



# Building High-Performance Cache for Fun and Profit

*A personal account on working in the industry*

Yao Yue ([@thinkingfish](https://twitter.com/thinkingfish))

# When I quit school

- 3 year into PhD
- 3 internships
- 2 papers as first author
- no production experience

*Felt at a loss- I was not doing great or meaningful research.*



# Now

- Sr. Staff SWE at Twitter
- Expert on cache
- Manage a team of 8
- Starting to do research again...

*I know what I am doing, sort of.*



# What cache?

- Distributed, in-memory store running in data centers
- A cornerstone in scalability
- Conceptual simplicity
- Stringent runtime requirements
  - Usually the service with the highest throughput in a Web SOA
  - ... and lowest latency



**Phase 1.**

**Drudgery and Insight**

# What I did

- Incidents, so many incidents
  - Source code
  - System & hardware
  - Use cases & requirements
- Machines, so many machines
  - Operations
  - Capacity and efficiency
- A little bit of feature
- A lot of debugging, refactoring

# What I learned



- What works in production?
- Take responsibility
  - Not the same as blame
  - More like ownership
- How to collaborate
  - Call out BS (respectfully)
  - Care about others' problems
  - It's not a competition
- Form good habits



**Phase 2.**

**Design & implement *Pelikan***



# What I did

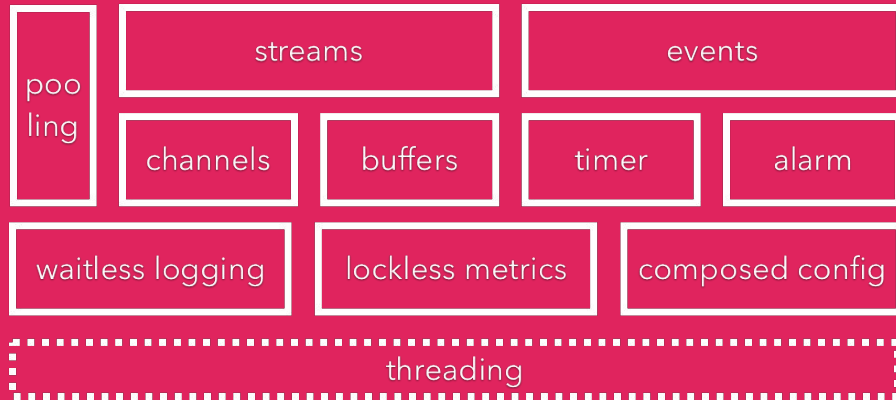
- Characterize *cache*
  - Purposes and environment
  - Desirable features & behavior
- Clean-slate architecture
  - Avoid inherent design flaws
  - Iteration friendly
- Disciplined implementation
  - Borrow and refactor
  - Write my own



# What I did

- Latency-oriented threading
  - Data plane, control plane
- Predictable memory behavior
  - Fragmentation management
  - Reuse & preallocation
- Minimize contention
  - Lockless metrics
  - Pauseless logging
  - SPSC queues

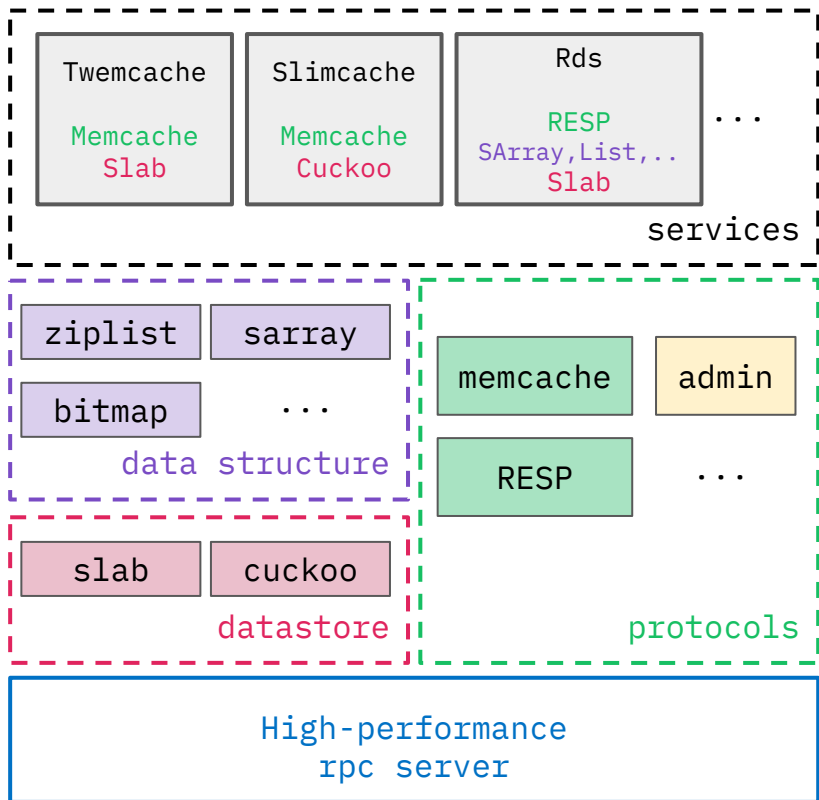
## cache common







# What I did



# What I learned

- Production-guided design
- Abstractions
  - Most important contract
  - Accurately substantiated
- Craftsmanship & aesthetics
  - Attention to details
  - Simplicity
  - Symmetry
  - Consistency

# What I did

- Migration plan
- Getting support
  - Merit of the technology
  - Branding
  - Value to the business
- Work on user experience
  - For customer
  - For developer

# What I learned

- Management is important
- Marketing is important





**Phase 3.**

**Pivot and Revisit**

# What I did

- Cache, canary in the perf mine
- I created a new team
  - To work on performance, and
  - Neglected problems, and
  - Research
- Continue to work on cache  
(without being on cache)



# What I learned

- Performance work saves (a lot) of money!
- Find the right problem
- Meta skills & universal patterns
- Articulate your vision

# Ongoing...

- Pelikan with new hardware
  - What does ***persistent memory*** mean for cache availability and maintenance? [research]
  - How to best productionize ***userspace networking?***
- New data structures
- Better analytics [research]
- Highly automated operations

# What I learned



- Research opportunity exists in real problems
- Zoom in, zoom out
- Keep up with the changing landscape



# What research taught me...

- Strong opinions, loosely held
- Perseverance is under-appreciated
  - Getting by, giving up is common
  - The desire to advance state of the art
- Navigate a vast and vague space
  - Asking the right questions
  - Connecting the dots
  - Working on the boundary
- Communication