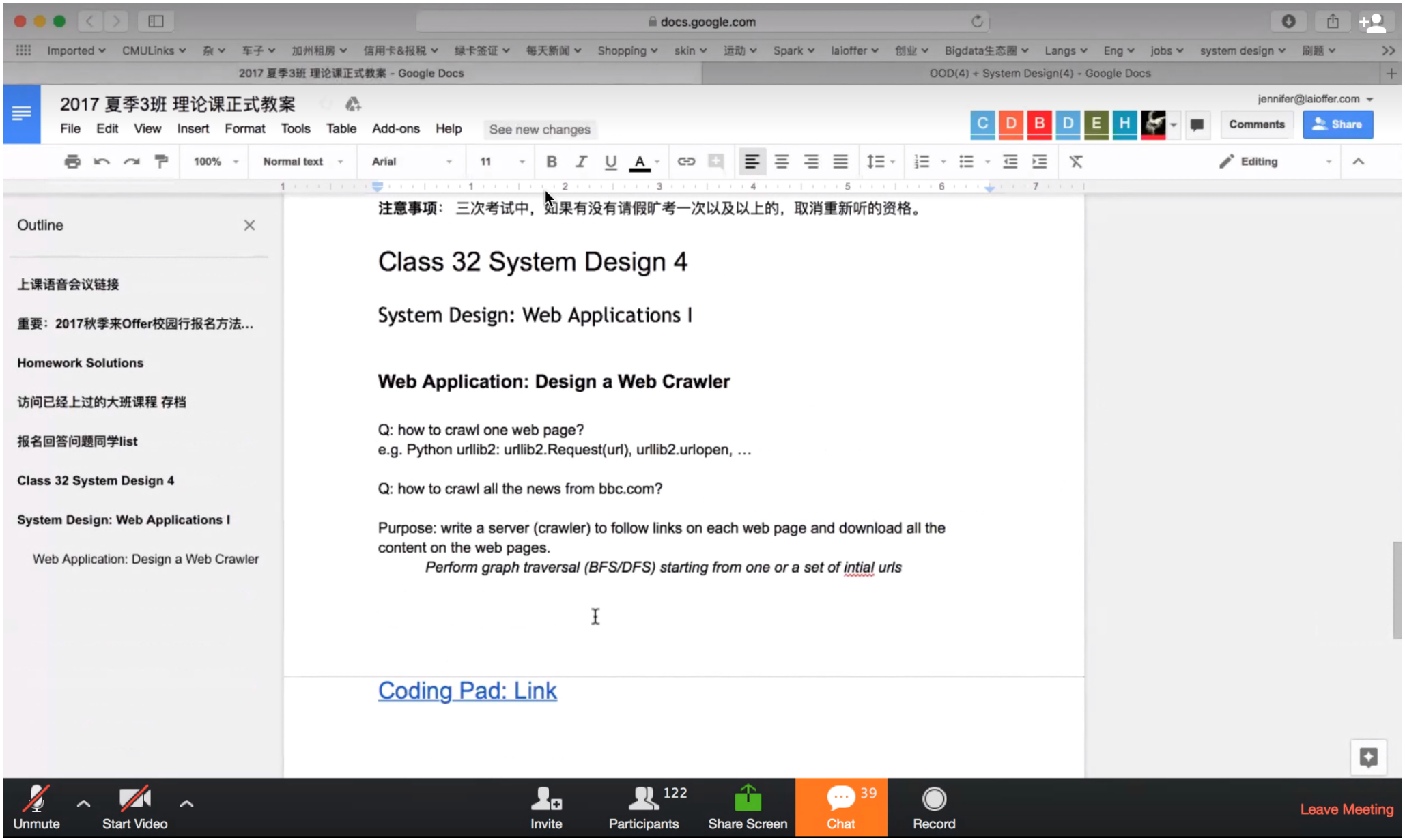
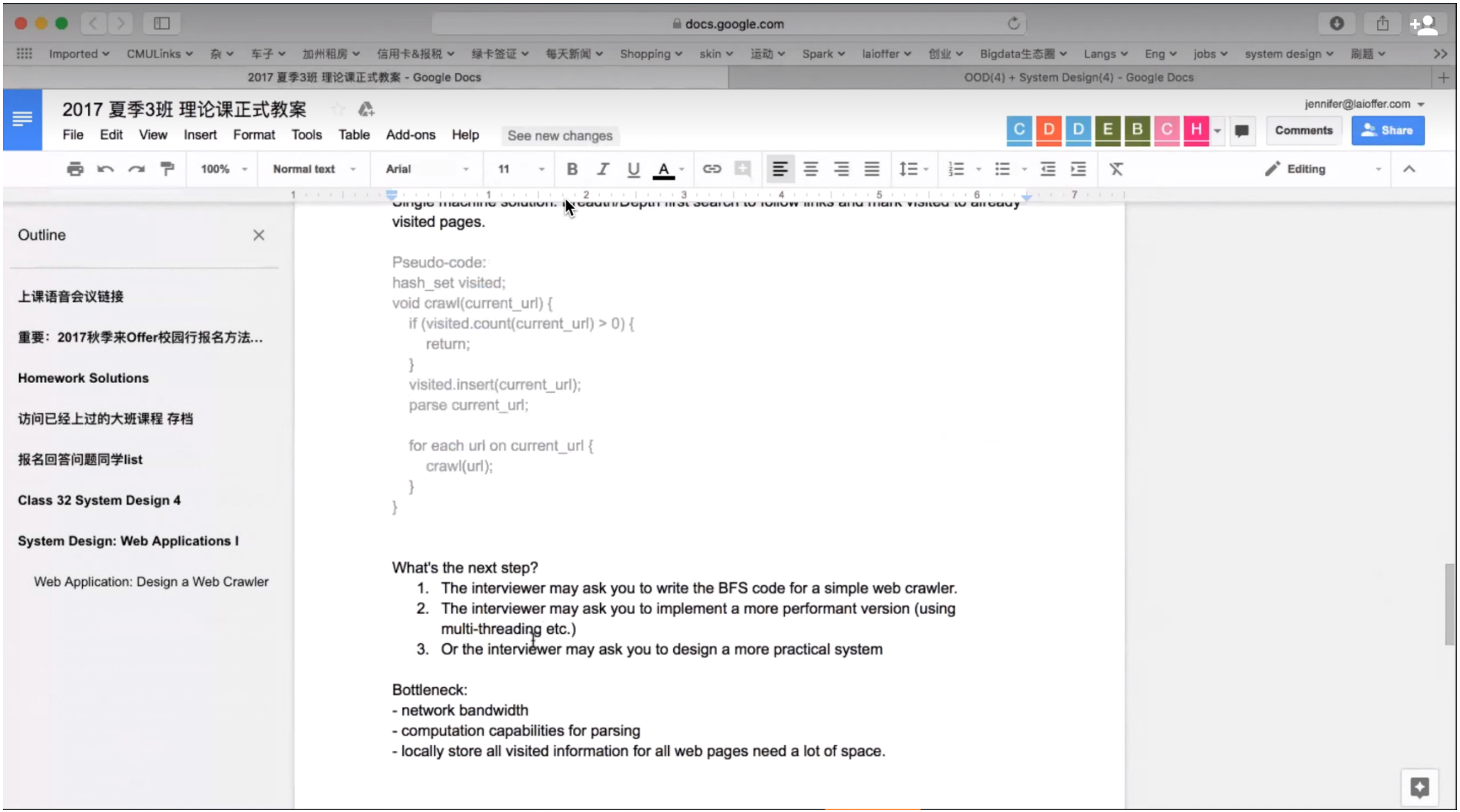
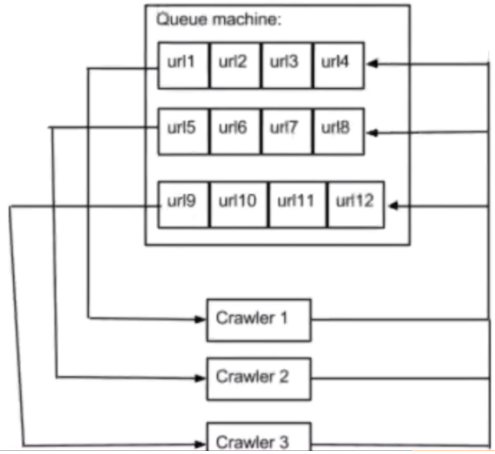

