[置顶] 每天进步一点点——五分钟理解一致性哈希算法(consistent hashing)

标签： 算法 分布式

2014-04-11 00:21 260919人阅读 评论(94) 收藏 举报

分类：

OpenStack（21）

一致性哈希算法在1997年由麻省理工学院提出的一种分布式哈希（DHT）实现算法，设计目标是为了解决因特网中的热点(Hot spot)问题，初衷和CARP十分类似。一致性哈希修正了CARP使用的简 单哈希算法带来的问题，使得分布式哈希（DHT）可以在P2P环境中真正得到应用。

一致性hash算法提出了在动态变化的Cache环境中，判定哈希算法好坏的四个定义：

1、平衡性(Balance)：平衡性是指哈希的结果能够尽可能分布到所有的缓冲中去，这样可以使得所有的缓冲空间都得到利用。很多哈希算法都能够满足这一条件。

2、单调性(Monotonicity)：单调性是指如果已经有一些内容通过哈希分派到了相应的缓冲中，又有新的缓冲加入到系统中。哈希的结果应能够保证原有已分配的内容可以被映射到原有的或者新的缓冲中去，而不会被映射到旧的缓冲集合中的其他缓冲区。

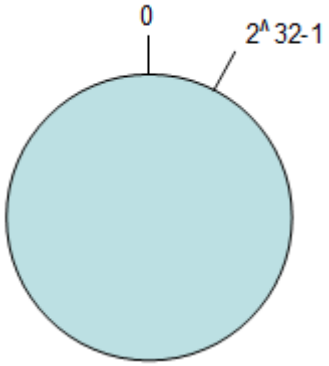
3、分散性(Spread)：在分布式环境中，终端有可能看不到所有的缓冲，而是只能看到其中的一部分。当终端希望通过哈希过程将内容映射到缓冲上时，由于不同终端所见的缓冲范围有可能不同，从而导致哈希的结果不一致，最终的结果是相同的内容被不同的终端映射到不同的缓冲区中。这种情况显然是应该避免的，因为它导致相同内容被存储到不同缓冲中去，降低了系统存储的效率。分散性的定义就是上述情况发生的严重程度。好的哈希算法应能够尽量避免不一致的情况发生，也就是尽量降低分散性。

4、负载(Load)：负载问题实际上是从另一个角度看待分散性问题。既然不同的终端可能将相同的内容映射到不同的缓冲区中，那么对于一个特定的缓冲区而言，也可能被不同的用户映射为不同 的内容。与分散性一样，这种情况也是应当避免的，因此好的哈希算法应能够尽量降低缓冲的负荷。

在分布式集群中，对机器的添加删除，或者机器故障后自动脱离集群这些操作是分布式集群管理最基本的功能。如果采用常用的hash(object)%N算法，那么在有机器添加或者删除后，很多原有的数据就无法找到了，这样严重的违反了单调性原则。接下来主要讲解一下一致性哈希算法是如何设计的：

环形Hash空间

按照常用的hash算法来将对应的key哈希到一个具有2^32次方个桶的空间中，即0~(2^32)-1的数字空间中。现在我们可以将这些数字头尾相连，想象成一个闭合的环形。如下图



把数据通过一定的hash算法处理后映射到环上

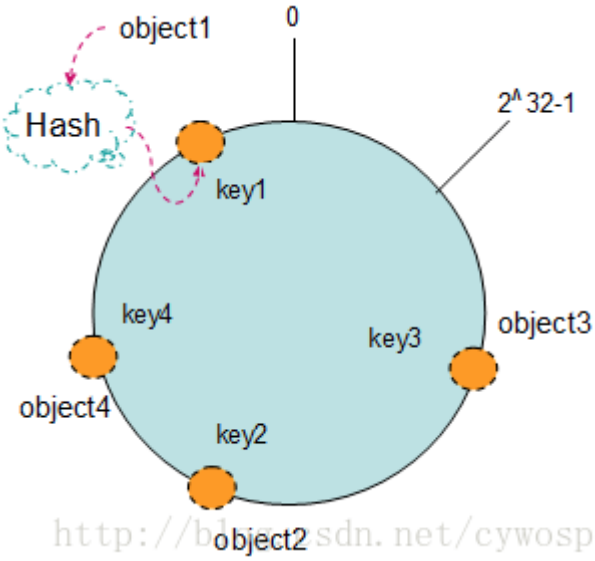
现在我们将object1、object2、object3、object4四个对象通过特定的Hash函数计算出对应的key值，然后散列到Hash环上。如下图：

