

Unfortunately in terms of technologies I am mainly familiar with the AWS stack, so many of my technology choices will be AWS.

Load Balancer – Load balancing is a common technique for high-throughput services to reduce latency, maximize throughput, better utilizing resources, etc. I am most familiar with using Apollo for this, however, since Apollo is not publicly available, I believe Nginx is also suitable for the task. Additionally, our service should be deployed in multiple regions in case an entire server/server room goes down.

Microservice architecture – Another common technique to make it easy to deploy changes and isolate failures. Also makes it very easy to spin up new instances. Between this and the load balancer, it should not impact us at all if a single machine goes offline.

Distributed messaging system – Similar to the load balancer, scaling horizontally to maximize throughput/minimize latency. Kafka is suitable for this task.

NoSQL database – Unfortunately I am most familiar with Dynamo, which would not be suitable for this service due to the high quantity of data. I believe something like Cassandra should be more suitable due to its high scalability and distributed properties. We can easily use master-slave replication here since one service will only write to the database and the consuming service will only read.