GeekBand 极客班

互联网人才 + 油站!

GeekBand 极客班

www.geekband.com

系统设计中七剑客

GeekBand 极客班

大纲

同步
网络
数据库
分布式
性能
估算
面向对象
案例
- 社交网站信息流
- 日志统计
- 网络爬虫
- 电商产品页面

GeekBand 极客班

Introduction

System design:

1-3 rounds in interviews

For new-grad: not required/simple

For experienced: important to show your knowledge and thoughtful ideas

Knowledge, Design, Communication!

GeekBand 极客班

Concurrency

Thread vs. Process

Consumer and Producer

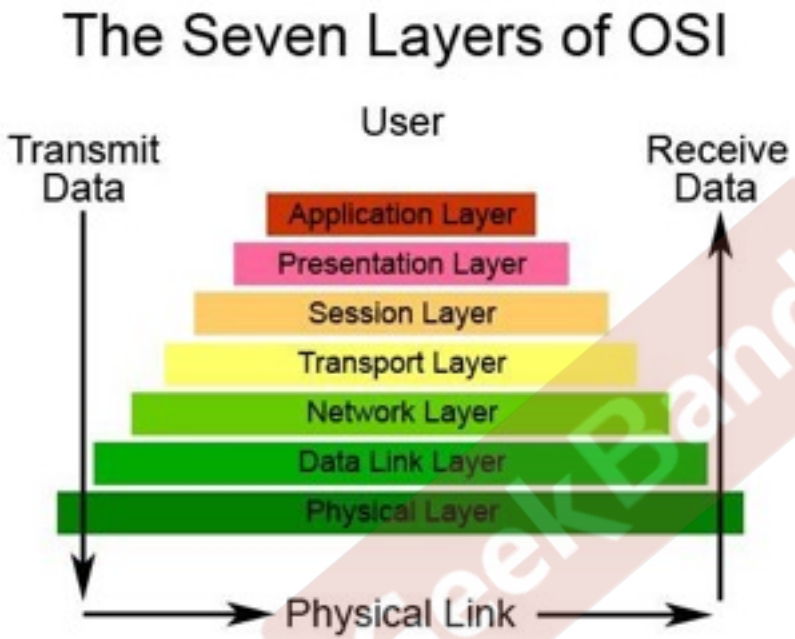
Blockingqueue

Tracking:

Synchronized, Asynchronized

GeekBand 极客班

Network



GeekBand 极客班

Visit URL

What happens after you typed a URL in your browser and pressed return key?

GeekBand 极客班

Database

Relational DB vs. KV Store

Sharding vs. Clustering

TinyURL:

Store the mapping from shortlink code to full URL. The record/ document:

code: varchar(8)

url: varchar(1000)
created_at: timestamp

We also need to store the reverse mapping from URL back to code.

GeekBand 极客班

Distribute System

How to scale Tiny URL service?

- Stateless frontend servers behind a load balancer
- Sharded/replicated database (on shortlink code)
- Memcached to scale read traffic
- Spread write load
- Locally buffered event tracking + async flush to high-throughput message queue
- Use a distributed unique ID generator (64-bit)

GeekBand 极客班

Performance

Cache is KEY!

Numbers Everyone Should Know

L1 cache reference	0.5 ns
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	25 ns
Main memory reference	100 ns
Compress 1K bytes with Zippy	3,000 ns
Send 2K bytes over 1 Gbps network	20,000 ns
Read 1 MB sequentially from memory	250,000 ns
Round trip within same datacenter	500,000 ns
Disk seek	10,000,000 ns
Read 1 MB sequentially from disk	20,000,000 ns
Send packet CA->Netherlands->CA	150,000,000 ns

GeekBand 极客班

Estimation

How many piano tuners are there in the entire world?

Tiny URL: How much is total storage?

URL Length 10 - 1000 chars.

Total accumulated URL number 100 M

New URL registrations are on the order 100,000/day (1/sec)

Redirect requests are on the order of 100M/day (1000/sec)

Geek Band 极客班

Design Pattern

23 patterns:
MVC
Singleton
Factory
Iterator
Decorator
Facade

GeekBand 极客班

案例

News Feeds

Stats Server

Web Crawler

Amazon Product Page

GeekBand 极客班

News feed

Define feed

Organize
- aggregate
- dedup
- sort

GeekBand 极客班

Level 1.0

Database Schema:

- User
- Friendship
- News

Get Newsfeed:

merge news

Newsfeed vs News

UserId	*Name*	*Age*
1	*Jason*	*25*
2	*Michael*	*26*

FriendshipId	*SourceId*	*TargetId*
1	*1*	*2*
2	*2*	*1*

NewsId	*AuthorId*	*Content*
1	*2*	*“Hehe”*
2	*1*	*“Lala”*

GeekBand 极客班

Why bad?