Hash(object1) = key1；

Hash(object2) = key2；

Hash(object3) = key3；

Hash(object4) = key4；



将机器通过hash算法映射到环上

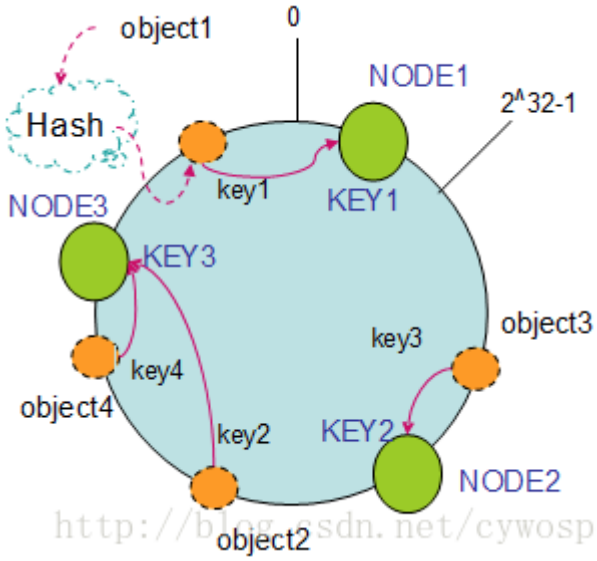
在采用一致性哈希算法的分布式集群中将新的机器加入，其原理是通过使用与对象存储一样的Hash算法将机器也映射到环中（一般情况下对机器的hash计算是采用机器的IP或者机器唯一的别名作为输入值），然后以顺时针的方向计算，将所有对象存储到离自己最近的机器中。

假设现在有NODE1，NODE2，NODE3三台机器，通过Hash算法得到对应的KEY值，映射到环中，其示意图如下：

Hash(NODE1) = KEY1;

Hash(NODE2) = KEY2;

Hash(NODE3) = KEY3;



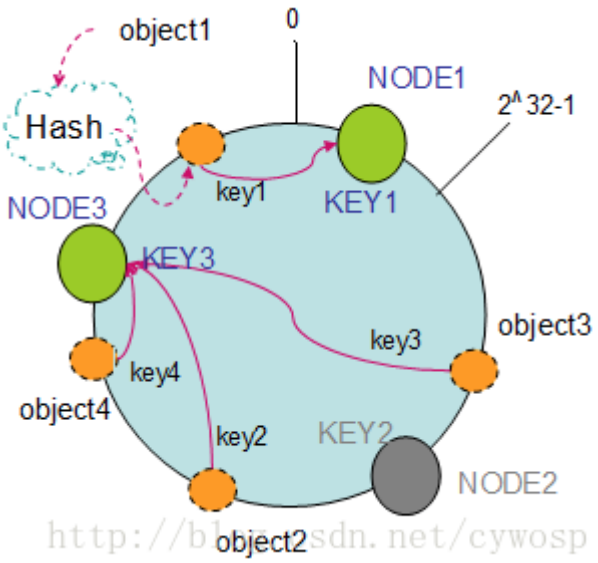
通过上图可以看出对象与机器处于同一哈希空间中，这样按顺时针转动object1存储到了NODE1中，object3存储到了NODE2中，object2、object4存储到了NODE3中。在这样的部署环境中，hash环是不会变更的，因此，通过算出对象的hash值就能快速的定位到对应的机器中，这样就能找到对象真正的存储位置了。

机器的删除与添加

普通hash求余算法最为不妥的地方就是在有机器的添加或者删除之后会照成大量的对象存储位置失效，这样就大大的不满足单调性了。下面来分析一下一致性哈希算法是如何处理的。

