E-Book

Why Your Game Needs Real-Time Leaderboards

redis

Leaderboards are an essential part of any game with multiple players—and have been since the early days of video games. Take *Space Invaders*. It holds legendary status for any number of reasons related to gameplay, but a big part of its popularity (and staying power) came from its introduction of competitive high scores to arcades back in 1978. Suddenly given a way to gauge their skill against others, players came back again and again to take their shot at the high score crown.

That core competitive concept is still at the heart of leaderboards today. They're an essential part of creating an engaging multiplayer experience that motivates players to see just how high they can get. Players stay engaged and active, which, in turn, means they contribute more revenue to your game.

But we've come a long way from a simple list of high scores in each arcade cabinet. Today's games have thousands of leaderboards for different events, regions, teams, and more. Each player profile on those leaderboards incorporates a host of different data types and variables. And, most importantly, players expect those leaderboards to be instantaneously updated every time they check their ranking—no matter where they're logging in from.

It's a lot of data. And you need to deliver all of it at the performance players expect without spending so much time on this one feature that you neglect the rest of the game. The only way you're going to do that is with a high-speed, real-time database, one that natively supports data models ideally suited for leaderboards.

What does real-time mean?

Research into human response indicates applications have roughly 100 milliseconds (ms)—one third of the time it takes to blink—before users feel like they’re waiting for a response. To be considered real-time, a leaderboard needs to send a request, have it processed, receive the response, and present it to the player in less than 100ms.

Games (and players!) ask more of leaderboards than ever

Leaderboards have gotten more sophisticated

Leaderboards tap into that competitive spirit and incentivize people to try that much harder and play that extra bit more. But they can also be demoralizing. Sure, you might spend an extra 30 minutes (or make an in-game purchase) to jump from #23 to #19 and get that sweet Top 20 reward, but you might just as easily give up if you're #15,375,892 out of 20 million.

Effective gamification requires attainable goals and the ability to have a sense of achievement and accomplishment. If the competitive field is too big, there's a high potential for player apathy, which usually results in players deleting your game. But if you focus the field to a manageable level, players are more engaged and stay an active user that much longer. In the 80s, each high score list was unique to a specific arcade game, physically dividing leaderboards by location to create segmentation. While this narrowed the field, it also meant high score lists dominated by the local exceptional players, which made it easy for more casual players to walk away. Today, leaderboards are much more sophisticated—and personalized.

Modern multiplayer games create leaderboards for different leagues, player rankings, locations, factions, events, and more. A racing game, for example, might have a separate leaderboard for each player level range (1-5, 6-10, 11-15, etc.), then break that leaderboard down by track, then further break that leaderboard down by car. As a result, a player can see how they rank against players with a similar level of upgrades on the same track with the same base car model. Instead of competing against a field of millions (some of whom they never have a chance of beating), they're only competing against a couple hundred people—and on a more equal playing field. That level playing field makes them more engaged and keeps them playing.

But maintaining that level playing field can be more complicated than just personalization. Cheating is, sadly, a very real concern with multiplayer games, and letting cheaters dominate the rankings is a good way to lose players fast. To prevent this, leaderboards often have additional cheating analytics engines or checks built into their ranking systems. These help keep the game fair, but also increase the complexity of the leaderboard services.

All that sophistication and personalization requires complex architectures of services and technologies. And through them all runs a massive amount of data as leaderboard applications process millions of factors for millions of users. It creates a huge demand on the database (and the database team) behind the leaderboards, but personalization complexity isn't the only demand on leaderboards.