100+ friends

1 Query --> Get friends list

1 Query -->

```
SELECT news
WHERE timestamp>xxx
AND sourceid IN friend list
LIMIT 1000
```

IN is slow

Either Sequential scan or 100+ index queries

GeekBand 极客班

Level 2.0

Pull vs Push

Pull: Get news from each friend, merge them together. (NewsFeed generated when user request)

Push: NewsFeed generated when news generated. (we have another table to store newsfeed, may cause duplicate news)

Push:

1 Query to select latest 1000 newsfeed.

100+ insert queries (Async)

Disadvantage: News Delay.

Level 3.0

- Popular star (Justin Bieber)
  - Flowers 13M +
  - Async Push may cause over 30 minutes (13M+ insertions, delay too long)
- Push + Pull
  - for popular star, don’t push news to flowers
  - for every newsfeed request, merge non-popular user newsfeed (push) and popular users newsfeed (pull)

GeekBand 极客班

Level 4.0

Push disadvantage:

- Realtime
- Storage (Duplicate)
- Edit

Go back to PULL:

- Cache users' latest (14 days) news
- Broadcast multiple request to multiple servers (Shard by userId).
- Merge & sort newsfeed
- Cache newsfeeds for this user with timestamp

GeekBand 极客班

Click Stats Server

How are click stats stored?

- A poor candidate will suggest write-back to a data store on every click
- A good candidate will suggest some form of aggregation tier that accepts clickstream data, aggregates it, and writes back a persistent data store periodically
- A great candidate will suggest a low-latency messaging system to buffer the click data and transfer it to the aggregation tier.
  - If daily, storing in hdfs and running map/reduce jobs to compute stats is a reasonable approach
  - If near real-time, the aggregation logic should compute stats

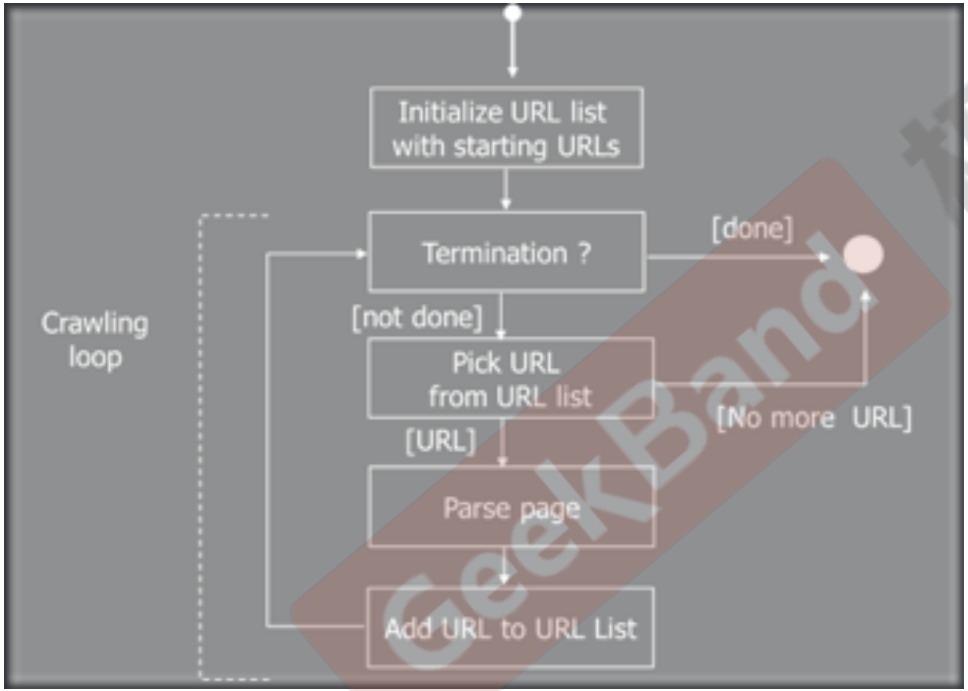
极客班

Cache Requirement

[a] When a request comes look it up in the cache and if it hits then return the response from here and do not pass the request to the system
[b] If the request is not found in the cache then pass it on to the system
[c] Since cache can only store the last n requests, Insert the n+1th request in the cache and delete one of the older requests from the cache
[d]Design one cache such that all operations can be done in O(1) – lookup, delete and insert.

GeekBand 极客班

Web Crawler



GeekBand 极客班

Amazon Product Page

The product page includes information such as

a) product information
b) user information
c) recommended products (what do other customers buy after viewing this item, recommendations for you like this product, etc)

GeekBand 极客班

Reference

http://highscalability.com/

The Log: What every software engineer should know about real-time data's unifying abstraction

Job Interviews: How should I prepare system design questions for Google/Facebook Interview?

HOW TO ACE A SYSTEMS DESIGN INTERVIEW

<Design Pattern>

<Design_Patterns_For_Dummies.pdf>

http://www.hiredintech.com/app

GeekBand 极客班