Cloudflare Workers for Gaming

Before seeing any specific worker verticals track real development time, it's essential we understand where infrastructure frustrations in gaming diverge from those of web applications, and where our solutions can leave game developers with lasting value. To gain a grasp for the specifics of these issues, we can look to three areas in the gaming industry:

1. Service outage postmortems - understand current failure modes for gaming technologies
2. Game developer pains - find out what makes building game infrastructure uniquely difficult
3. Emerging trends in gaming - project which technologies will become important to developers in the future

A holistic approach that synthesizes data from where game tech has failed in the past, what currently makes game tech difficult, and where the most important challenges for the game tech of the future lie would not only allow us to target game developer pain points, but provide real business value and ensure Cloudflare's relevance in gaming for years to come.

Product Analysis

Currently, I believe serverless architectures present attractive options for small and mid sized game developers. Although developers of large, triple A titles may find the FaaS model interesting, moving game deployments around is nontrivial- likely even infeasible for games with particularly large player bases. Instead, we should focus our efforts around the segment of new mid sized game developers looking to rapidly scale. Doing so has the potential to maximize both the likelihood of Cloudflare adoption and the value our workers provide customers with.

Cloudflare Workers can provide game developers with three key advantages. First, workers reduce latency by performing computation on the edge- geographically closer to most users than the average game server. As network speeds increase dramatically over the next decade, this may prove invaluable for many modern real-time games, keeping connection loss and action delays to a minimum. Second, game developers no longer need to pay for unused capacity. Considering gaming manages relatively volatile compute workloads, this has the potential to massively cut costs for devs, especially as user volume grows. Third, workers address key infrastructure pains around load balancing, scaling, and deployment times, making life easier for developers.

Product Additions

While the advantages above are indeed relevant to game development, they generalize to benefits achievable with other distributed computing platforms, and product additions that cater more specifically to customers in gaming could greatly incentivize game developers to make the jump to Cloudflare tech. Load-balancing and scaling are problems popular cloud providers have robust solutions for, and global infrastructure, while not as omnipresent as Cloudflare's network, does exist in good-enough force on alternative platforms. I believe two additions could prove useful to make Cloudflare Workers particularly attractive for gaming:

1. Game platform component ecosystem - provide libraries for common backend service templates in gaming
2. Satellite workers - offload non-critical isolates to idle game clients during peak utilization events

I believe providing libraries with fully featured, extensible backend services for gaming (matchmaking, player positioning, state management, etc.) to game devs as an infrastructure provider has the unique potential to shine in gaming, as game code is intertwined with the hardware it runs on in ways much other software is free to ignore. Not only do games need to be extremely performant, but they need to be highly secure, particularly from the very things Cloudflare has already largely solved. Already optimized, secured modules that address elements common to most online multiplayer games could save game developers loads of time building with workers, and promote adoption at early stages of game development.

Another addition (perhaps more long-term) involves distributing workers themselves with game clients, using client devices to process non-critical workloads for each other. As more and more of the world gets on the internet and gets access to games (and the computers to run them), I believe user churn rates between large titles will increase, and massive infrastructures will need to be spun up and down in relatively short order as a result. To handle workloads that scale directly with user volumes, having client devices mitigate some of the computation work during burst events in a reliable fashion appears an eventuality, and getting started on the foundation for that early couldn't hurt.

Iteration and Improvement

Before releasing this vertical, there are a couple key steps we can take to ensure we're addressing the right concerns in ways that game developers can adopt with little hesitation. First, we should build a game on workers, internally. I believe doing so can accomplish two things at once, identifying bugs/areas of pain for game developers that may eventually use Cloudflare tech, and creating resources that could later on be used to demo our technologies to potential clients. I'm not a game developer, and I'm sure there are many on the Cloudflare workers team that aren't too, so this would be a great way to learn about the many design patterns and infrastructure problems of gaming, while establishing relationships with customers.

Second, we should run trial programs with teams beginning game development at newer studios, providing them with any resources developed during the internal game development process above, supporting their transition to Cloudflare workers, and building in important feature requests. Doing so would not only cement Cloudflare's name in gaming, but provide case studies that could very well provide some larger, mid-sized game studios the extra bit of confidence required to convert to Cloudflare tech.

Product Assessment and Risks

While there are plenty of different KPI's we could set around latency, development pace, costs saved, gamer satisfaction, and uptime, we can call workers for gaming a success if it's an option game developers are willing to *migrate* to. Converting deployment tech is hard- as one of the first few engineers to investigate the (at the time, shocking) prospect of moving real-time advertising technologies to Kubernetes, I know this all too well. If games are not simply built on top of workers for gaming, but moved to workers for gaming, we're on the right track.

Perhaps the greatest risk I can imagine workers for gaming facing involves not actually developing workers *for gaming*. There are a plethora of fascinating problems in distributed computing to go after, some of which even carry substantial business value, but without actually answering to the needs of game developers, there's no real reason for creators to adopt Cloudflare workers over some other platforms. Other risks do exist as well, including technical limitations surrounding storage capacities, developer movement away from workers while scaling up, and recent in-house implementations of the same core technologies.