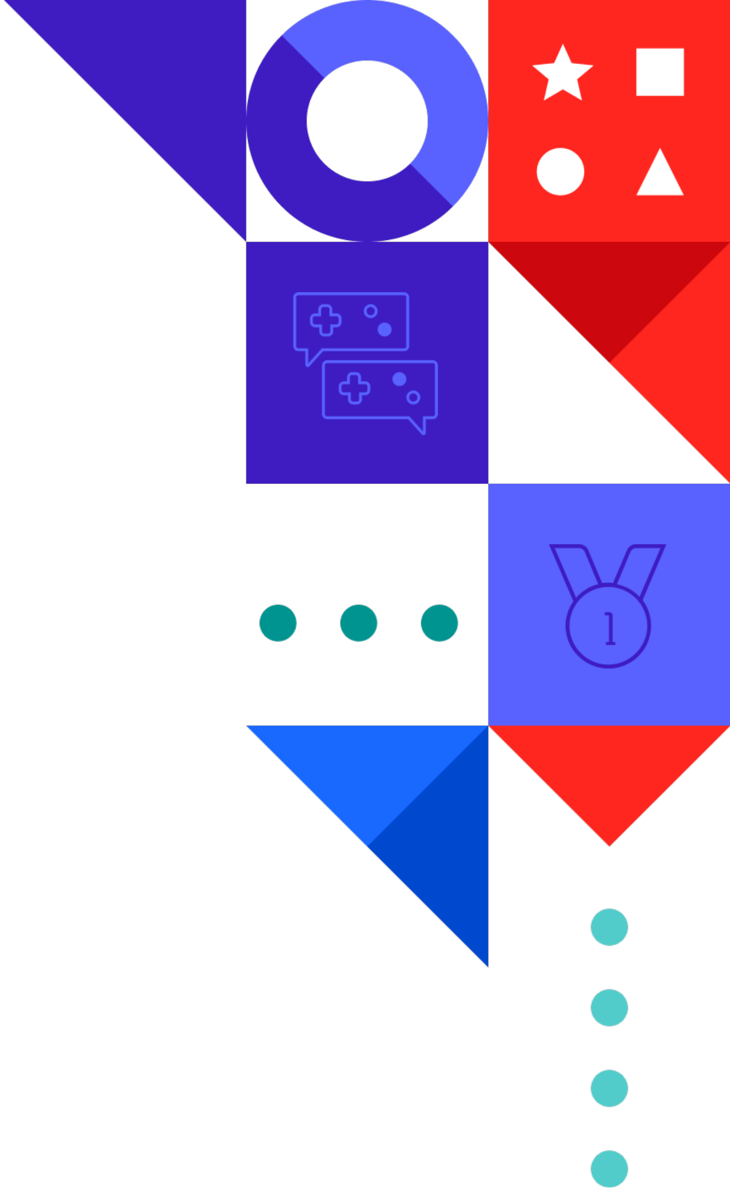
Players expect fast, up-to-date leaderboards tailored to them

Sophisticated, personalized leaderboards are now the norm. Over the last decade or so, they've become a crucial part of more and more games. As a result, players are so used to seeing leaderboards tailored to their playing experience that it's gone from a nice-to-have to expected behavior.

During that same time, speeds in every part of games have increased. Players expect all of their gaming experiences to feel instantaneous and to be constantly updated—and that includes leaderboards. The idea of waiting a full second to see your ranking is laughable, especially when leaderboards have time limits, like with tournaments. Players will be competing up to the last second, and they expect the leaderboard to respond accurately in real time so they know whether to make that last push.

The result? Players expect real-time responsiveness of massively complicated leaderboards applications. This is a tall enough order, but there's one more wrinkle: player location. Games have millions of users logging in from around the globe, and all of them expect real-time leaderboards no matter where they are. They don't care that the person ranked right above them is located thousands of miles away. If they're going to spend in-game resources (or purchase in-game currency) to fight one more battle, play one more puzzle, or run one more race before the timer runs out, they expect to see the exact amount of points they need to get the higher ranking.

All this adds up to a pretty tall order for leaderboards. The application needs to be sophisticated enough for personalization, fast enough for real-time responsiveness, optimized for leaderboards data models, and distributed enough to be available everywhere your players are.



Data technology requirements to pull off real-time leaderboards

How do these demands translate into technical requirements for the database behind the leaderboard? After all, it's not enough to say the leaderboards have to be fast. If it's going to achieve real-time speed, a leaderboard service requires a database that meets these specific needs:

- **Ideal leaderboard data models** – One-size-fits-all data models can struggle to meet real-time demands, forcing database teams to spend a lot of time and effort optimizing the database. The right database should natively support data models ideally suited for real-time leaderboards.

- **High database concurrency** – Millions of scores, rankings, and records need to be updated and queried concurrently. The leaderboard databases have to be able to handle concurrent operations—and be able to intelligently handle concurrent write conflicts.

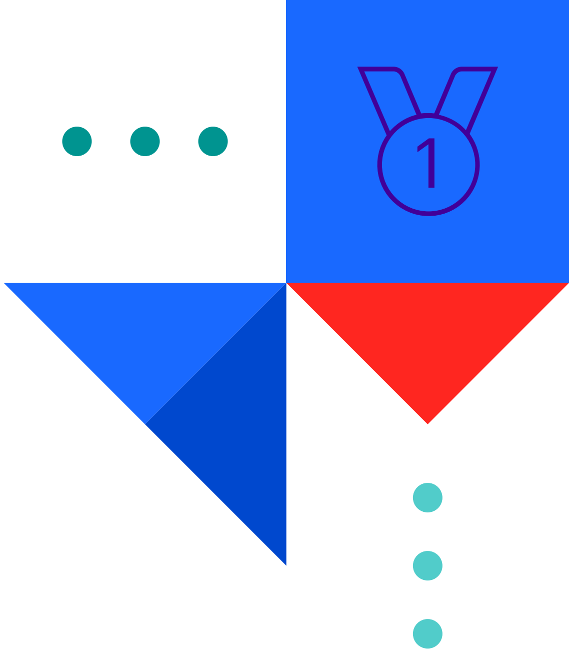
- **High throughput** – Read and write throughput must be high enough to handle millions of write and read requests. This throughput also needs to be able to respond to peaks caused by player surges, like during the last 5 minutes of a tournament.

