LinkedIn SRE: From Inception to Global Scale

Bruno Connelly & Viji Nair
About Us

Bruno Connelly
VP, Engineering

Joined LinkedIn in April 2010

Background in ISP and Consumer Internet

Initial experience team building in Asia in 2005
About Us

Viji Nair
Director, SRE

Joined LinkedIn in January 2012

16 Years as an SRE + Startups

Open source evangelist and contributor

Working with global teams since 2006
OUR VISION

Create economic opportunity for every member of the global workforce
Growing Global Network

- 500M+ Members
- 100K Articles published weekly
- 100M+ Monthly Unique Visitors
- 2 New sign-ups per second
- 50% Active members use LinkedIn Messaging weekly
- 40% yr/yr increase in engaged feed sessions weekly
<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge QPS</td>
<td>340K</td>
</tr>
<tr>
<td>Services in production</td>
<td>1.5K</td>
</tr>
<tr>
<td>Graph QPS</td>
<td>770K</td>
</tr>
<tr>
<td>Graph Edges</td>
<td>60B</td>
</tr>
<tr>
<td>Data storage</td>
<td>600TB</td>
</tr>
<tr>
<td>Peak Data QPS</td>
<td>12M</td>
</tr>
<tr>
<td>Kafka Messages published/day</td>
<td>2T</td>
</tr>
<tr>
<td>Kafka Messages consumed/day</td>
<td>4T</td>
</tr>
</tbody>
</table>
Production Application Footprint

- OREGON
- VIRGINIA
- TEXAS
- SINGAPORE

4 Active Data Centers
Looking back... 2010
Entire Production Footprint

2010

1 Active Data Center

2 Data Centers

CHICAGO

LOS ANGELES
LinkedIn Operations

Classical, stratified model: Systems, Networks, Applications, DBA

Heavy-weight processes driven by tickets and heroes

Culture of not trusting developers in any deployed environments

Huge wall and growing frustration between Dev and Ops teams (and in ops itself)

7 engineers in total made up NOC, SRE, Release Operations: “Site Operations”

On-call was horrible
Member Growth

7 Years of Tech Debt

32% YOY Growth

We were here
Is the Site Up?

- Peak traffic periods Mon-Wed ~ 6-10am
- Regular capacity related outages Mon-Wed ~ 6-10am
- Zero tolerance for failure in the application stack
- Near zero instrumentation
- Bi-weekly downtime maintenances
Let’s make a few changes

- change software development model
- code contributions to the main application stack
- cheaper datacenters
- dev driven deployments
- active/active serving model
- more data centers
- move faster
- replace java serialized objects over RPC with REST APIs
- modernize our application stack
- graceful degradation
- remove monolithic databases
- remove hardware load balancers
- auto escalation
- 24/7 deployments
- automated datacenter buildout
- move to service oriented architecture
- 3x3 deployments
- self service everything
- auto remediation
- move faster
Core Principles

1. Site Up
2. Empower Developer Ownership
3. Operations is an Engineering Problem
Everyone should be able to deploy code safely
Self-service Deployments

1. “Canary” to a single production instance
   - 15K+ Successful commits/day

2. EKG: automated metrics-based validation
   - 200+ Code promotions/day

3. Promote to a single production data center
4. Promote to remaining production data centers
   - 600+ Feature ramps/day

5. Ramp features slowly to the member base
Create a culture of operational metrics

“What gets measured gets fixed.”
Self-service Instrumentation and Monitoring

- **600M+** Total metrics
- **340K** Alerts processed/min
- **10M** Metrics ingested/sec
- **23K** Graph dashboards

- Kafka
- Metrics collectors
- Metrics API
- Alerting
- Visualization

Java applications
Non-java applications
REST API
IRIS
We don’t want a traditional NOC [permanently]
Self-service Remediation and Escalation

- Alerts
- Nurse
- Salt
- Deployment
- Metrics
- Notify (IRIS, JIRA, etc.)
- Correlation Engine
- Feedback

- 15K Remediation Plans
- 9K Escalation Plans
- 17K Executions/day
Unexpected outcomes from a different model

- Tiered escalation systems did not scale and incentivized undesirable outcomes
- Introduced “Production SRE”; SREs solving NOC problems via software
- Created NOC to SRE transition program
Scaling the team into India
Ground Rules

1. Follow the sun is not the goal
2. Start simple
3. Same culture, same hiring bar, same everything
We made mistakes. Principles & playbook.

We learned from them.
Lesson 1: Bootstrapping Remote Teams Is Hard

1. Applying US model did not work
   - Invest in potential & new grads

2. Negative association of the “SRE” title
   - Standardize SRE title

3. Pre conceived notion of US based companies
   - Evangelize SRE role
Lesson 1: Bootstrapping Remote Teams Is Hard

Outcomes:

1. Don’t expect hiring patterns from other markets to necessarily work.
2. Leverage other sources: College Grads, High-Potential junior engineers.
3. Leverage the community to help; you’re likely not alone.
Lesson 2: Create Your Own Identity

1. Mirroring teams does not work
   Invest based on value addition & local needs

2. Work patterns are different
   Focus on opportunity and strengths

3. Need engaging work to sustain teams
   Provide quality work for the teams
Lesson 2: Create Your Own Identity

OUTCOMES

- Don’t map teams 1:1 across offices
- Use lulls in workload to give them hard problems they can own
- Find the right balance, quality vs quantity
Lesson 3: One Size Does Not Fit All

1. Every team is unique
   - Curate to specific team need

2. Equal ownership is critical to success
   - Equal partnership in running a product

3. Right leadership is essential
   - Alignment on leadership commitment & investment
Lesson 3: One Size Does Not Fit All

OUTCOMES

Focus on teams that have shown that they can collaborate

Mistakes will happen; own them and fix them quickly
Bangalore Today

Strong team presence:
- 60+ SREs across 10 teams
- 100+ application developers
- Owns LinkedIn core messaging platform

Building region-specific products:
- Performance focused mobile client
- New college grad placements
- Filling India’s blue-collar labor gap
SRE Globally Today

- Sunnyvale (SNV)
- San Francisco (SF)
- Bangalore (BLR)
- New York City (NYC)

300+ SREs across four global offices

Learnings from BLR led to other regional expansions
Takeaways from our Journey

- Directionally good ideas haven’t failed us yet
- Unique office identities leveraged for other engineering presences
- It really does take a village
- This isn’t easy, but it is possible
Thanks!