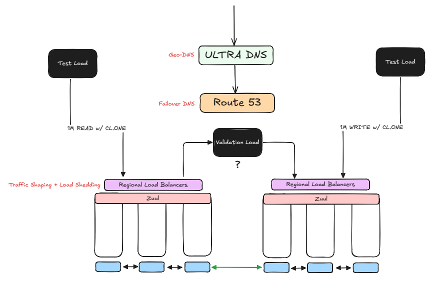
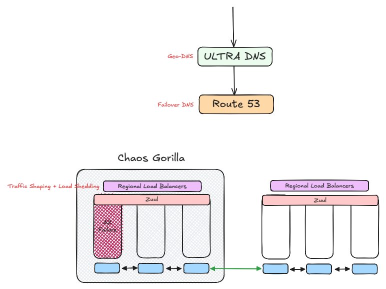
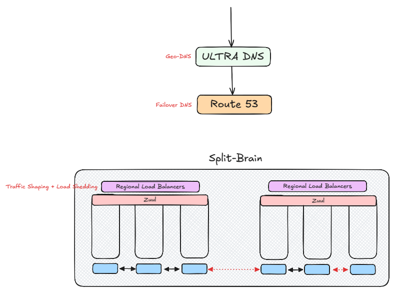
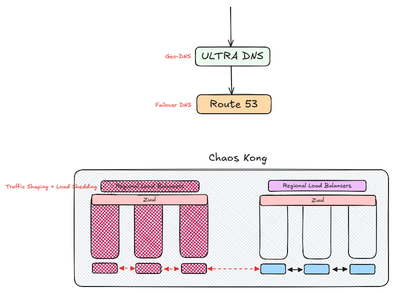
(R) Active-Active for Multi-Regional Resiliency









Notes

Active-Active for Multi-Regional Resiliency

Resiliency Page

> 회복 탄력성 Page

region-wide ELB outage. Page

> https://netflixtechblog.com/isthmus-resiliency-against-elb-outages-d9e0623484f3 Page

higher resiliency and availability — a full multi-regional Active-Active solution. Page

Failure — function of scale and speed.

---

failure rates an organization is dealing with depend largely on 2 factors: scale of operational deployment and velocity of change. Page

> 실패율은 1. 배포 규모 2. 변화의 속도 두 가지 요소에 영향을 받는다. Page

Ultimately, if you're running at scale and still pursuing high velocity — things will break all the time. Page

> 배포의 규모가 너무 크면 하드웨어 실패가 발생하기 쉽고, 변화의 속도가 너무 빠르면 소프트웨어 실패가 발생하기 쉽다. Page

At Netflix, our internal availability goals are 99.99% — which does not leave much time for our services to be down. Page

> 99.99%의 Availability라면 실제 서비스 Down에 허용하는 시간은 얼만큼일까? Page

Complete regional infrastructure outage is extremely unlikely, but our pace of change sometimes breaks critical services in a region, and we wanted to make Netflix resilient to any of the underlying dependencies. Page
Leveraging the principles of Isolation and Redundancy: a failure of any kind in one Region should not affect services running in another, a networking partitioning event should not affect quality of service in either Region. Page

> 특정 Region에서의 어떤 실패도 다른 지역에서 실행되고 있는 서비스에 영향을 미치지 않아야 하고, 네트워크 단절 상황(Region 간 연결 단절 / 네트워크 장애 / Latency 증가)이 어떤 Region의 서비스 품질에도 영향을 미치지 않아야 한다. Page

Active-Active Overview

Active-Active solution gets all the services on the user call path deployed across multiple AWS Regions — in this case US-East-1 in Virginia and US-West-2 in Oregon. Page
Services must be stateless — all data / state replication needs to handled in data tier. Page
needs to handled in data tier. Page

> 데이터의 저장 및 복제 작업이 Application 계층이나 서비스 계층이 아닌 데이터 계층에서 이루어져야 한다는 의미이다. Page

They must access any resource locally in-Region. Page
This includes resources like S3, SQS, etc. This means several applications that are publishing data into an S3 bucket, now have to publish the same data into multiple regional S3 buckets. Page
have to publish the same data into multiple regional S3 buckets. Page

> 구현 방식? >
>
> - 단일 버킷 업로드 로직은 멀티 리전 업로드로 변경
> - S3 Cross Region Replication (CRR) 사용
> - S3 다중 리전 액세스 포인트 활용 Page

there should not be any cross-regional calls on user’s call path. Data replication should be asynchronous. Page
There are several technical challenges in order to achieve such a setup Page
- Effective tooling for directing traffic to the correct Region Page
- Traffic shaping and load shedding, to handle thundering herd events Page

