网络爬虫及其分布式应用

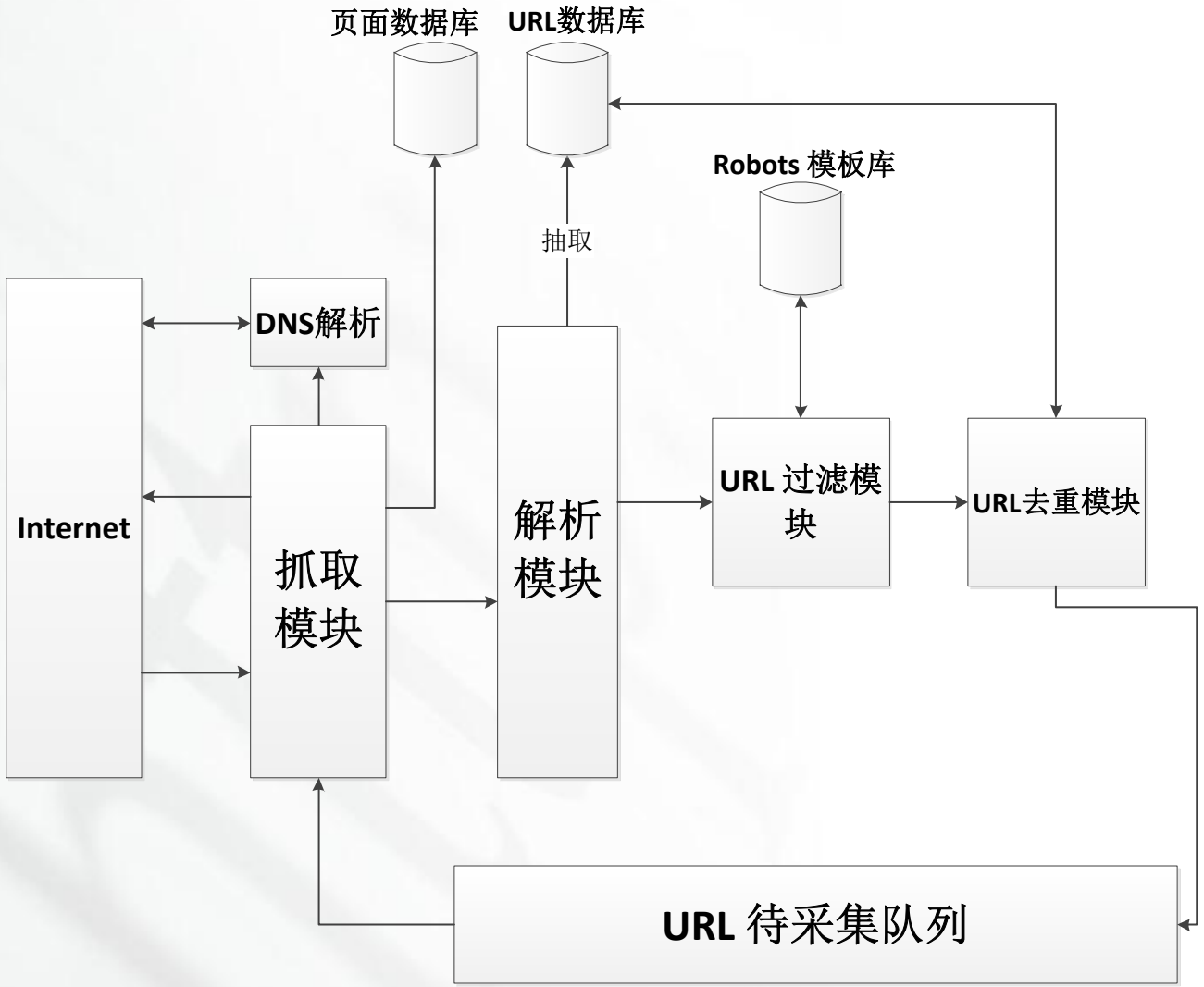
演讲者：张立鑫

什么是网络爬虫

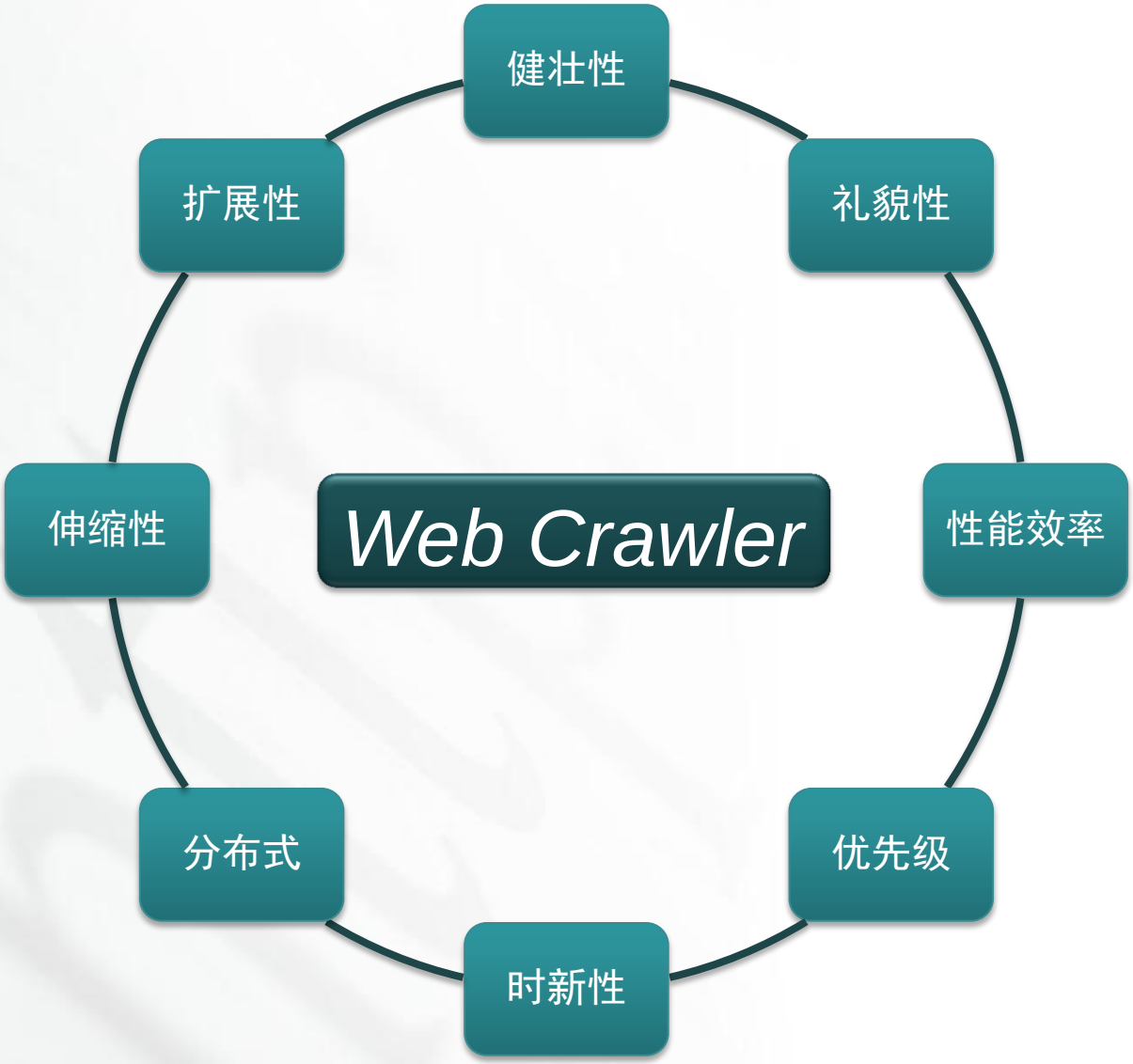
网络爬虫是搜索引擎系统中十分重要的组成部分，它负责从互联网中搜集网页，采集信息，这些网页信息用于建立索引从而为搜索引擎提供支持，它决定着整个引擎系统的内容是否丰富，信息是否即时，因此其性能的优劣直接影响着搜索引擎的效果。