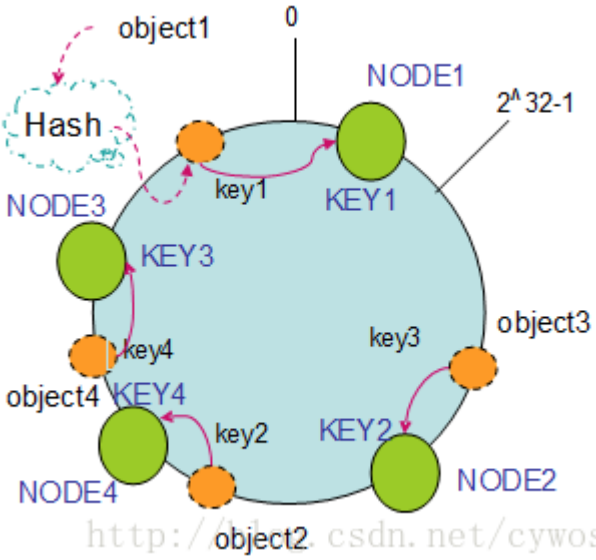
1. 节点（机器）的删除

以上面的分布为例，如果NODE2出现故障被删除了，那么按照顺时针迁移的方法，object3将会被迁移到NODE3中，这样仅仅是object3的映射位置发生了变化，其它的对象没有任何的改动。如下图：



2. 节点（机器）的添加

如果往集群中添加一个新的节点NODE4，通过对应的哈希算法得到KEY4，并映射到环中，如下图：



通过按顺时针迁移的规则，那么object2被迁移到了NODE4中，其它对象还保持这原有的存储位置。通过对节点的添加和删除的分析，一致性哈希算法在保持了单调性的同时，还是数据的迁移达到了最小，这样的算法对分布式集群来说是非常合适的，避免了大量数据迁移，减小了服务器的的压力。

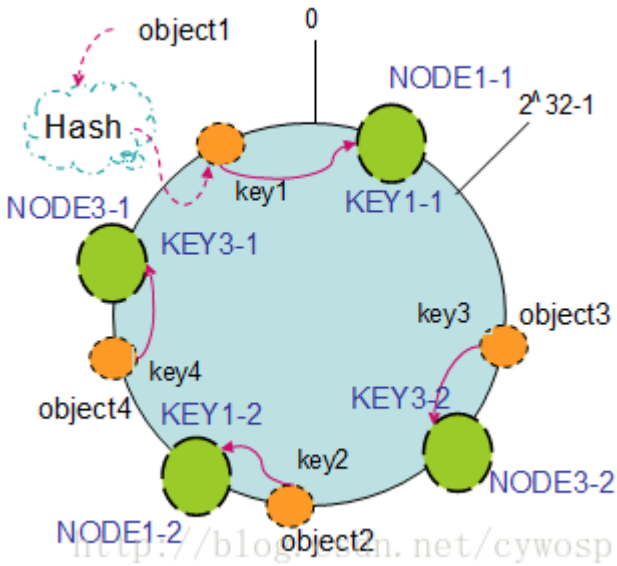
平衡性

根据上面的图解分析，一致性哈希算法满足了单调性和负载均衡的特性以及一般hash算法的分散性，但这还并不能当做其被广泛应用的原由，因为还缺少了平衡性。下面将分析一致性哈希算法是如何满足平衡性的。hash算法是不保证平衡的，如上面只部署了NODE1和NODE3的情况（NODE2被删除的图），object1存储到了NODE1中，而