> 섬세한 트래픽 관리와 부하 감소를 통해 Thundering Herd 현상을 처리할 수 있어야 한다. >
>
> - 트래픽 관리를 위해서는 Rate Limiting / Priority Queueing / Token Bucket Algorithm 등을 구현할 수 있다.
> - Load Shedding을 위해서는 Request Dropping / Circuit Breaker / Fallback Mechanism 등을 도입할 수 있다. Page

thundering herd events Page

> 다수의 클라이언트가 동시에 동일한 리소스에 접근하려고 시도하는 상황을 의미한다. >
>
> - 일반적으로 트래픽 급증에서 발생
> - 서버 과부하 / 응답 지연 / 시스템 장애로 이어진다. Page

- State / Data asynchronous cross-regional replication Page

DNS — Controlling user traffic with Denominator

> Denominator? 분모 Page

UltraDNS Page
Route53 entries Page
Denominator project provides a single client library and command line that controls multiple DNS providers Page
UltraDNS provides us ability to directionally route customers from different parts of North America to different regional endpoints. Page

> 북미를 더 세부 지역으로 나누어 트래픽을 정확하게 분배할 수 있는 기능 :) `Directional DNS` 기능! Page

We didn't want to use a latency based routing mechanism because it could cause unpredictable traffic migration effects. Page

> 지연 기반 라우팅은 사용자에게 가장 낮은 지연 시간을 제공하는 지역으로 트래픽을 전달한다. 하지만... >
>
> - Region A의 네트워크가 일시적으로 혼잡해서 지연 시간이 100 -> 500으로 증가하는 경우 라우팅 시스템이 이를 감지하고 트래픽을 Region B로 전환하고, A -> B -> A -> B가 발생할 수 있다 (시스템 안전성이 떨어지는 효과) Page

By using Route53 layer between UltraDNS and ELBs, we have an additional ability to switch user traffic, and the Route53 API provides reliable and fast configuration changes that are not a strong point for other DNS vendor APIs. Page

> **ELB: Elastic Load Balancer / Route53: Amazon의 Route 53 (신뢰성이 높고 빠른 변경 적용이 가능하다! 자동화도 잘 되어 있다.)** >
>
> - Ultra DNS에서 전달된 트래픽을 ELB로 연결하기 전에 트래픽을 세밀하게 제어하는 별도의 계층 역할을 Route53이 수행하였다.
> - ELB를 통해 트래픽은 여러 AZ로 분산되며, 각 인스턴스의 상태를 모니터링하여 건강하지 않은 인스턴스를 자동으로 제외
> - Route53은 ELB로 트래픽을 라우팅하기 위해 Alias Record를 사용 (CNAME보다 효율적이고, Root Domain에서도 사용 가능하다) Page

Switching traffic using a separate Route53 layer makes such change much more straightforward Page

Zuul — traffic shaping

---

https://github.com/Netflix/zuul Page

All of Netflix Edge services are fronted with the Zuul layer. Page
Zuul provides us with resilient and robust ways to direct traffic to appropriate service clusters, ability to change such routing at runtime, and an ability to decide whether we should shed some of the load to protect downstream services from being over-run. Page
The enhancements were in several areas: Page

Zuul을 더 발전시켜 Active Active의 유즈케이스를 더 지원할 수 있게끔 했다. Page

- Ability to identify and handle mis-routed requests. Page
User request is defined as mis-routed if it does not conform to our geo directional records. Page

Geo-Directional Records와 일치하지 않는 요청을 식별. 단일 디바이스에서의 요청이 여러 region에 걸쳐 처리되는 일을 방지한다. Page

We also have controls for whether to use “isthmus” mode to send mis-routed requests to the correct AWS Region Page

정확한 Region에 요청을 보낼 수 있도록 제어한다 Page

- Ability to declare a region in a “failover” mode Page

보통은 Failover 모드가 되었다고 하면 요청을 다른 곳으로 보내려고 하는데, ZUUL에서의 Failover 모드는 다른 Region에 의존하지 않고 모든 요청을 자신의 Region 내에서 직접 처리하려고 시도하는 모드이다. (네트워크 지연이나 추가적인 장애를 방지하기 위한 설계이다). 해당 리전에 문제가 생겨도 그 영향을 다른 곳에 퍼뜨리지 않기 위한 모드이다. Page

this means it will no longer attempt to route any mis-routed requests to another region, and instead will handle them locally Page

장애 상황에서 서비스의 독립성과 안전성을 유지하도록 한다. Page

- Ability to define a maximum traffic level at any point in time, so that any additional requests will be automatically shed (response == error), in order to protect downstream services against a thundering herd of requests. Page

Rate Limiting과 유사한 역할도 ZUUL이 해주고 있다. Page

Replicating the data — Cassandra and EvCache

One of the more interesting challenges in implementing Active-Active, was the replication of users' data/state. Page
Apache Cassandra as our scalable and resilient NoSQL persistence solution. Page
the product’s multi-directional and multi-datacenter (multi-region) asynchronous replication. Page

Cassandra가 제공하는 장점: Multi-directional and Multi-datacenter (Multi-region) 비동기 복제 Page