- **Low latency** – A certain amount of time will always be taken up by data transfer and processing, which means your database needs to take up as little latency as possible for leaderboards to be served to players in real time.

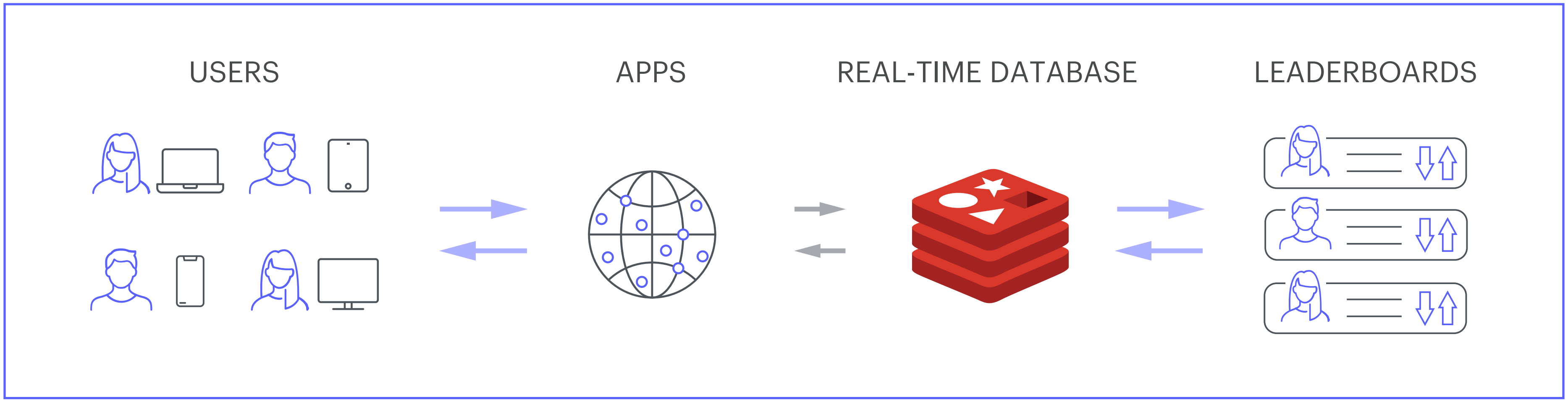
- **Scalability with consistent performance** – Even if a database can deliver the data models, concurrency, throughput, and low latency required, it's not going to truly support leaderboards if it can't scale. The database needs to seamlessly scale up (and down) without any drop in performance.

- **Flexible deployability** – Your infrastructure is going to be determined by a large number of requirements—more than just your leaderboards. To integrate as seamlessly as possible, the database behind your leaderboards needs to be deployable in any environment.

- **Global synchronization** – Millions of players will access the same data across the globe, and they expect that data to be consistent. The local cached leaderboards need to seamlessly replicate and synchronize to provide an up-to-date global leaderboard with both local latency and consistent data.



Data is the heart of any leaderboard. Your leaderboards are only going to respond as fast as your data. If you want to meet the high expectations of your game and your players, then your leaderboard database needs to meet these technology requirements.



