sites, social media sites, or messaging apps. However, databases quickly fail when used for real-time applications, even with extremely low tick counts and throughput.

There is no ubiquitous enterprise-level equivalent to a database for real-time applications, so companies are left to try and invent a solution of their own. For instance, a few successful game companies have crafted their own proprietary real-time game engine. However, these systems tend to be highly guarded pieces of intellectual property that provide one advantage or another over competitors.

Countless other companies have not been so fortunate. It is more common for a company to launch a commercial hit only to find out that they are unable to scale their servers to meet demand. The company then morphs from a game developer into a networking scale company, dedicating nearly all of their resources to keep their servers from crashing.

The solutions for scaling a real-time application can take years to iron out, and many companies don't survive. Those that do are burdened with large, complex systems that hobble their products for the remainder of their company's history.

The typical compromise for scaling large systems is to shard them, mitigating the effects of their inability to scale. Most systems today limit games to around 100-250 users per server. It's even more painfully evident in virtual reality experiences, which are typically limited to 15-20 full-fidelity avatars per shard.

Statabases provide a solution for these problems. The statabase is the first and only enterprise-level product that works as a companion to traditional databases. It offers a scalable architecture on which to build real-time, state-based applications that realize 200-500 times more productivity per server than other solutions on the market, yielding energy, time, and cost savings to match.

What a Statabase Does

A statabase maintains a collection of data structures that comprise the state of a system. The central, authoritative server processes client and server actions, which generate events[1] that manipulate those data structures. These modifications are synchronized in real-time[2] across all subscribed clients.

A statabase excels at the types of operations that databases are not very good at, and vice versa. In that regard, they make great companions, but serve opposite functions.

Databases are ideal for storing vast amounts of historical data and performing complex relational queries relatively quickly. However, this perceived speed is no match for a statabase, which typically can access data for real-time processing 1,000 times more efficiently. The tradeoff for this speed is that statabases have no recollection of past events; they are solely focused on the immediate, present state of the system and synchronizing that state to all subscribed clients.

How it Works

In the same way that databases impose a programming schema compatible with data storage and retrieval, statabases also impose a schema, but one that is optimal for processing vast amounts of real-time data, synchronized across vast numbers of clients.

Clients connect to a statabase via persistent socket connections, which are maintained by one or more edge servers that are capable of handling in excess of one hundred thousand concurrent clients.[3] By adding multiple edge servers to meet demand, a system can scale out to process vast numbers of clients.[4] Incoming actions from these clients are serialized and processed in turn by the central server. Results are pushed back to any affected server or client to ensure coherency.

The nature of the data structures, actions, and events within a statabase is a consequence of the specific function being performed, such as playing a game, an algorithm for stock trading, or operating an autonomous vehicle. In each case, the core

[1] The current statabase technology can process approximately 500K events per second, but this is anticipated to reach a rate of 5-10M events per second.
[2] The speed of updates to clients is constrained by the speed of light, plus a small amount of processing time typically in the range of 10-30ms from the time of the event.
[3] The number of connections depends on the nature of the application. Applications with extremely high tick count or throughput may be limited to several thousand clients.
[4] There is no known limitation to the number of edge servers that can be added to a statabase network.

technology remains identical; the changes in the application-defined objects alone are the items that need to be specified.

Relevant events generated by a statabase can be imported into an adjoining database for subsequent data warehouse operations. Further mechanisms also exist to handle redundancy, archiving, recovery and event replay modes as well as disaster recovery.

Why it's Useful

For applications that require synchronization of real-time state across large networks, the use of statabases can yield code bases that are hundreds of times smaller than existing approaches, and which can be easily maintained by a few programmers when they may have previously required hundreds. Statabases typically reduce the compute power required to process and synchronize real-time data by a factor of 200-500 versus other existing technologies, saving enormous amounts of energy and money.

Historically, building large, scalable real-time systems has yielded solutions that consume in excess of 99% of the available compute resources for networking and synchronizing data. As a result, companies devote proportional amounts of time and effort building complicated networks, rather than improving their core application.

In the same way that enterprise-level databases free companies from having to create their own storage and retrieval systems, statabases free companies from having to create their own real-time state networks, allowing them to focus on their core product.

The task of defining the object tree, along with the actions and events that operate on it is akin to defining a set of tables or schema and stored procedures within a database. Since this constitutes only a small percentage of most systems, the time and effort required to build a statabase applications is minimized. This means solutions for new applications can be written, tested, and debugged by a team of only a few expert coders in a very short period of time.

How RP1 Uses Statabases

RP1 uses statabases to synchronize all of the foundational real-time activity of its 3D virtual ecosystem. This includes updating connected users with up to 1K high-fidelity[5] avatars along with server-side mixed spatial audio for any and all avatars within hearing

[5] 6 DOF, head/arm/hand/finger movement plus facial features without predefined animations.

distance. Because of the extremely high tick rate of the system[6] and the extremely high throughput,[7] RP1 requires one edge server for every 4-6K connected clients.