docs.google.com
2017 夏季3班 理论课正式教案 - Google Docs
OOD(4) + System Design(4) - Google Docs
2017 夏季3班 理论课正式教案
File Edit View Insert Format Tools Table Add-ons Help See new changes
jennifer@laioffer.com
Comments
Share
Editing
Outline
上课语音会议链接
重要：2017秋季来Offer校园行报名方法...
Homework Solutions
访问已经上过的大班课程 存档
报名回答问题同学list
Class 32 System Design 4
System Design: Web Applications I
Web Application: Design a Web Crawler
注意事项：三次考试中，如果有没有请假旷考一次以及以上的，取消重新听的资格。
Class 32 System Design 4
System Design: Web Applications I
Web Application: Design a Web Crawler
Q: how to crawl one web page?
e.g. Python urllib2: urllib2.Request(url), urllib2.urlopen, ...
Q: how to crawl all the news from bbc.com?
Purpose: write a server (crawler) to follow links on each web page and download all the content on the web pages.
Perform graph traversal (BFS/DFS) starting from one or a set of intial urls
Coding Pad: Link
Unmute
Start Video
Invite
Participants 122
Share Screen
Chat 39
Record
Leave Meeting

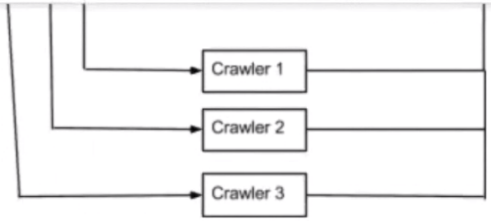

docs.google.com
2017 夏季3班 理论课正式教案 - Google Docs
OOD(4) + System Design(4) - Google Docs
2017 夏季3班 理论课正式教案
File Edit View Insert Format Tools Table Add-ons Help See new changes
jennifer@laioffer.com
Comments
Share
Editing
Outline
上课语音会议链接
重要：2017秋季来Offer校园行报名方法...
Homework Solutions
访问已经上过的大班课程 存档
报名回答问题同学list
Class 32 System Design 4
System Design: Web Applications I
Web Application: Design a Web Crawler
visited pages.
Pseudo-code:
hash_set visited;
void crawl(current_url) {
if (visited.count(current_url) > 0) {
return;
}
visited.insert(current_url);
parse current_url;
for each url on current_url {
crawl(url);
}
}
What's the next step?
1. The interviewer may ask you to write the BFS code for a simple web crawler.
2. The interviewer may ask you to implement a more performant version (using multi-threading etc.)
3. Or the interviewer may ask you to design a more practical system
Bottleneck:
- network bandwidth
- computation capabilities for parsing
- locally store all visited information for all web pages need a lot of space.

3. Or the interviewer may ask you to design a more practical system

Bottleneck:
- network bandwidth
- computation capabilities for parsing
- locally store all visited information for all web pages need a lot of space.

We need a distributed crawler system:
The problem for a distributed crawler system is how to maintain the information and synchronize with each other. e.g., one machine has visited one page, how does the rest of the machines know?