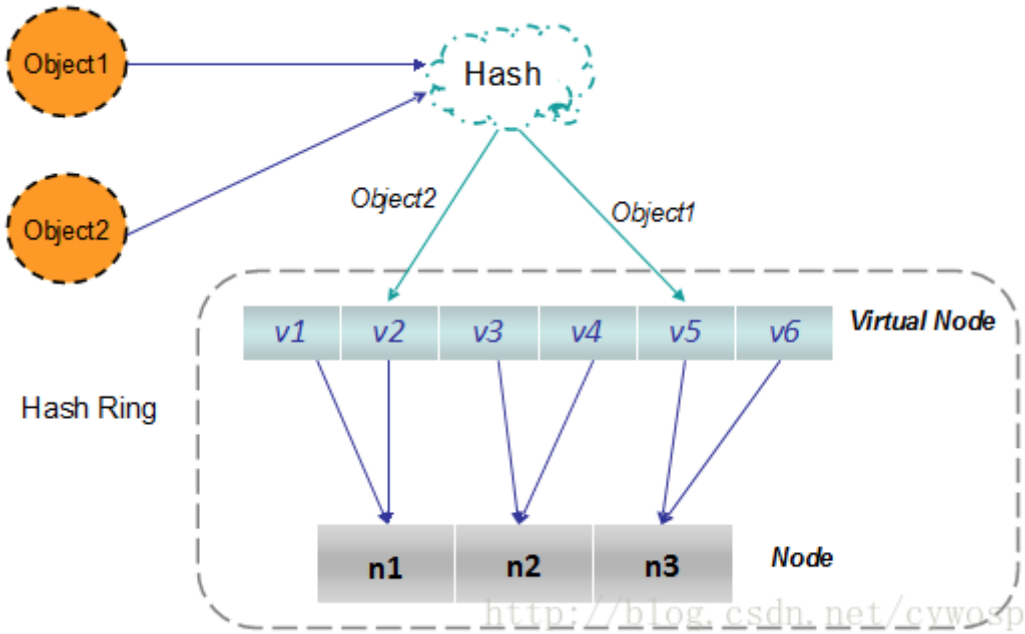
object2、object3、object4都存储到了NODE3中，这样就照成了非常不平衡的状态。在一致性哈希算法中，为了尽可能的满足平衡性，其引入了虚拟节点。

——“虚拟节点”（ virtual node ）是实际节点（机器）在 hash 空间的复制品（ replica ），一实际个节点（机器）对应了若干个“虚拟节点”，这个对应个数也成为“复制个数”，“虚拟节点”在 hash 空间中以hash值排列。

以上面只部署了NODE1和NODE3的情况（NODE2被删除的图）为例，之前的对象在机器上的分布很不均衡，现在我们以2个副本（复制个数）为例，这样整个hash环中就存在了4个虚拟节点，最后对象映射的关系图如下：



根据上图可知对象的映射关系：object1->NODE1-1，object2->NODE1-2，object3->NODE3-2，object4->NODE3-1。通过虚拟节点的引入，对象的分布就比较均衡了。那么在实际操作中，正真的对象查询是如何工作的呢？对象从hash到虚拟节点到实际节点的转换如下图：



“虚拟节点”的hash计算可以采用对应节点的IP地址加数字后缀的方式。例如假设NODE1的IP地址为192.168.1.100。引入“虚拟节点”前，计算 cache A 的 hash 值：
Hash(“192.168.1.100”);
引入“虚拟节点”后，计算“虚拟节”点NODE1-1和NODE1-2的hash值：
Hash(“192.168.1.100#1”); // NODE1-1
Hash(“192.168.1.100#2”); // NODE1-2

参考：
[1] http://blog.huanghao.me/?p=14

上一篇 TFS Erasure code实现方案
下一篇 每天进步一点点——ubuntu 13.10中安装google的gtest库

- 一致性Hash问题总结
- 一致性Hash算法的深入理解
- 使用虚拟节点改进的一致性哈希算法
- memcache的一致性hash算法使用
- 用sharding技术来扩展你的数据库（hash分布扩…

猜你在找

查看评论

70楼 qq_18681645 2017-07-07 11:37发表

感谢分享

69楼 w1047667241 2017-06-21 01:01发表

要点总结:
1、所谓的哈希算法，就是尽量做到随机无序，如果数据量足够的话，就可以看作是打乱顺序、平均分配位置的算法。
2、新增的redis物理节点，必须经过哈希算法，获取到n个虚拟节点（分身），每个虚拟节点遵循顺时针原则，管理散列在哈希环（哈希桶）上对应的对象（http请求）；
3、这样，即便是新增或者删除真实的节点，都不会影响太大，因为每个真实的redis节点都可以看作是平均分配在整个系统当中的，1个节点的失效最多引起n个虚拟节点的实效，考虑到虚拟节点有散列平均分配的性质，基本不会造成灾难性影响。

