Laundry
Laundry
Laundry
Scaling

Edge
Scaling Problems

— Debuggability
— Infrastructure
— Managing
Scalability
Trade Offs
Accessible
Five Use Cases

- Melnitz
- Mantis
- Passport
- Device Types
- Sharding
Scaling Patterns for Netflix's Edge
Justin Ryan

Melnitz
Melnitz

Set<CustomerId>  Cookie Expiration
Bloom Filter: Compact Set which can answer the question is an object “Possibly in Set”?
Scaling Patterns for Netflix's Edge

Justin Ryan

{\{x, y, z\}}
Melnitz
Trade Offs

— Probabilistic Data Structure
— Off the Shelf Components
Five Use Cases

- Melnitz - Distributed Bloom Filter
- Mantis
- Passport
- Device Types
- Sharding
Scaling Patterns for Netflix's Edge

Justin Ryan
Scaling Patterns for Netflix's Edge

Justin Ryan

Mantis

2 million request per second *

3 kilobytes per request *

10 micro-services =

57 GB/s =~ 4.6 PB / day
Naive Approach
Mantis

Mantis Agent

Mantis Master

Mantis Source Job

Elastic Search

Kibana

Kafka

Scaling Patterns for Netflix's Edge
Justin Ryan
Mantis
Trade Offs

— Mantis Query Language
— Scale Solution to Zero
Five Use Cases

- Melnitz - Distributed Bloom Filter
- Mantis - Event Querying
- Passport
- Device Types
- Sharding
Passport

[Diagram]

- L7 Proxy
- API
- Mid Tier A
- Mid Tier B
- Mid Tier C
- Service Plan

Scaling Patterns for Netflix's Edge
Justin Ryan
Scaling Patterns for Netflix's Edge

Justin Ryan

Cache

L7 Proxy

API

Mid Tier A

Mid Tier B

Mid Tier C

Service Plan
Scaling Patterns for Netflix's Edge

Justin Ryan
Passport Trade Offs

- Data Passing
- Heavy Data Structures
Five Use Cases

- Melnitz - Distributed Bloom Filter
- Mantis - Event Querying
- Passport - Heavy Data Structure
- Device Types
- Sharding
Device Types
Device Types

NFAPPL-01-IPAD7=5-09BEC57F64CA5F837C75F924B94D → 1395

NFAPPL-01-IPAD3=4-15B2D9B33A994CC373017A9C6586 → 1007

NFAPPL-02-IOS2=4-FT7WHG8H2UR2MCJ4YKR3P0DFH6 → 964

NFANDROID1-PRV-P-GOOGPIXEL=2-7351-0F441DF98B8D99074

NFANDROID1-PRV-4339-0E50FC7582396F5C4496ED935F2F48ED
Device Types
Device Types
Device Types

- Mid-Tier A
- Mid-Tier B
- Mid-Tier C
- Mid-Tier D
- Mid-Tier E
- Mid-Tier F
- DTS
Device Types

- DTS
- PubSub
- Mid-Tier A
- Mid-Tier B
- Mid-Tier C
Device Types

DTS → PubSub

Mid-Tier A

Mid-Tier B

Mid-Tier C

Device Types

Device Types

Device Types
Device Types
Device Type Trade Offs

- Business Tuned Fallbacks
- Leverage Existing Infrastructure
Five Use Cases

- Melnitz - Distributed Bloom Filter
- Mantis - EventQuerying
- Passport - Heavy Data Structure
- Device Types - Device Grouping
- Sharding
MSL

— Message Security Layer == MSL

— Secure Messaging Framework
Sharding MSL

L7 Proxy ➔ API
Sharding MSL
CPU Cost Per Request

31% decrease
Latency

30% decrease

20% p99 decrease

Scaling Patterns for Netflix's Edge
Justin Ryan
Sharding MSL

- % of Wall Clock in GC
- GC Pause Time
Sharding

3 = 1 + 1
Sharding Trade Offs

— Operational Overhead
Five Use Cases

— Melnitz - Distributed Bloom Filter
— Mantis - Event Querying
— Passport - Heavy Data Structure
— Device Types - Device Grouping
— Sharding - Runtime Refactoring
**Bonus Trade Offs**

— Do the laundry
Thank You.

Justin Ryan
jryan@netflix.com

Twitter: @quidryan
References

- github.com/Netflix
  - eureka
  - zuul
  - mantis
  - msl
- Spinnaker
- Apache Pulsar
- Google Guava (Bloom Filter)