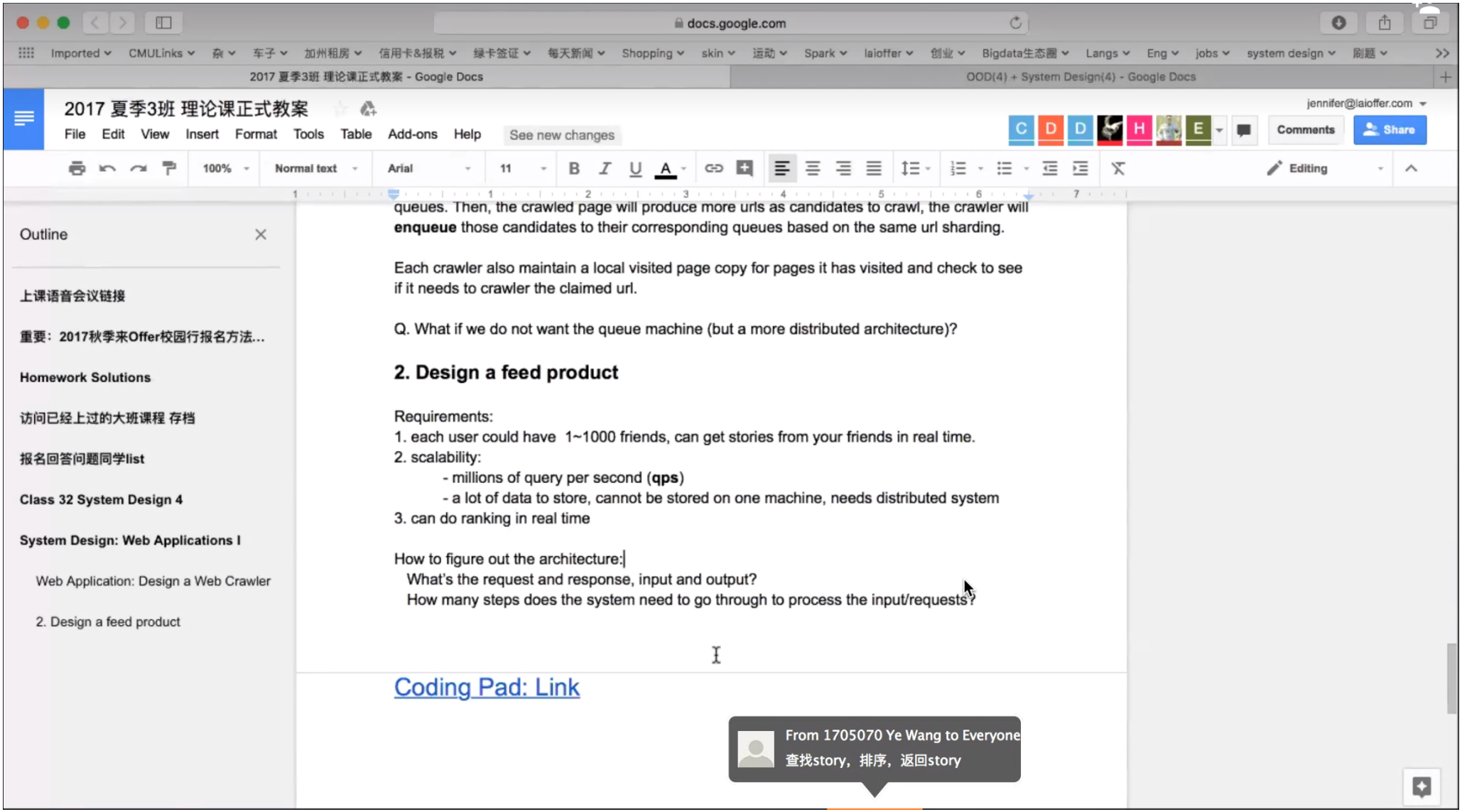


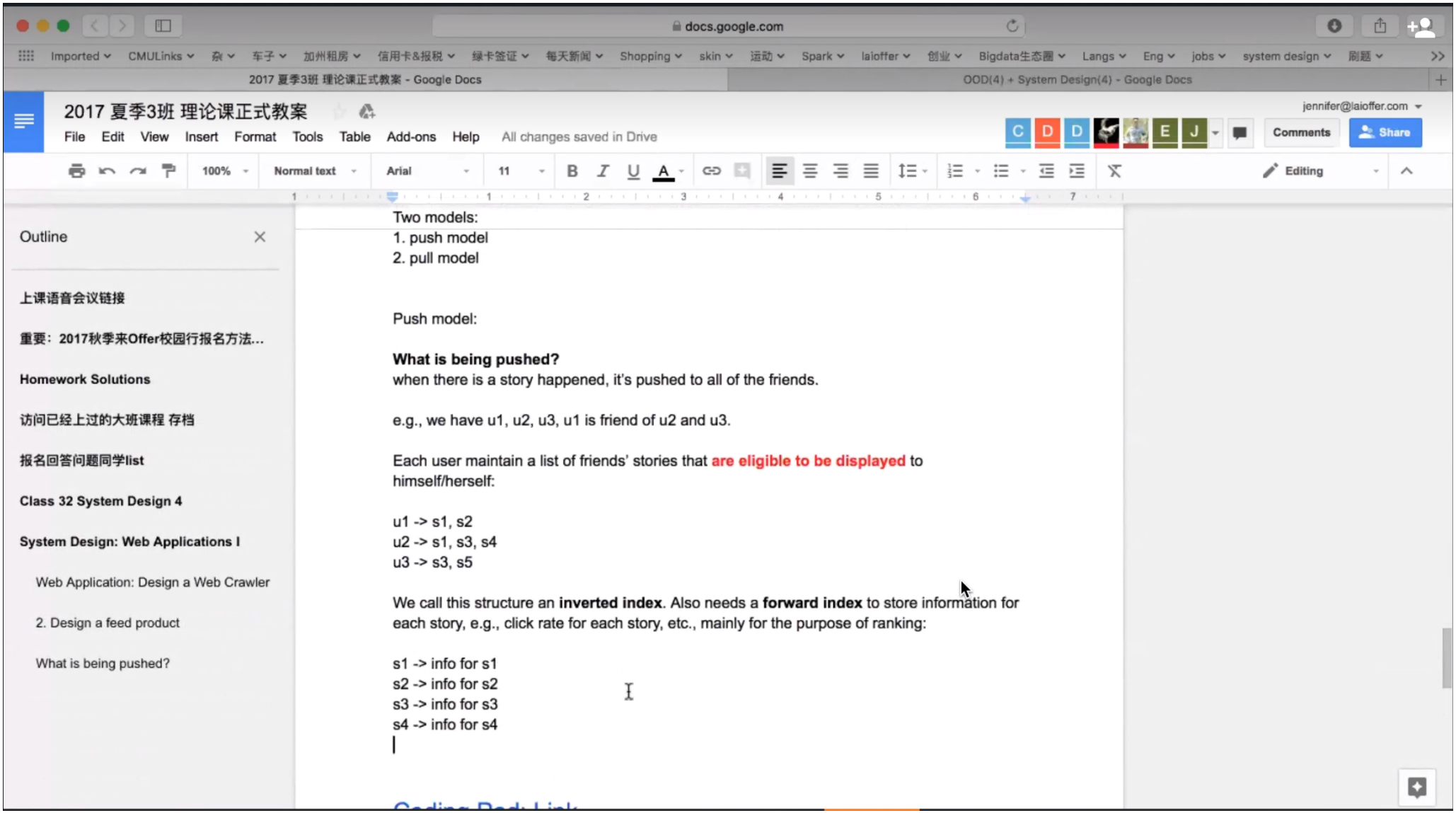
The queue is **sharded** by url. One url is **deterministically** going to one of the queues. Each crawler corresponds to one of the queues, i.e., it will only claim url to crawl from one of the queues. Then, the crawled page will produce more urls as candidates to crawl, the crawler will **enqueue** those candidates to their corresponding queues based on the same url sharding.