网络爬虫的基本工作原理：
1）从一个初始URL集合中挑选一个URL，下载该URL对应的页面；
2）解析该页面，从该页面中抽取出其包含的URL集合，接下来将抽取的URL集合再添加到初始URL集合中；
3）重复前两个过程，直到爬虫达到某种停止标准为止。

网络爬虫的基本组成结构与工作流程



网络爬虫应该提供的功能特性



拒绝蜘蛛协议（REP）

拒绝蜘蛛协议
（Robots Exclusion Protocol）

通过在网站根目录下放置robots.txt（统一小写）来告诉网络爬虫，此网站中的哪些内容是不能或不应该被搜索引擎的爬虫获取的，哪些是可以被网络爬虫获取的。robots.txt协议并不是一个规范，而只是约定俗成的，所以并不能保证网站的隐私。

robots.txt包含的基本规则如下：

1、User-agent ：爬虫名字的
2、Disallow：不允许爬取的
3、Allow：允许爬取的
4、Crawl-delay：爬取时间间隔
5、#：注释
6、通配符使用：“$”匹配行结束符，“*”匹配0或多个任意字符。

举例：

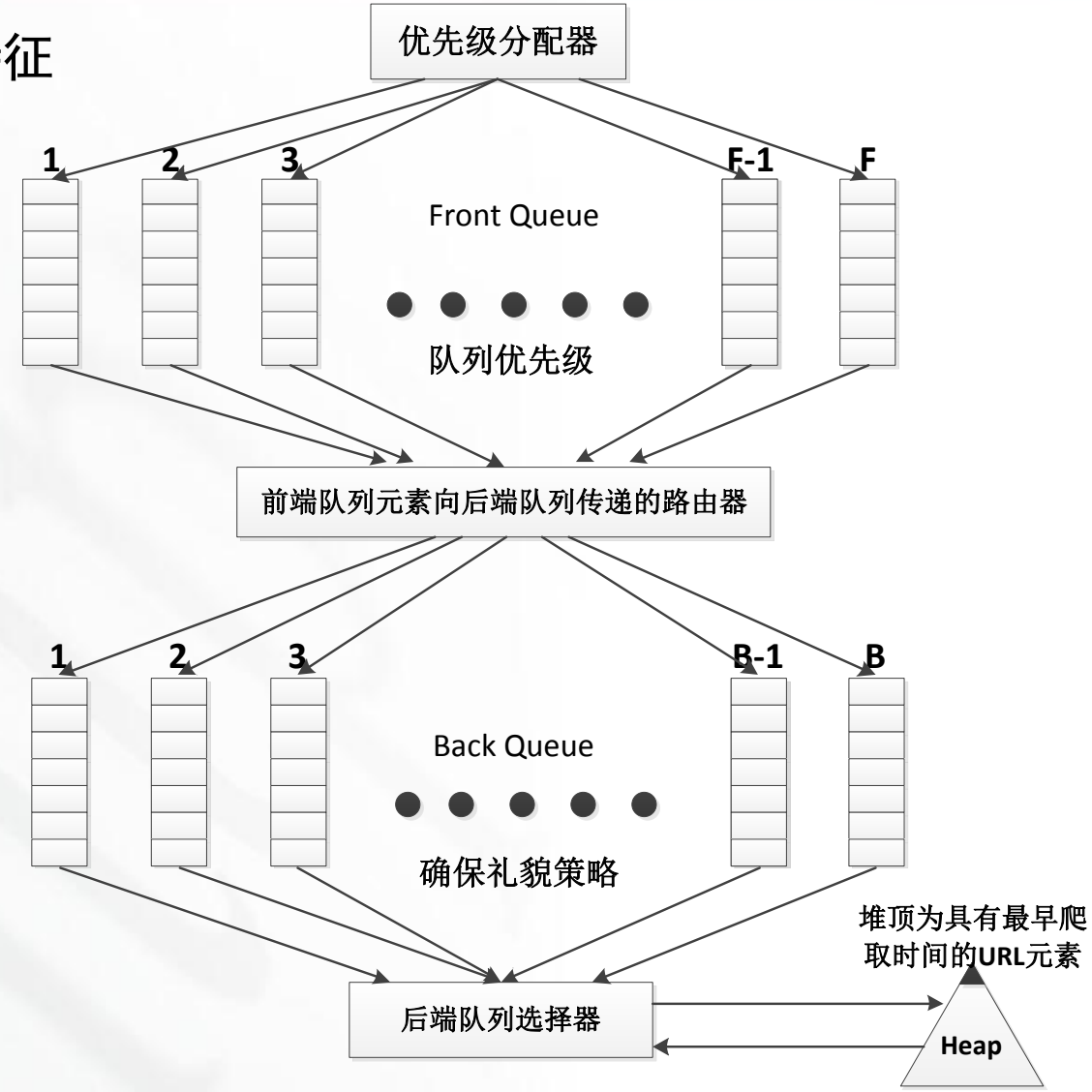
User-agent: * 这里的*代表的所有的搜索引擎种类，*是一个通配符
User-agent: baiduspider 表示以下策略针对名为baiduspider的网络爬虫
Disallow: /ABC/ 这里定义是禁止爬寻ABC目录下面的目录
Disallow: /abc/*.htm 禁止访问/abc/目录下的所有以".htm"为后缀的URL(包含子目录)。
Disallow: /*?* 禁止访问网站中所有的动态页面
Disallow: /mp3$ 禁止抓取网页所有的以.mp3后缀结尾的资源文件
Allow: .htm$ 仅允许访问以".htm"为后缀的URL
Allow: .gif$ 允许抓取网页和gif格式图片
Crawl-delay: 10在对本主机服务器上次访问结束后需要等待10秒钟才可以进行下一次的访问请求

