Linkedin
How Robust Monitoring Powers High Availability for LinkedIn Feed

Rushin Barot
SREcon17
LinkedIn
Overview

- Linkedin Feed architecture
- Monitoring
- High availability for Feed
- Best practices
LinkedIn in 2016

- 467 million member accounts
- 107 million active users, 40 million+ messaging conversations per week
- Kafka: 1.4 trillion messages/day across 1400 brokers
- 2-3x increase in referral traffic YoY
Linkedin Feed Architecture

- Linkedin
  - Business Logic
    - DataStore
    - Data ingestion
    - TimeOrdered DB
  - Mid Tier
    - Business Logic
    - LinkedIn
Monitoring

What do we monitor
Monitoring

What do we monitor
Monitoring

How do we monitor

- **Ingraphs**
  > Frontend to round-robin database (rrd)
- **Autometrics**
  > Self-service metrics collections
- **Nurse**
  > Auto Remediation
- **Iris**
  > Similar to pagerDuty
High Availability for Feed

- Data partition
  -------------> 720 partitions
- Replicas
  -------------> 3 replicas
- Back up and Restore
  -------------> separate back up nodes
- Quotas
  -------------> throttle
- Fail-over
- Live load test
Best Practices

- Automated hardware issue detection.
- A/B testing
- Proof rules (aka Error budgeting)
- Canaries
- Dark Canaries
Dark Canary

- Mirror server in production which receives a copy of production traffic
- Responses are never sent to clients
- Source server and dark canary
Dark Canary
Dark Canary
Specifics for Feed

- Read-Only Requests
- Scalability using traffic multiplier
- Dark Canary failures have no impact on production
- Differentiation from production traffic
  - Page key: dark canary requests have separate page keys.
  - Tree id: Separate tree id
Dark Canary Usage

- New build verification
  - Detection of schema changes, code issues, misconfigurations
- New colo certification
- Warming up caches in multi-colo environment
- Capacity planning
Dark Canary
Capacity Measurement via Redliner

- Use live traffic in production environment
- No significant impact on production node
- Minimal maintenance overhead
Dark Canary
Capacity Measurement via Redliner
Results
System Stats - CPU

![Graph showing CPU utilization over time for Source Node and Dark Node. The graph indicates a significant increase in CPU utilization for the Dark Node starting from 3 hours onwards.]
System Stats - Memory

![Graph showing Memory Usage over Time for Source Node and Dark Node.](graph.png)
Advantages and Limitations

Pros

● Proactive measure to catch issues
● Low cost of testing with real production data and traffic patterns
● No tests implementation/maintenance is required
● Configurable traffic multiplier
● New code can be deployed and tested in parallel with staging before pushing it to regular Canary

Cons

● Read-Only requests
● Recommended for services closer to the bottom of the stack due to calls to production downstream services
● Additional hardware for Dark Canary
Thank you
References

Linkedin Tech Blog

- Followfeed Architecture
- Nurse
- Ingraphs
- Autometrics

Books

Site Reliability Engineering

Paper

Capacity Measurement and Planning for NoSQL Databases by Ramya Pasumarti, Rushin Barot, Susie Xia, Ang Xu, Haricharan Ramachandra [IEEE SDI 2017]