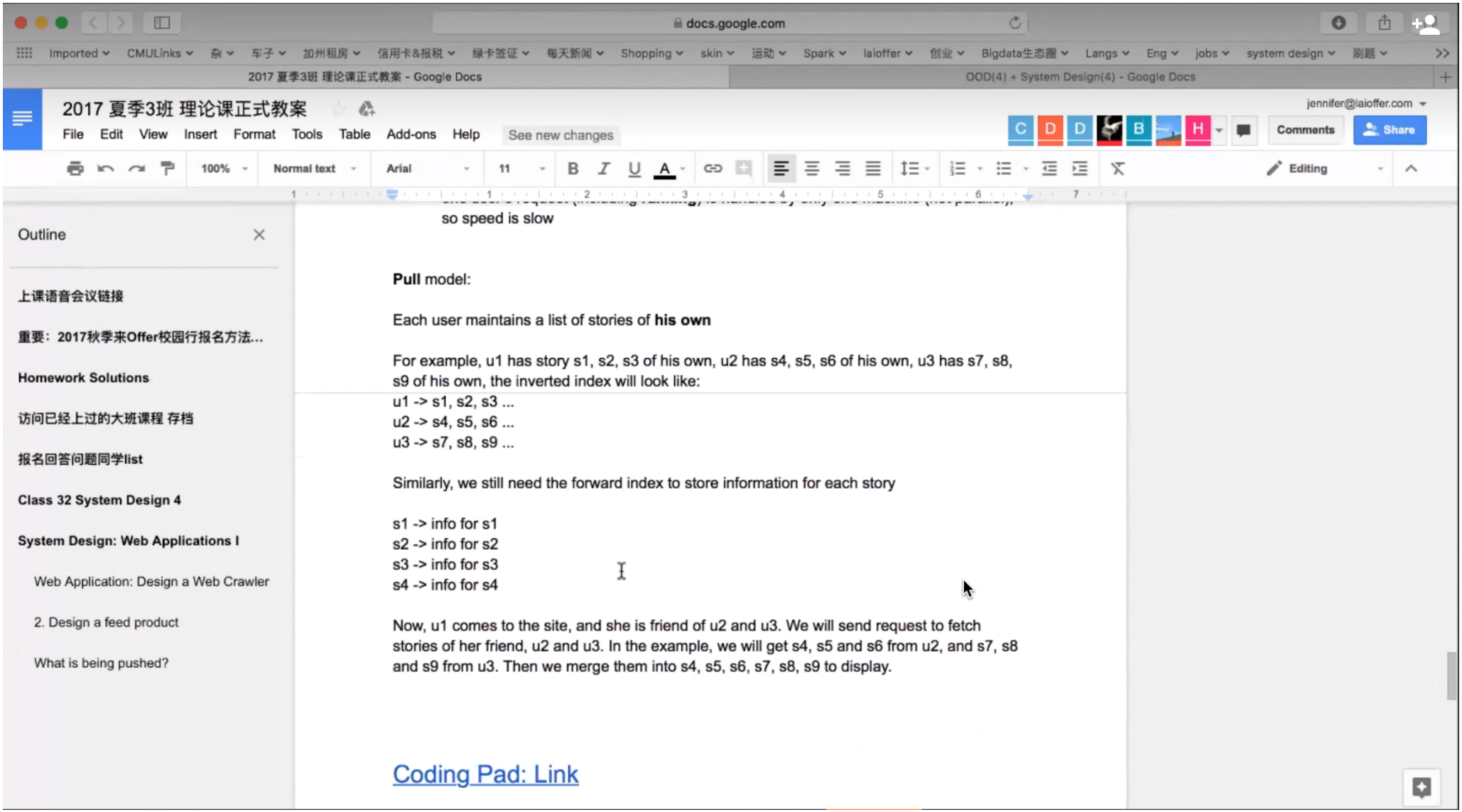
Each crawler also maintain a local visited page copy for pages it has visited and check to see if it needs to crawler the claimed url.

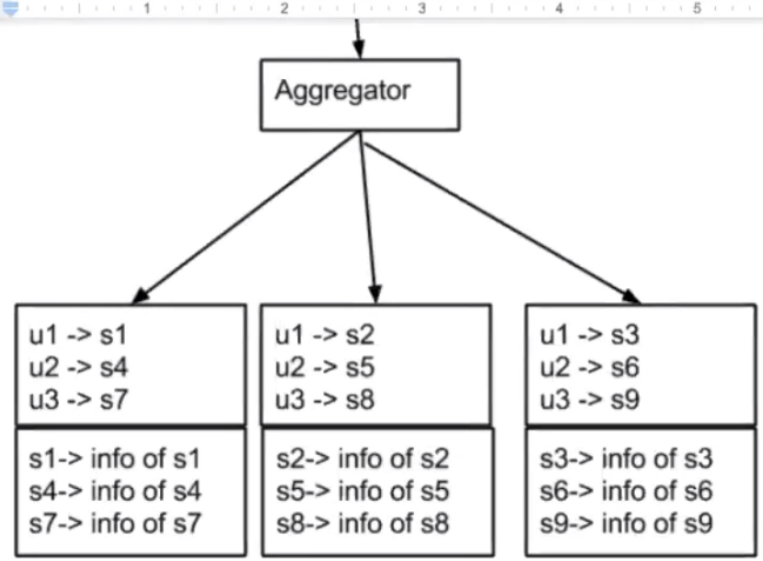
Q. What if we do not want the queue machine (but a more distributed architecture)?