URL待采集队列设计（Mercator）

具有优先级，礼貌性特征

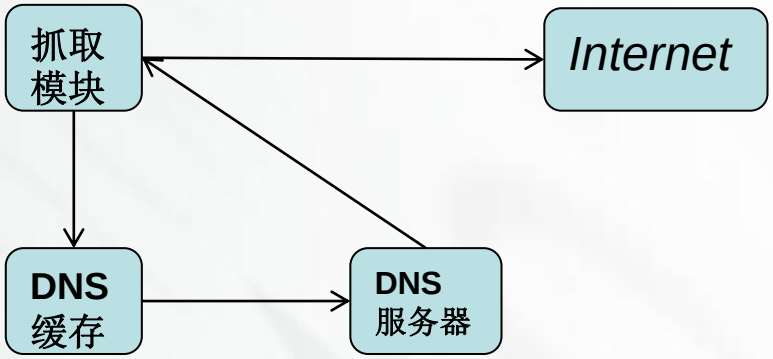
主机	Back队列序号
www.baidu.com	23
www.google.com	25
www.seu.edu.cn	17

主机名->Back队列序号



页面抓取模块

页面抓取模块是网络爬虫系统中与互联网交互最频繁的模块。它主要通过HTTP协议与网络服务器通信。



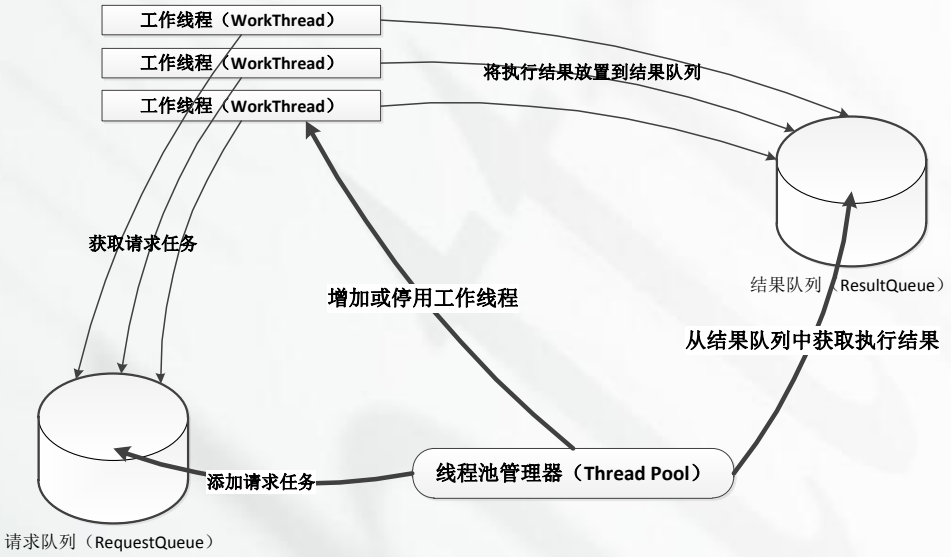
响应码	含义	举例
1xx	信息，请求收到，继续处理	100（继续）
2xx	成功，行为被成功地接受、理解和采纳	200（成功）
3xx	重定向，为了完成请求，须对重定向地址再次发起请求	301，302（重定向到其他位置）
4xx	客户端错误，请求包含语法错误或者请求无法实现	404（找不到资源文件），403（禁止）
5xx	服务器错误，服务器不能实现一种明显无效的请求	500（内部服务器错误）

HTTP响应码

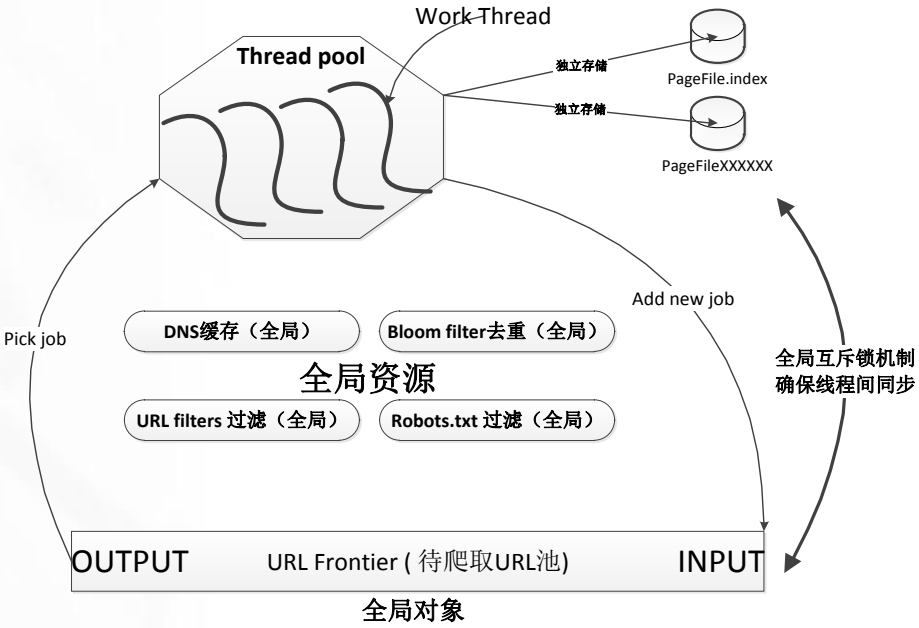
基于线程池的多线程页面采集

线程池

尽可能减少创建和销毁对象的次数，特别是一些很耗资源的对象创建和销毁，这样就引入了“池”的概念，“池”的概念使得人们可以定制一定量的资源，然后对这些资源进行复用，而不是频繁的创建和销毁。