Netflix has operated multi-region Apache Cassandra clusters in US-EAST-1 and EU-WEST-1 before Active-Active. Page
most of the data stored in those clusters, although eventually replicated to the other region, was mostly consumed in the region it was written in using consistency levels of CL_LOCAL_QUORUM and CL_ONE Page

> 실제로 데이터가 쓰인 장소에서 Consume되는 경우가 많더라...! Page

consistency levels of CL_LOCAL_QUORUM and CL_ONE Page

> 카산드라의 일관성 수준?
>
> - CL_LOCAL_QUORUM: 읽기 또는 쓰기 작업 시 로컬 리전 내에서 과반수의 노드가 응답해야만 성공으로 처리. 강한 일관성을 보장하고 지연 시간 또한 감소한다 (동일 리전에서만 처리하므로)
> - CL_ONE: 읽기 또는 쓰기 작업 시 로컬 리전 내. 1개 노드만 응답해도 성공으로 간주한다. (최소한의 노드 응답만 필요하므로 속도가 매우 빠르며, 데이터의 즉각적인 일관성보다는 가용성을 우선시한다) Page

- This lead us to perform an experiment where we wrote 1 million records in one region of a multi-region cluster. We then initiated a read, 500ms later, in the other region, of the records that were just written in the initial region, while keeping a production level of load on the cluster. Page

> 이걸 직접 해볼 수 있었다는게 대단하다. Page

we need to guarantee that data tier reads are fast — generally in a single-millisecond range. Page
we also front our Cassandra clusters with a Memcached layer, Page
Managing memcached in a multi-regional Active-Active setup leads to a challenge of keeping the caches consistent with the source of truth. Page
we added remote cache invalidation capability to our EvCache client Page
Whenever there is a write in one region, EvCache client will send a message to another region (via SQS) to invalidate the corresponding entry. Page

Automating deployment across multiple environments and regions

we doubled our deployment environments from two to four: Test and Production, US-East and EU-West Page
to six — adding Test and Production in the US-West region. Page
we quickly realized that our developers should not have to go through a sequence of at least 6 (some applications have more “flavors” that they support) manual deployment steps in order to keep their services consistent through all the regions. Page
our Tools team developed a workflow tool called Mimir, based on our open source Glisten workflow language, that allows developers to define multi-regional deployment targets and specifies rules of how and when to deploy. Page
Typically we wait for many hours between regional updates, so we can catch any problems before we deploy them world-wide. Page

Monkeys — Gorilla, Kong, Split-brain

Chaos Monkey — probably the most well known member of Simian Army runs in both Test and Production environments, and most recently it now includes Cassandra clusters in its hit list. Page

- Chaos Gorilla Page
It takes out a whole Availability Zone, in order to verify the services in remaining Zones are able to continue serving user requests without the loss of quality of service. Page
- Split-brain Page

> https://blog.naver.com/alice_k106/221310093541 Page

This is a new type of outage simulation where we severed the connectivity between Regions. Page
services in each Region continued to function normally, even though some of the data replication was getting queued up. Page
- Chaos Kong Page
This is the biggest outage simulation we've done so far. Page
We augmented what normally would have been an emergency traffic shifting exercise with extra steps so that users that were still routed to a "failed" region would still be redirected to the healthy region. Page
Also, we shifted traffic a bit more gradually than we would normally do under emergency circumstances in order to allow services to scale up appropriately and for caches to warm up gradually. Page

Real-life failover

- One of our middle tier systems in one of the regions experienced a severe degradation that eventually lead to the majority of the cluster becoming unresponsive. Page

> 실제로 특정 Region의 한 Middleware가 성능 저하를 겪어 클러스터 내 대부분의 노드가 응답하지 않는 사례가 발생했었다. Page

we decided to exercise the failover and route user requests to the healthy region. Page

> 장애 리전의 사용자 요청을 건강한 리전으로 즉시 전환하기 위해 Geo-DNS 설정을 변경하고 Zuul 라우팅 규칙도 업데이트함 + 대상 리전의 서비스가 추가 트래픽을 처리할 수 있도록 AWS가 Auto Scaling을 수행함 (인스턴스 증설) Page

We could then spend time triaging the root cause of the problem, deploying the fix, and subsequently routing traffic back to the now healthy region. Page

Next steps in availability

The project itself still has an upcoming Phase 2 — where we'll focus on operational aspects of all of our multi-regional tooling, and automate as many of current manual steps as we can. Page
We're also continuing to tackle some of the more difficult problems. Page
- how do you deal with some of your dependencies responding slowly, or returning errors, but only some of the time? Page
To help us learn how our systems deal with such scenarios, we have a Latency Monkey — it can inject latencies and errors (both client and server-side) at a given frequency / distribution. Page

> 미친... Page

Summary

The non-technical level of complexity is even more difficult , especially given that many other important projects were being worked on at the same time. Page