68楼 w1047667241 2017-06-21 01:00发表

要点总结:
1、所谓的哈希算法，就是尽量做到随机无序，如果数据量足够的话，就可以看作是打乱顺序、平均分配位置的算法。
2、新增的redis物理节点，必须经过哈希算法，获取到n个虚拟节点（分身），每个虚拟节点遵循顺时针原则，管理散列在哈希环（哈希桶）上对应的对象（http请求）；
3、这样，即便是新增或者删除真实的节点，都不会影响太大，因为每个真实的redis节点都可以看作是平均分配在整个系统当中的，1个节点的失效最多引起n个虚拟节点的实效，考虑到虚拟节点有散列平均分配的性质，基本不会造成灾难性影响。

67楼 wonderomg 2017-06-13 18:57发表

好nice的图

66楼 Small_Croco 2017-05-09 14:21发表

引入虚拟节点还是不能解决负载均衡
只是人为的认为把负载均衡变小了
但是实际有可能引入虚拟节点后，负载均衡反而变得更不均衡了

65楼 栗振娟 2017-04-05 17:04发表

刚了解，多谢分享。

64楼 郭二关 2017-03-31 09:12发表

LZ你好，我大致理解了你的文章，但是对于如何创建及分配虚拟节点方面还有点疑问，楼主可以推荐一些资料给我参考一下吗？谢谢

63楼 gyk668cn 2017-03-13 16:44发表

已经慢慢理解一致性哈希了。谢谢分享。

62楼 zhushuangyin 2017-02-09 18:45发表

博主写的很棒，我也总结了自己对一致性哈希算法的理解。总结如下，欢迎交流
http://www.zsythink.net/archives/1182

61楼 马金兴 2017-02-07 11:53发表

学习了，谢谢分享

60楼 RayexCui 2017-01-11 16:43发表

涉及到平衡性的问题，虚拟节点”（ virtual node ）是实际节点（机器）在 hash 空间的复制品（ replica ）
我的理解是不能彻底解除平衡性的问题，但是可以尽量负载均衡，利用虚拟节点遍布环形可以起到在一个节点移除的情况下，它顺时针下个节点压力过大的问题。虚拟节点有几个，相当于把平衡性问题的严重度变成了几分之一

59楼 RayexCui 2017-01-11 16:41发表

如果采用常用的hash(object)%N算法，那么在有机器添加或者删除后，很多原有的数据就无法找到了，这样严重的违反了单调性原则。接下来主要讲解一下一致性哈希算法是如何设计的?
其原理是通过使用与对象存储一样的Hash算法将机器也映射到环中 ----核心就是这一句，我之前还理解hash就是环形的取模，其实不对，hash本身就是环形的，但是之前的机器节点都是线性的，现在一致性hash是把机器节点也变成环形上

58楼 技术菌的blog 2016-10-14 11:41发表

讲得不错。但不得不说还是有点照抄嫌疑哦。我以为图是你原创的呢，原来是codeproject上的

57楼 koala958 2016-09-27 11:15发表

添加或者删除之后会照成大量的对象存储位置失效
添加或者删除之后会造成大量的对象存储位置失效

一实际个节点（机器）对应了若干个“虚拟节点”
一个实际节点（机器）对应了若干个“虚拟节点”