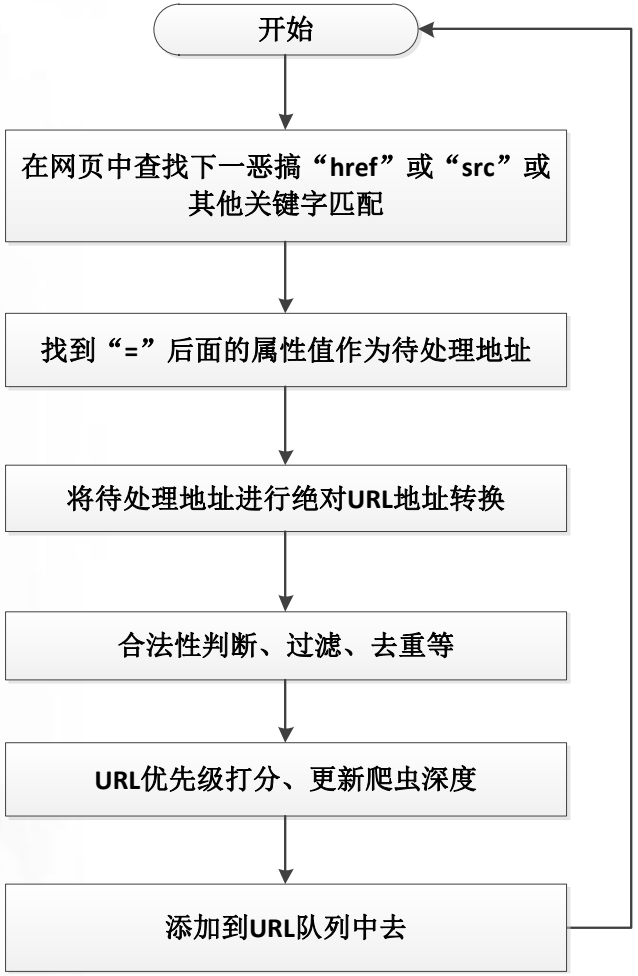
简单的线程池工作模型



网络爬虫中线程池工作模型

页面解析模块

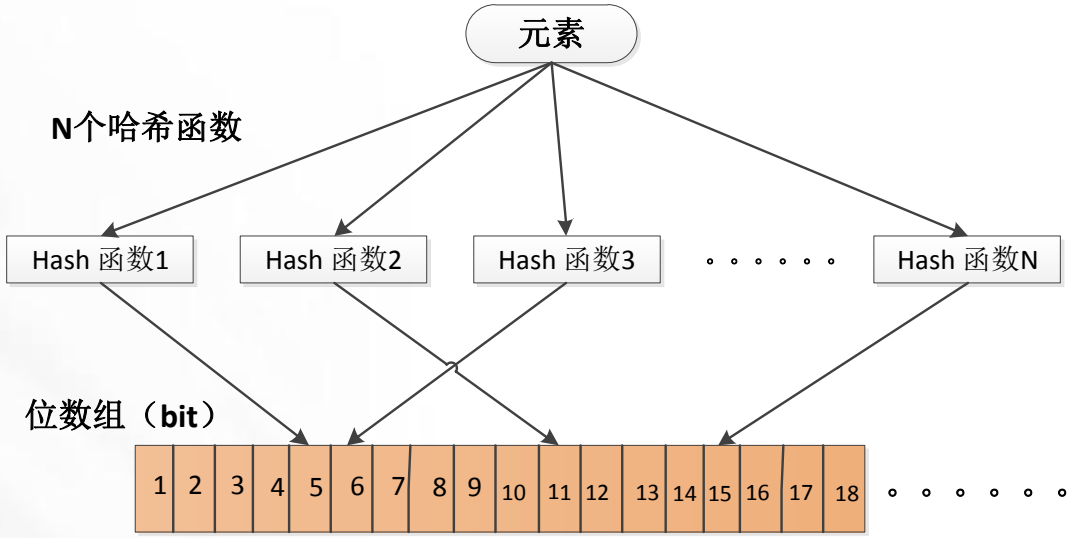
对于URL的提取，应该保证有较好的容错性、能全面的提取并且要保证提取的算法具有较高的效率。



一种简单的URL提取方案

URL去重

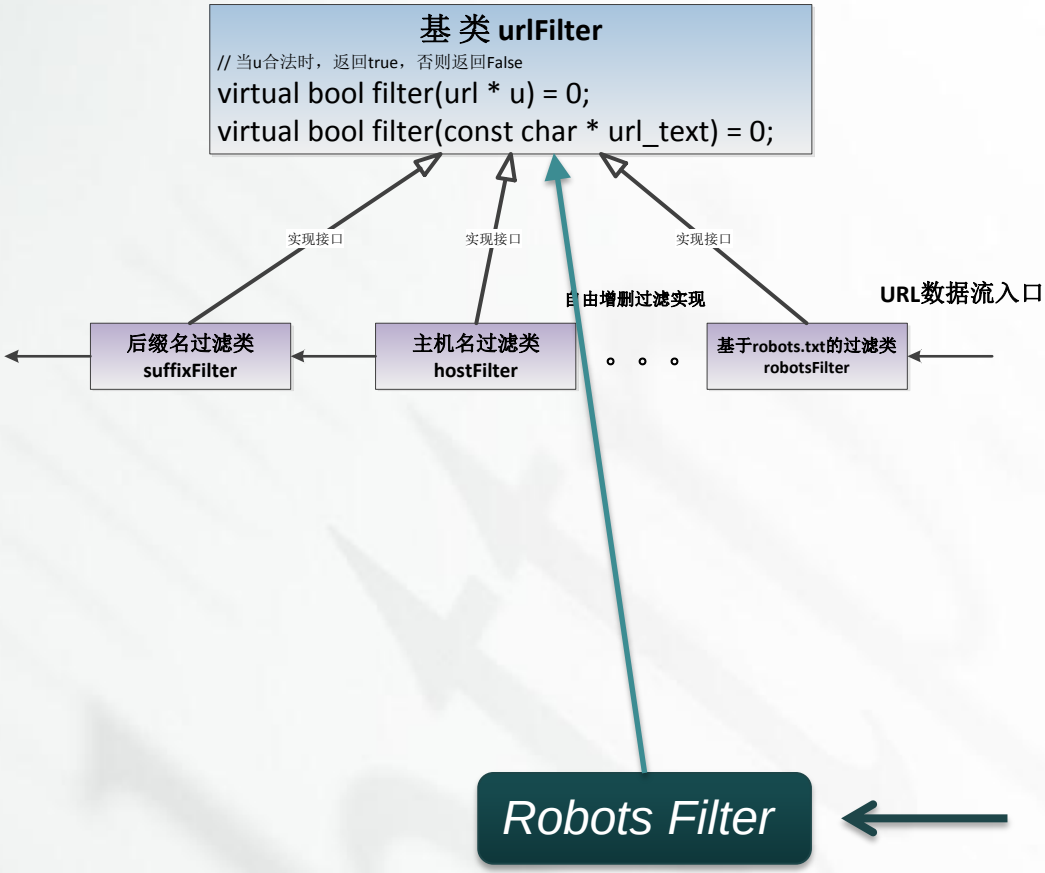
Bloom Filter是一种空间效率很高的随机数据结构，它利用位数组很简洁地表示一个集合，并能判断一个元素是否属于这个集合。Bloom Filter的这种高效是有一定代价的：在判断一个元素是否属于某个集合时，有可能会把不属于这个集合的元素误认为属于这个集合（false positive）。因此，Bloom Filter不适合那些“零错误”的应用场合。而在能容忍低错误率的应用场合下，Bloom Filter通过极少的错误换取了存储空间的极大节省。



Bloom Filter 示意图

```
Get:arr[code/CHARBITSIZE] & (1<<(code%CHARBITSIZE))
Set:arr[code/CHARBITSIZE] |= (1<<(code%CHARBITSIZE));
```

URL过滤器



拒绝蜘蛛协议（REP）

robot.txt:

1、**User-agent**: 该项的值用于描述搜索引擎robot的名字。

2、**Disallow**: 该项的值用于描述不希望被访问的一组URL，这个值可以是一条完整的路径，也可以是路径的非空前缀，以Disallow项的值开头的URL不会被 robot访问。

3、**Allow**: 该项的值用于描述希望被访问的一组URL，与Disallow项相似，这个值可以是一条完整的路径，也可以是路径的前缀，以Allow项的值开头的URL 是允许robot访问的。

4、**Crawl-delay**: 设置为整数值，表示在对本主机服务器上次访问结束后需要等待多长时间才可以进行下一次的访问请求。 (网络爬虫的礼貌策略)