USERS

APPS

REAL-TIME DATABASE

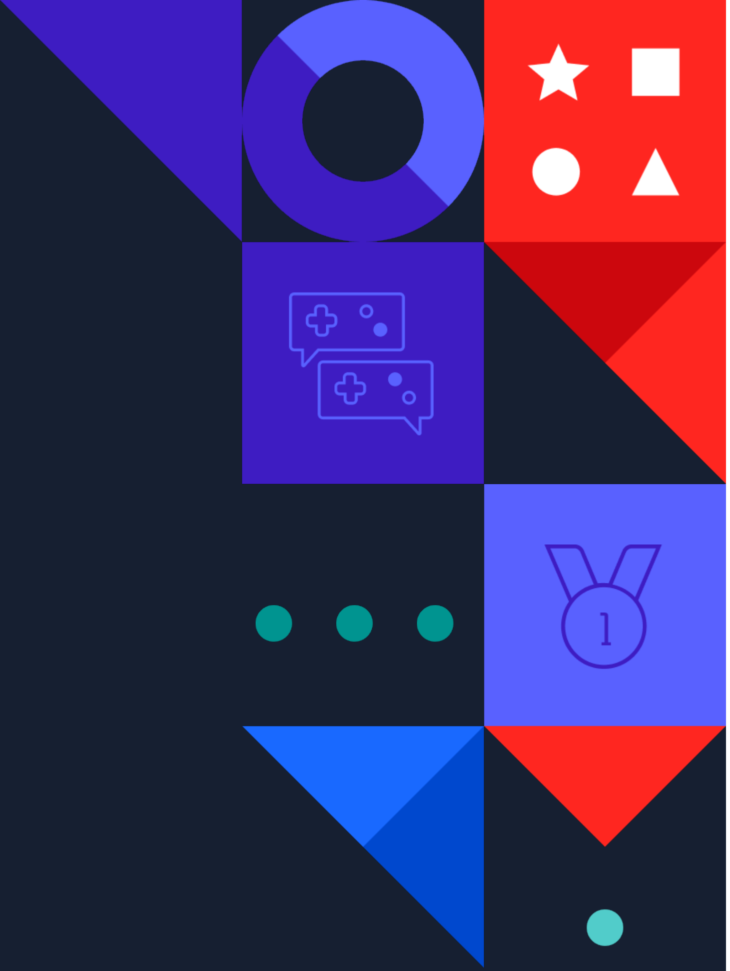
LEADERBOARDS

Real-time data demands a real-time database

Remember that definition of real-time responsiveness from the beginning? If your leaderboard is going to respond fast enough for players to feel no latency, then the whole process from request to processing to response to serving the data to players has to happen in less than 100ms.

Unfortunately, the data still has to travel to the servers, then be processed by the servers. Network time can easily take up 50ms. Server and infrastructure time can take up another 50ms. Which leaves somewhere between 0ms and 1ms for the database to respond if the leaderboard is going to have a real-time response.

And that's where a real-time database comes in. A real-time database is a database with <1ms latency. Redis, for example, is an in-memory NoSQL database. By storing data in the memory instead of on disks, and in a non-relational database instead of a slower relational database, Redis is able to achieve <1ms latency. This makes it a perfect candidate for a leaderboard database.



Part of that real-time speed and reliability also comes from matching the right data model to the use case. Real-time leaderboards sort and arrange data in sets. Every time the data is pulled from one data structure and converted to another, it takes time—and creates a potential breakdown point. When you use the right data model, you remove unnecessary conversions, increasing speed and reliability. As a bonus, the increased simplicity also makes it faster for developers to implement and maintain, so they can focus their time on other features. Sorted Sets, a data type natively supported by Redis, is a perfect example. Like the name suggests, it stores data in sets where data is pulled in order instead of being ordered afterwards. Not only does this make it faster to get ranges of data already sorted by scores, but it also cuts down on developer effort by making it simple to integrate the leaderboard data with the database.

All that being said, to truly be real time, a database needs to deliver that real-time speed everywhere, at any time. It's not enough for real-time responsiveness in just a few locations. Or when there's only a small number of players. Every player needs to get a leaderboard response in less than 100ms, no matter how many players are online—or where they are.

So in order to truly be a real-time database, the database needs the scalability and availability to provide local latency around the globe. To accomplish this, it should include features like native support for data models ideally suited to leaderboards, geo-duplication, automated resharding for scaling, 99.999% availability, and the ability to deploy in any environment where you need it.

Conclusion

Real-time leaderboards are essential for any modern multiplayer game. They keep players engaged, which increases playing time, DAUs, MAUs, and, in turn, revenue. Make sure your leaderboards have the power they need to give players a real-time response with a real-time database.

Find out other ways a real-time database can level up your game and increase player engagement. Go to redis.com/gaming or download the Level Up Your Gametech with a Real-time Database white paper.

About Redis

Data is the lifeline of every business, and Redis helps organizations reimagine how fast they can process, analyze, make predictions, and take action on the data they generate. Redis provides a competitive edge to any business by delivering open source and enterprise-grade data platforms to power applications that drive real-time experiences at any scale. Developers rely on Redis to build performance, scalability, reliability, and security into their applications.

Born in the cloud-native era, Redis uniquely enables users to unify data across multi-cloud, hybrid, and global applications to maximize business potential. Learn how Redis can give you this edge at redis.com.