Coding Pad: Link


docs.google.com
2017 夏季3班 理论课正式教案 - Google Docs
OOD(4) + System Design(4) - Google Docs
2017 夏季3班 理论课正式教案
File Edit View Insert Format Tools Table Add-ons Help See new changes
jennifer@laioffer.com
Comments
Share
Editing
Outline
上课语音会议链接
重要：2017秋季来Offer校园行报名方法...
Homework Solutions
访问已经上过的大班课程 存档
报名回答问题同学list
Class 32 System Design 4
System Design: Web Applications I
Web Application: Design a Web Crawler
2. Design a feed product
queues. Then, the crawled page will produce more urls as candidates to crawl, the crawler will enqueue those candidates to their corresponding queues based on the same url sharding.
Each crawler also maintain a local visited page copy for pages it has visited and check to see if it needs to crawler the claimed url.
Q. What if we do not want the queue machine (but a more distributed architecture)?
2. Design a feed product
Requirements:
1. each user could have 1~1000 friends, can get stories from your friends in real time.
2. scalability:
- millions of query per second (qps)
- a lot of data to store, cannot be stored on one machine, needs distributed system
3. can do ranking in real time
How to figure out the architecture:
What's the request and response, input and output?
How many steps does the system need to go through to process the input/requests?
Coding Pad: Link
From 1705070 Ye Wang to Everyone
查找story，排序，返回story


docs.google.com
2017 夏季3班 理论课正式教案 - Google Docs
OOD(4) + System Design(4) - Google Docs
2017 夏季3班 理论课正式教案
File Edit View Insert Format Tools Table Add-ons Help All changes saved in Drive
jennifer@laioffer.com
Comments
Share
Editing
Outline
上课语音会议链接
重要：2017秋季来Offer校园行报名方法...
Homework Solutions
访问已经上过的大班课程 存档
报名回答问题同学list
Class 32 System Design 4
System Design: Web Applications I
Web Application: Design a Web Crawler
2. Design a feed product
What is being pushed?
Two models:
1. push model
2. pull model
Push model:
What is being pushed?
when there is a story happened, it's pushed to all of the friends.
e.g., we have u1, u2, u3, u1 is friend of u2 and u3.
Each user maintain a list of friends' stories that are eligible to be displayed to himself/herself:
u1 -> s1, s2
u2 -> s1, s3, s4
u3 -> s3, s5
We call this structure an inverted index. Also needs a forward index to store information for each story, e.g., click rate for each story, etc., mainly for the purpose of ranking:
s1 -> info for s1
s2 -> info for s2
s3 -> info for s3
s4 -> info for s4


docs.google.com
2017 夏季3班 理论课正式教案 - Google Docs
OOD(4) + System Design(4) - Google Docs
2017 夏季3班 理论课正式教案
File Edit View Insert Format Tools Table Add-ons Help See new changes
jennifer@laioffer.com
Comments
Share
Editing
Outline
上课语音会议链接
重要：2017秋季来Offer校园行报名方法...
Homework Solutions
访问已经上过的大班课程 存档
报名回答问题同学list
Class 32 System Design 4
System Design: Web Applications I
Web Application: Design a Web Crawler
2. Design a feed product
What is being pushed?
so speed is slow
Pull model:
Each user maintains a list of stories of his own
For example, u1 has story s1, s2, s3 of his own, u2 has s4, s5, s6 of his own, u3 has s7, s8, s9 of his own, the inverted index will look like:
u1 -> s1, s2, s3 ...
u2 -> s4, s5, s6 ...
u3 -> s7, s8, s9 ...
Similarly, we still need the forward index to store information for each story
s1 -> info for s1
s2 -> info for s2
s3 -> info for s3
s4 -> info for s4
Now, u1 comes to the site, and she is friend of u2 and u3. We will send request to fetch stories of her friend, u2 and u3. In the example, we will get s4, s5 and s6 from u2, and s7, s8 and s9 from u3. Then we merge them into s4, s5, s6, s7, s8, s9 to display.
Coding Pad: Link



Now, each index server only need to store all information of stories sharded to itself. e.g., index server 1 only store s1, s4, s7. After getting request from aggregator to fetch stories for u2 and u3 (who are u1's friend), index server 1 do 3 things:
1. get s4 and s7 from inverted index,
2. fetch ranking info for s4 and s7 and rank them, say, the rank result is (s7, s4)
3. return (s7, s4) to aggregator

Aggregator receives the partially ranked list from n ver 1,2,3, it will do a merge sort finally and return it to browser to display.

Cons:

From 1704148 yujie Cao to Everyone
挺清楚地