5、**#:** 一些robots.txt 会有注释，#后面都是注释内容，处理时需要过滤掉。

6、**通配符使用**： “$” 匹配行结束符， “*” 匹配0或多个任意字符。

DNS解析

DNS解析是网络爬虫的瓶颈问题之一。

1、请求需多次DNS主机间转发，非常耗时

2、gethostbyname() 系统api一般都是同步通信机制

改善方法：

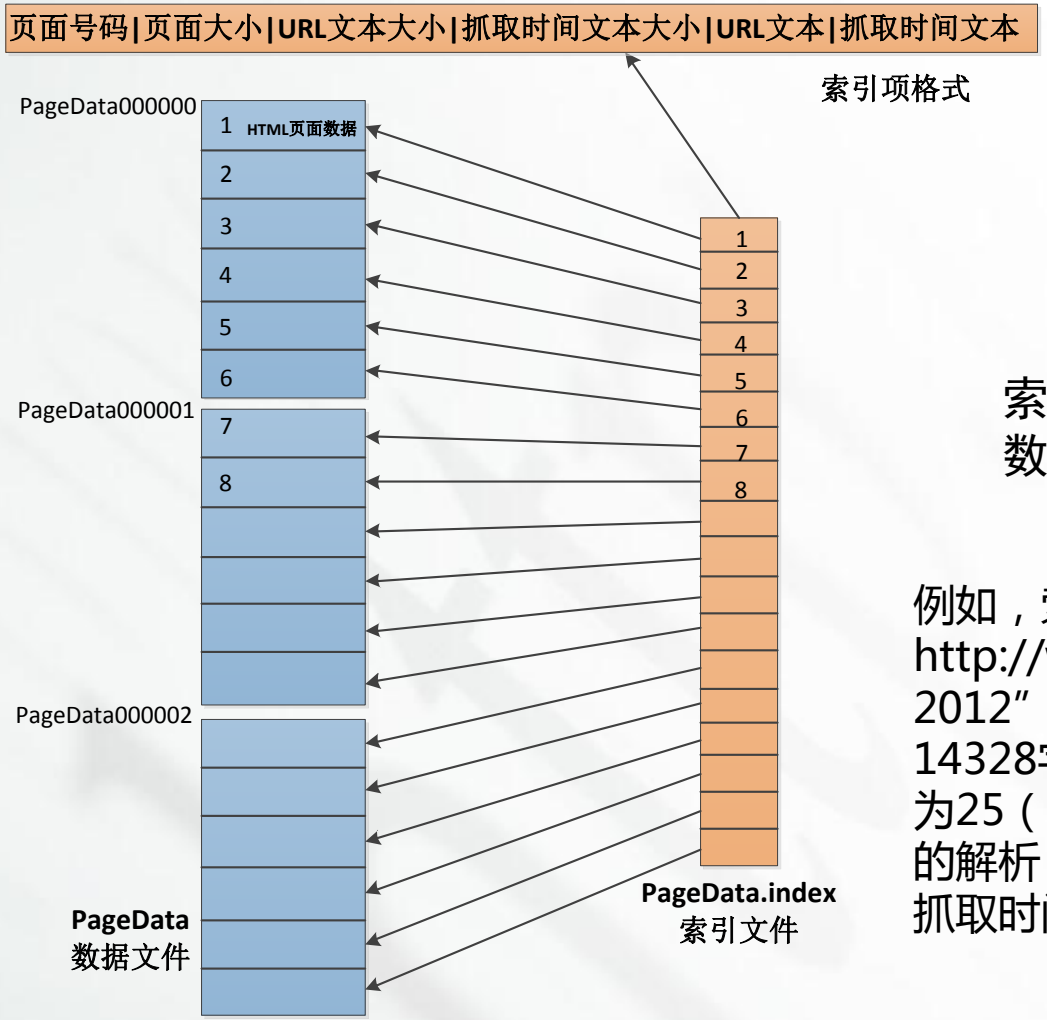
1、使用DNS缓存

2、为每一个爬取后的URL附加一个IP地址属性

3、使用异步DNS查询通信机制

4、多线程模型

页面数据存储



在无附加条件的情况下，为保证较高的存储效率，弃用传统关系型数据库，改用基于纯文件机制的存储模型。

索引文件：主要存储下载页面的URL
数据文件：主要存储下载页面的数据文件

例如，索引项记录“6 14328 22 25 http://www.seu.edu.cn Fri Mar 30 08:31:34 2012”表示该页面号码为60，页面数据大小为14328字节，URL文本大小为22，时间文本大小为25（这两个文本大小主要便于对后面真实文本的解析），URL文本为 http://www.seu.edu.cn，抓取时间为Fri Mar 30 08:31:34 2012。

分布式网络爬虫

为什么进行网络爬虫的分布式研究：

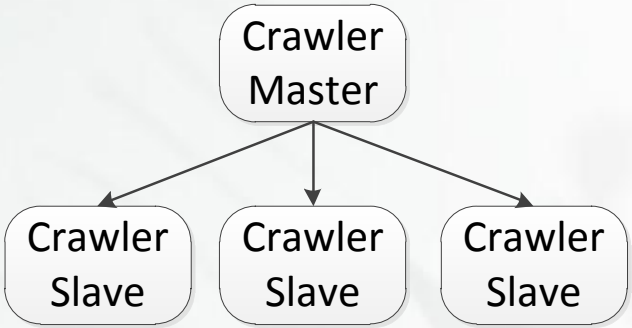
1、使用多台爬虫主机进行信息采集，可以将爬虫机器放在待采集的网站附近，这是因为地理上的距离会影响网络传输的效率。长距离的网络连接会具有较低的吞吐量和较长的延迟时间。

2、单机爬虫需要处理对所有的目标主机的信息采集，这就要维护很多的中间数据，比如一些中间缓存、URL队列数据、链接检查等等，为减轻单机的负担，应该按某种分配策略将一部分任务分配给其他爬虫机器，这样可以降低存储和检查的开销。