56楼 wu020708 2016-09-25 23:43发表

这个东西京东java岗面试居然问。。

55楼 大脸萌嘟嘟 2016-09-02 10:16发表

好好学学

Re: andy2lee 2016-09-27 15:24发表

回复KobeSilent：看来你面试的职位级别比较高啊，哈哈

Re: 大脸萌嘟嘟 2016-12-14 16:17发表

回复olijino：没呀，才工作一年而已

54楼 hengshan 2016-08-30 23:28发表

没搞懂别人抄你的，还是你抄别人的

Re: 曹学亮 2016-09-11 11:03发表

回复hengshan：习惯就好了

53楼 yysx 2016-07-22 16:29发表

楼主，请教个问题，在节点没有宕掉的情况下，node分布是不均匀的（是根据节点名称hash得到的），这怎么保证负载是均衡的

Re: _花烛 2016-09-06 14:20发表

回复u010066934：不知道你看到redis原理，槽的分布，节点和槽关系自己配就行了，意思是node1能存的范围可以自己配，不一定必须是所处位置

52楼 qq_26875119 2016-07-18 17:44发表

讲解的很好，值得学习。 楼主，大赞。

51楼 末日流云18 2016-07-18 14:07发表

你好，我想问一下有没有使用一致性哈希算法的具体例子，比如分库分表时的数据源路由等

Re: zlp1992 2016-09-17 10:35发表

回复qq_29583513：我也觉得这个可以用于数据库的分库，然后负责数据访问的负载均衡

50楼 jia635 2016-07-11 16:20发表

楼主讲解的非常好，终于懂了，谢谢！

49楼 qq_35174438 2016-05-31 11:25发表

我想问一下,就是我能先用余数算法,当要添加或删除的时候在执行哈希一致性吗?如果不行的话,我是不是要直接用哈希一致性就不用余数了

48楼 海涛zht666 2016-05-30 17:28发表

楼主写的很好，关键点上配上形象的图示，深入清楚的把复杂问题简单化了，5分钟就能看明白，感谢分享！

47楼 下刀 2016-05-30 16:01发表

这个环形Hash空间 是什么意思?

46楼 air_kev 2016-04-27 16:48发表

经过特定的hash算法计算之后 怎么能知道我计算完之后的这个对象存放到哪个node上面呢

45楼 xjyxbbmutou 2016-04-26 14:57发表

这个怎么解决碰撞问题，就是可能两台机器通过hash计算映射到环上的同一个node节点，怎么解决?

Re: yan005020671 2016-05-29 21:52发表

回复xjyxbbmutou：一个机器就是一个节点。。。

Re: 丹顶鹤是码农 2016-05-16 19:14发表

回复xjyxbbmutou：Hash本来就解决了碰撞的问题

44楼 WinMH 2016-04-16 17:02发表

浅显易懂

43楼 Brain先生 2016-04-16 10:53发表

懂了，讲的真好。

42楼 yulian529 2016-03-29 14:16发表

图文并茂，写得非常好！赞一个

41楼 tory_you 2016-03-25 14:53发表

非科班，半个小时都没有理解，不过还是谢谢楼主了

40楼 清晨Feelter 2016-03-21 13:14发表

写的真好，学习了，希望有时间多出。

39楼 清晨Feelter 2016-03-21 13:13发表

写的真好，学习了。希望有时间多出。

38楼 藏红 2016-03-16 09:43发表

真的是5分钟理解

37楼 榔头1号 2016-02-05 11:31发表

顶一下~

36楼 gaohui_2009_happy 2016-01-16 16:02发表

不错，赞一个，不过 有代码实例更好更能说明问题哈

35楼 熊猫小牛牛 2015-12-10 16:47发表

讲的非常好，谢谢

34楼 volshell 2015-12-07 23:34发表

单调性可以简单理解为：已哈希的数据，不会改变哈希值。

33楼 ljiabin 2015-11-26 10:30发表

分散性和负载不是很理解……
里面提到“终端”的概念，好像不同的终端看到的对象会在不同节点上，而下面哈希时没有提到不同终端的哈希，求指教?

32楼 wilfwild1 2015-11-23 17:02发表

学习

31楼 Owen292 2015-11-13 16:35发表

怎么理解这个环形Hash空间?

30楼 dev_test 2015-11-01 22:25发表

说的很不错，简单易懂!

29楼 jzhx107 2015-10-10 13:37发表

不错，谢谢，学习了!

28楼 hgsunyong 2015-10-07 16:55发表

图文并茂，谢谢分享!