The current configuration for RP1 utilizes a network of 33 statabase servers that can support approximately 100K concurrent avatars in a single shard. This is not the upper limitation of the network, but merely a limitation of our current hardware budget. This compares favorably to other known systems that are limited to 16 avatars on one server with no possibility for expansion across multiple servers.

Case Study

Online poker is an especially good case study due to its unique properties. Poker is a real-time game that only requires a tick count of 4 updates per second and throughput of approximately 1.5 Kbps per connected user.[8] Due to the nature of the game, it is important that all players be able to play in one shard.

In its heyday, there were hundreds of poker sites running on a few dozen proprietary poker systems. PokerStars was the most popular, and after more than a decade of development and improvements, it required over 800 servers and more than a dozen EMC^2 refrigerator-sized databases to support 100,000 concurrent users.[9] Another early leader was Party Poker, which required more than twice as many servers as PokerStars to accomplish the same feat.

> *A statabase can handle the same 100,000 concurrent users playing poker on approximately one-tenth of one server.*

We know from experience that the amount of compute time necessary to play the game of poker for 100,000 concurrent users for one year is approximately one core-day. Meaning, it takes just one standard CPU core 24 hours of compute time throughout the course of the year to process poker related activity. Less than 0.00003% of the compute resources in these systems are utilized to play the game this system was designed to play!

This begs the question: if they're not playing poker, what are the other 10,000-20,000 CPU cores doing? The answer: predominantly trying to coordinate network activity and synchronize state between all of the servers and their respective clients.

---

[6] RP1 operates at 64 ticks per second.
[7] RP1 averages approximately 1-2 Mbps per client in crowded areas.
[8] For comparison, a 3D virtual world with full fidelity avatars requires a tick count of 64 updates per second and approximately 1-2 Mbps per connected user.
[9] https://www.youtube.com/watch?v=Tvmk51WkTa0

PokerStars and Party Poker are among the very few successful sites that managed to figure out how to scale their system. Countless others never made it that far, having succumbed to an inability to scale their systems. Typical systems couldn't handle more than a few hundred concurrent users. And those that managed to climb into the thousands had architectural issues that crippled their systems in one way or another.

Plenty of sites were unable to launch due to technical issues; the failure rate of proprietary real-time poker systems is estimated to be in excess of 90%. The cost of propping up a new real-time poker system, including development costs and hardware is roughly $10-30M and can take anywhere from 2-4 years. Ongoing operation costs of a data center run in the millions of dollars per year.

> *Using a statabase, the same poker system can be built by 2 programmers in roughly 6 months for a cost of around $200K. The ongoing operational costs of the data center are approximately $10K per year.*

The Future of Statabases

The advent of the metaverse will usher in an unprecedented wave of real-time services that will require enterprise-level scale. With current technologies, this would necessitate the deployment of an equally unprecedented number of servers and human resources, since real-time services have historically demonstrated an uncanny resistance to scaling well. Statabases can reduce this requirement by a factor of 200-500, saving proportional amounts of energy, time, and money.

We can envision a not-too-distant future in which every company with a database department bifurcates into two departments: one that manages its historical software needs using databases, and one that manages its real-time software needs using statabases.

This will open up new business growth opportunities in cloud computing, web, and metaverse services with unparalleled profit margins due to the efficiencies that statabases offer. In some cases, a thousand servers can be replaced with one. The human resources needed to develop and maintain these systems can also be reduced.

The value for a company to migrate real-time products to a statabase is highly compelling, and undoubtedly a very lucrative offering for a leading software/technology company looking for a new evergreen channel of business.

Learn More

This document merely provides an introduction to and overview of statabase technology. For a deeper understanding of statabases and the role it plays in the broader scheme of the metaverse, along with additional information about the Metaversal Model Foundation, metaverse browsers, and spatial fabrics, download our white paper, *The Blueprint for the Open Metaverse*.

rp1.com/whitepaper

About RP1

RP1 is pioneering the next evolution of the internet with the first prototype metaverse browser that combines open standards and protocols with a proprietary software-based approach that scales to the world's population.

The metaverse browser allows 3D immersive applications to be delivered on-demand to any device, similar to how we consume web content today. RP1's spatial fabric makes it possible to navigate a 1:1 digital twin of our solar system and an entire universe from any device (PC, mobile, AR/VR) for business, education, and entertainment.

A global open demo of RP1 is coming soon at RP1.com. Contact us directly if you would like to see a demo privately.

About Metaversal Corporation

Metaversal is the world's leading technology company solving for the next generation of the spatial internet. Our mission is to build open software that connects people, services, and devices in a single, persistent metaverse.

We introduced RP1, the first prototype metaverse browser, MVMF, the first real-time API solution, and the statabase, a revolutionary network server architecture enabling unsurpassed vertical and horizontal scale. With open standards and services, RP1 will allow anyone to create and publish fully immersive XR experiences in the metaverse.

RP1.com rp1.com/discord @rp1browser