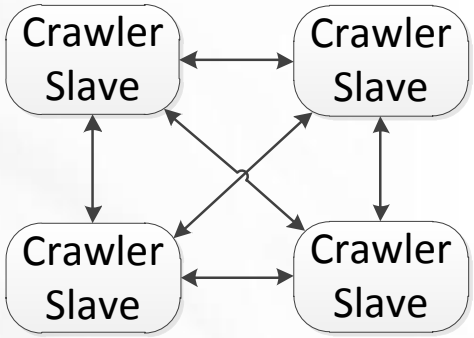
3、可以使用大量的计算资源，包括CPU、网络带宽、物理磁盘等等。

分布式网络爬虫架构设计

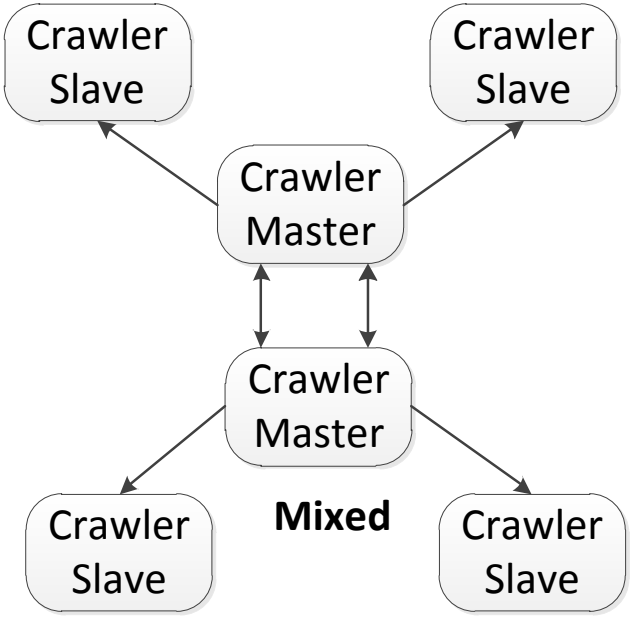
常见的简单分布式架构模型：



Master—Slave



Peer to Peer



Mixed

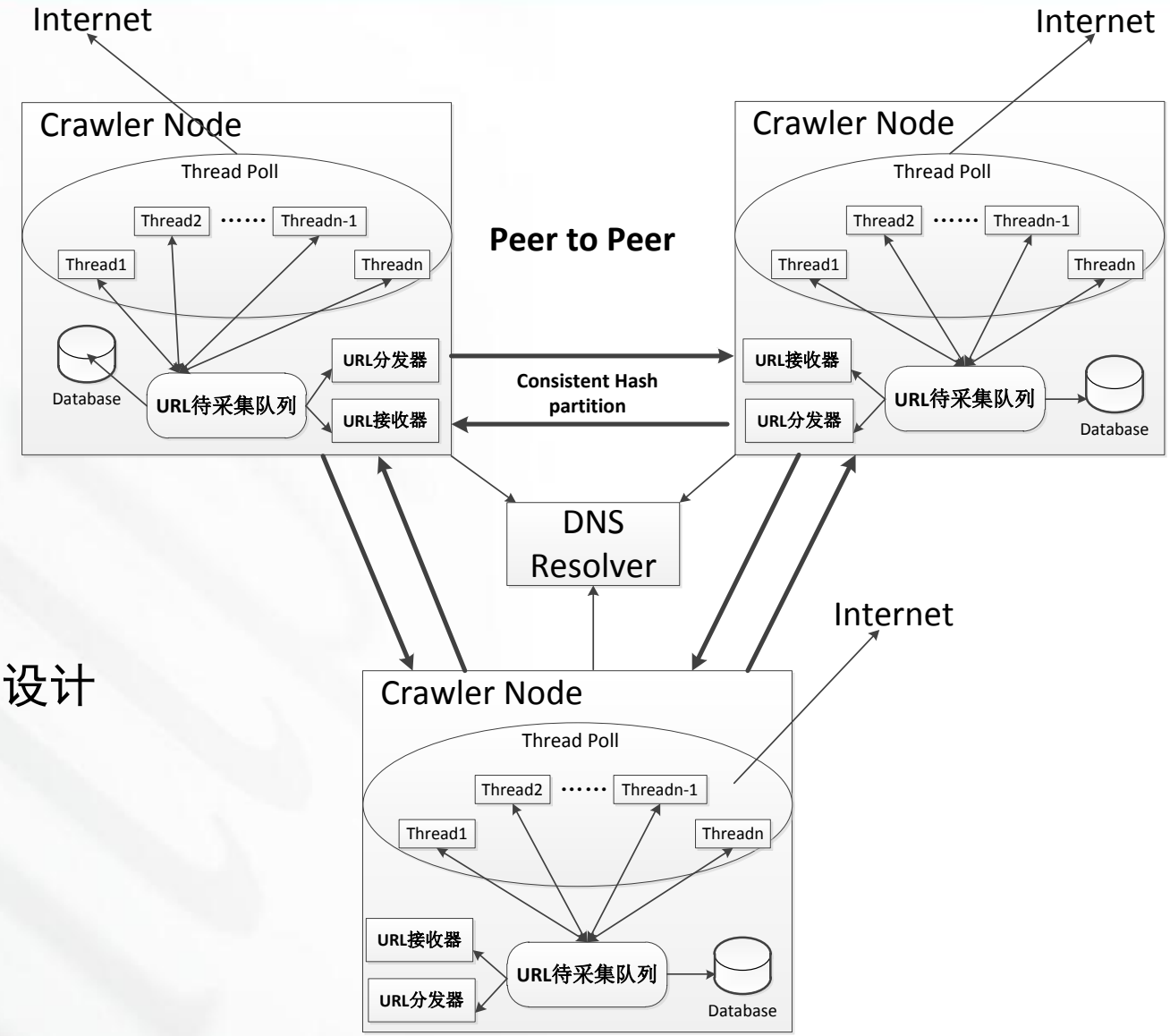
架构简单
扩展时只需更新Master
Master节点压力大，易成为瓶颈
Slave节点数量限制大

