Talk: "Big Data and High Performance Computing Solutions in the AWS" (marketing perspective)

?https://www.youtube.com/watch?v=naCUrtd6XJE

Talk: Scaling to 10 million users

Fun Fact: In 2014, the amount of capacity added daily on AWS exceeds the capacity to operate Amazon?s $7billion business in 2014.

Stages of growing:

- Initial: single machine (e.g., EC2 instance) running web+db, with DNS routed by AWS Route53.
- Users>100: Split db off into a db service instance (e.g., using Amazon Relational Data Store (RDS)), or another vm instance dedicated to the db.
- Users>1000: Replicate machine and db instance. Now we have two pairs of web+app and db. Let the dbs coordinate using replication. Balance between them using Elastic Load Balancer. Place each pair in different ?availability zones? so that a data center failure in one of the zones affects only one web+app and db pair?your system is still functional in the other zone.
- Users>10k: Add even more pairs. Consider more availability zones.
- Beyond: Move static content to S3 and serve using cloudfront for edge CDN service. Use elastic cache and/or dynamodb to cache state and reduce traffic to db instances.

Amazon autoscaling can use metrics from cloudwatch to make decisions to add more web nodes or db instances as load changes.

Use service-oriented architecture: this facilitates making your components stateless, replaceable, and scalable. Don?t let components talk directly to each other?use indirection so that either side of the communication can fail without affecting the other.

Overall advice: split processing into pieces that are loosely-coupled and stateless as is possible.

Other notes:

Spatial libs for AWS CloudSearch? and DynamoDB seem interesting. They are focused on really low latencies (10-100ms). Cloud Search has those latencies when doing lat/lon area searches.