架构稍复杂，所有节点进行通信
扩展时需要更新所有其他节点
无Master，不会出现单机热点
Slave节点数量限制小

架构复杂，实现难度大
Master与Master进行通信
Slave节点数量限制小

分布式网络爬虫架构设计

采用P2P架构模型的
分布式网络爬虫架构设计



爬虫节点间资源划分策略

URL（统一资源定位符）

http://www.seu.edu.cn:80/news/abc.html?id=3

传送协议　服务器地址　端口号　路径　资源文件名（查询）

划分粒度

- URL文本整体
- URL的主机名（IP地址）
- 逻辑映射表

分布式网络爬虫系统近似一个缓存系统，为什么？

每个爬虫节点维护一定范围内的
URL的DNS缓存、去重缓存、robots.txt的缓存等等

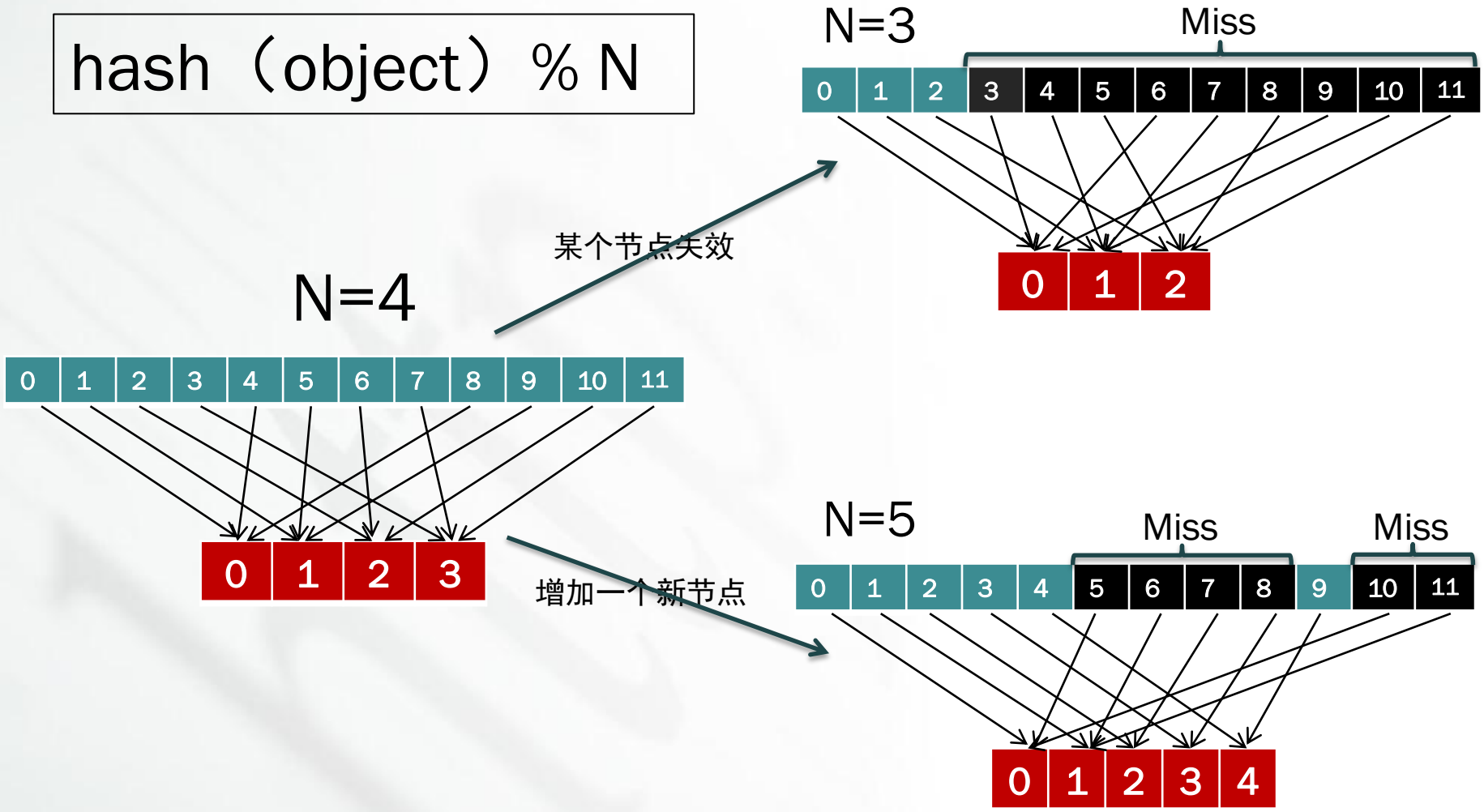
Hash算法需要满足

- 负载平衡性（Load Balance）
  哈希的结果能够尽可能平衡或按比例分布到所有的节点中去
- 单调性(Monotonicity)
  当减少节点时，不会产生大量的hit miss；
  当增加节点时，部分资源会映射到新的节点，分摊压力
- 一致性(Consistent)
  所有节点维护着相同的映射关系，同一个资源不会被映射到不同的节点中去

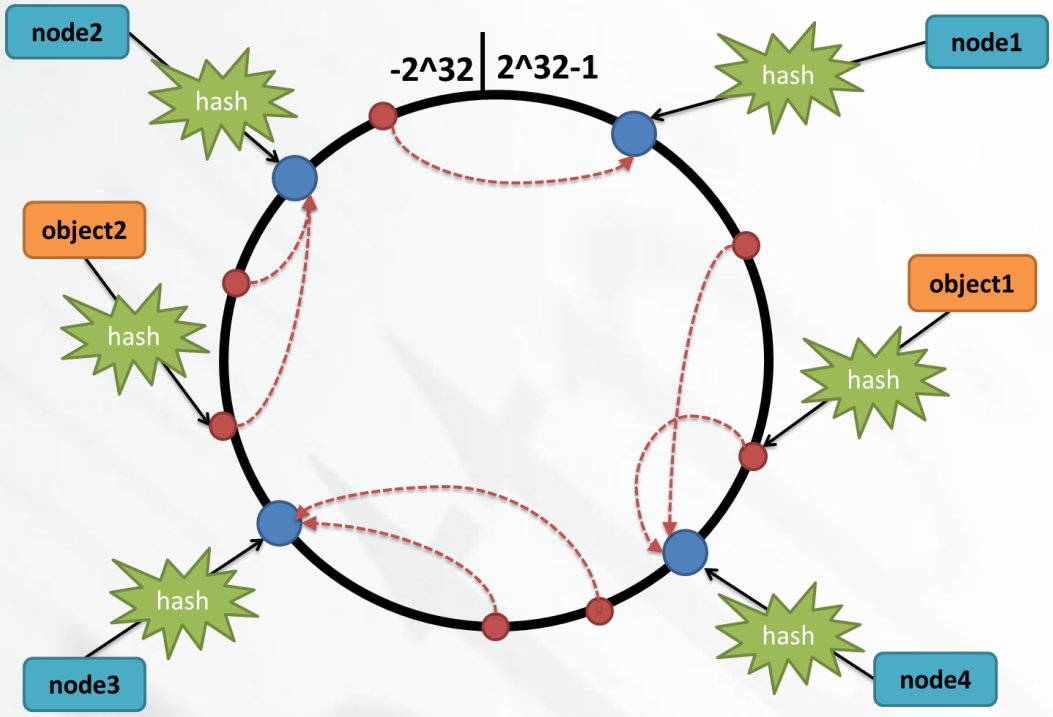
简单哈希算法的劣势

简单哈希函数：

hash（object）% N



一致性哈希算法



一致性哈希算法图示

已解决：

单节点损坏，只会造成该节点上游的资源对象的映射变化，不会影响其他机器节点。

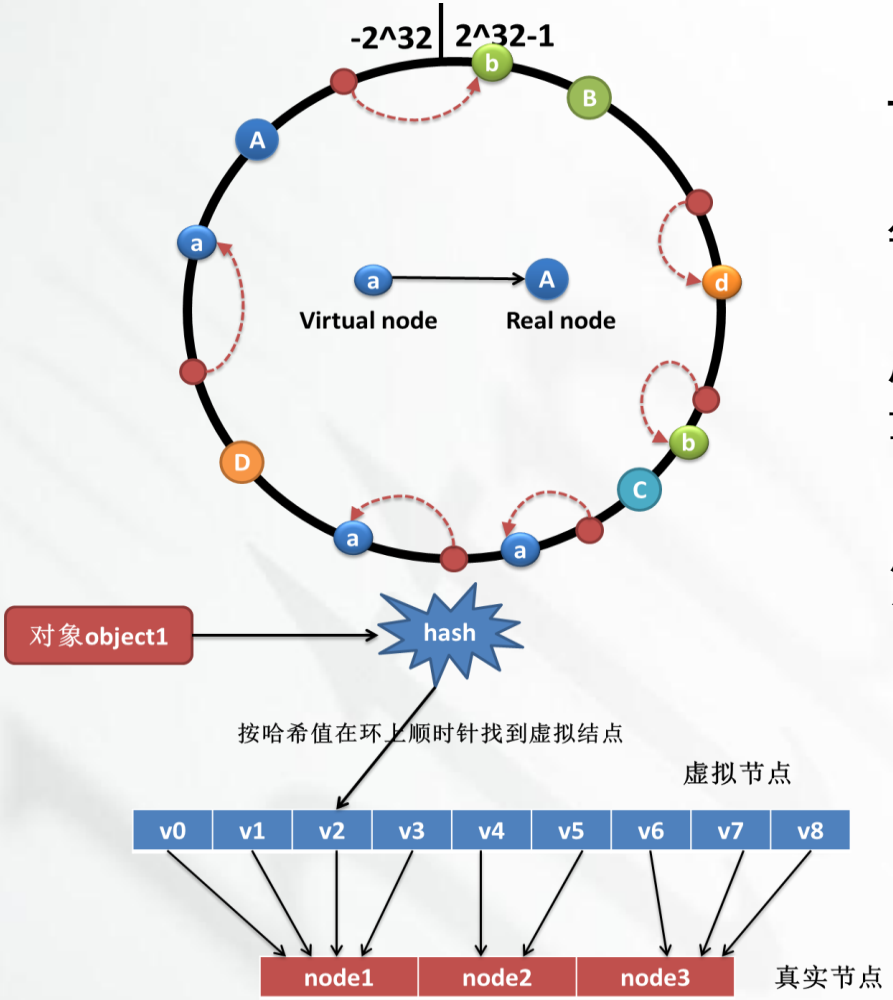
在相邻两节点间增加新节点，会分摊这两个节点间的压力。

仍需进一步解决：

如何负载均衡，均摊压力？

如何增、删节点时，压力会均摊到所有节点，而不会产生热点问题？

引入“虚拟节点”的一致性哈希算法



引入“虚拟节点”的一致性哈希算法示意图

一个真实节点对应N（N>=1）个虚拟节点

每个虚拟节点有一个指向真实节点的指针

虚拟节点的个数称为“复本个数”，其数量一定意义上代表真实节点的权重。

虚拟节点的个数比决定真实节点的压力分摊比。

爬虫节点间的通信机制

Json格式：是一种轻量级的数据交换格式。键值对。
流行，容易扩展，便于不同系统间进行数据交换。

URL数据结构

属性	值
HOST	www.seu.edu.cn
FILE	/news/123456.html
PORT	80
PRIORITY	20
DEAPTH	6
IPADDR	121.248.63.50

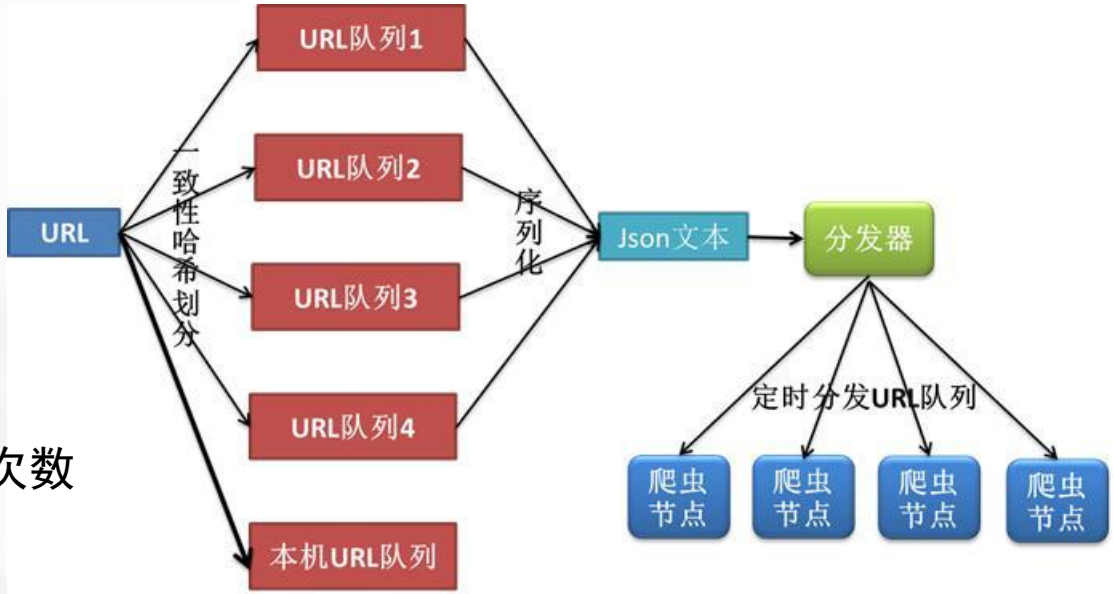
URL序列化后的JSON文本

{“HOST”:”www.seu.edu.cn”,”FILE”:
”/news/123456.html”,”PORT”:80,”
PRIORITY”:20,”DEAPTH”:6,”IPADDR
”:”121.248.63.50”}

基于JSON格式的URL序列化举例

爬虫节点间的通信机制

URL分发器



定时分发和接收，减少通信次数

URL接收器

实验

实验环境

系统实验参数	值
外网通信带宽最大值	500KB/S
爬虫线程数	20个
爬虫深度	2
种子URL	http://www.hao123.com
内存	2G DDR2 800
爬行时间	60分钟
爬虫节点间通信最大带宽	10M/S
操作系统	linux (Ubuntu 10.04 LTS)
编译环境	GCC

爬虫单节点机器基本配置

爬虫节点	IP地址	端口号	节点机器名	节点权重（即虚拟节点个数）
A	192.168.1.101	2001	machineA	150
B	192.168.1.102	2002	machineB	80
C	192.168.1.103	2003	machineC	200
D	192.168.1.104	2004	machineD	40

分布式多节点参数配置

实验

HTTP响应	数量	百分比
200(成功)	17798	90.23%
404(页面丢失)	981	4.97%
30X(重定向)（不算入总和）	1340	6.79%
重定向过多	34	0.17%
连接超时	537	2.72%
其他情况	373	1.89%
总和	19732	100%

HTTP请求响应结果统计表

No.	主机名	采集页面数
1	www.hao123.com	624
2	www.zol.com.cn	523
3	nanjing.baixing.com	498
4	www.pcauto.com.cn	489
5	www.iqiyi.com	487
6	www.app111.com	478
7	tv.sohu.com	471
8	www.taobao.com	449
9	www.ifeng.com	446
10	sports.163.com	445

爬虫对各服务器主机的采集情况

对爬虫节点A进行了统计。爬虫节点A共运行时间60分钟，共发出请求19723个爬行请求。爬虫成功爬取到的URL数量为17798个，共访问域名主机117个，下载数据文件大小1594.588MB。下载数据平均占用带宽450kb/s，上传数据平均占用带宽24kb/s。

实验

爬虫节点	A	B	C	D	原始比	实际比
A	0	4480	9840	2254	1:2.5:0.5	1:2.20:0.50
B	4622	0	6098	1285	1:1.33:0.27	1:1.32:0.28
C	9473	5551	0	725	1:0.53:0.27	1:0.59:0.08
D	6167	3269	10349	0	1:0.53:1.33	1:0.53:1.68

前十分钟各节点间URL分发情况

删除结点C

爬虫节点	A	B	D	原始比	实际比
A	0	1267	500	1:0.5	1:0.39
B	1211	0	87	1:0.27	1:0.07
D	1796	980	0	1:0.53	1:0.55

删除节点C后中间三分钟的各节点分发情况

新增结点E，虚拟结点个数为300

爬虫节点	A	B	D	E	原始比	实际比
A	0	16481	9331	51002	1:0.5:3.75	1:0.57:3.09
B	26225	0	6138	42859	1:0.27:2	1:0.23:1.63
D	9570	3992	0	22203	1:0.53:2	1:0.42:2.32
E	21221	14326	5112	0	1:0.53:0.27	1:0.68:0.24

增加新节点E后的各节点URL分发情况

爬虫节点	节点机器名	虚拟节点个数
A	machineA	150
B	machineB	80
C	machineC	200
D	machineD	40

分布式多节点初始参数配置


谢谢