27楼 rabb2 2015-09-10 08:23发表

讲的很清楚，多谢博主!

26楼 sinat_30806733 2015-08-24 10:50发表

您好，我是新浪微博@微博商业技术 账号的运营小编。我们每天会发布一篇好的技术文章。现在想转发您的这篇《每天进步一点点——五分钟理解一致性哈希算法》。不知道您同意吗?

25楼 一蓑烟雨_bupt 2015-07-27 13:57发表

讲的很详细，也很容易理解

24楼 klose 2015-07-08 11:16发表

看了这么多，就楼主这个最容易理解了！感谢！

23楼 gaohui_2009_happy 2015-06-12 09:28发表

感谢提出这么精辟的理解 ，但是有句话 “然后以顺时针的方向计算，将所有对象存储到离自己最近的机器中” ，最近节点，那么如果 object3 如果 比object1 离NODE1的节点进的话，应该怎么存储呢 ，是说转动圆环按照方向吧节点存到"就近"的节点，还是 "最近"的节点，这是不是有点模糊?

22楼 zhaorongsheng 2015-06-12 08:48发表

讲的非常棒，赞个

21楼 senliezheng 2015-06-02 22:26发表

楼主是参考这个网页的吧。http://www.codeproject.com/Articles/56138/Consistent-hashing

Re: wyj7260 2016-07-15 19:29发表

回复zhsenl：果然是翻译的文字

20楼 吃素了 2015-05-15 01:17发表

5分钟没看完，被打脸了

19楼 51412 2015-04-26 11:28发表

博客中的图是用visio画的吗?

Re: cywosp 2015-04-27 14:46发表

回复army27：不是，用office 2007 ppt

Re: 51412 2015-04-27 23:06发表

回复cywosp：使用ppt画的?

Re: cywosp 2015-04-28 09:41发表

回复army27：是的

Re: 51412 2015-04-28 15:41发表

回复cywosp： 使用ppt里的哪个功能画出来的?

18楼 在读中大 2015-04-11 22:44发表


楼主的牛逼

17楼 he037 2015-04-11 15:46发表


假如新增节点，它的虚拟节点是怎么分布的啊，重新分布吗？重新分布那数据迁移也很大，不重新分布，那怎么保证平衡性的啊

Re: cywosp 2015-04-14 11:01发表


回复he037：加入新节点时，会将该节点虚拟成多个虚节点，然后散列在环的不同位置中，最后将该环附近的数据迁移至该节点，已达到新的平衡。（一致性hash算法的主要目的是为了尽量减少数据迁移）

Re: air_kev 2016-04-28 16:45发表


回复cywosp：将该环附近的数据迁移至该节点，那么如何确定该环附近有哪些数据呢？怎么计算出来？而且新增一个node，那么我怎么知道他实际是在哪两个环的中间呢？

16楼 he037 2015-04-11 15:35发表


假如新增节点，虚拟节点会不会重新分布？如果重新分布那数据迁移量不是很大？

15楼 wellse 2015-03-30 19:24发表


图文结合很生动，理解了呵呵！

14楼 itfanr 2015-03-17 16:17发表

> 引用“cywosp”的评论：
>
> 回复houlianxue：office 2010

真心漂亮啊！

Re: Gavin-Li 2017-02-07 17:40发表


回复bjq1016：又见强哥。

13楼 y0908105023 2015-03-16 11:05发表


单调性写错了

12楼 crystal_dan7 2015-03-11 16:08发表


想请教楼主，在引入虚节点后，也是会出现不平衡的情况吧？

11楼 雨天要吃饭 2015-01-30 11:52发表


博主的图是用什么工具画的 很漂亮

Re: cywosp 2015-01-30 16:09发表


回复houlianxue：office 2010

10楼 jzhx107 2015-01-29 11:26发表

不错，谢谢楼主讲解!

9楼 展翅飞翔XD 2014-11-30 00:53发表

还是不知道怎么去实现，一般的hash算法都是32位(md5)甚至更多位，这样就不是2^32-1了啊，我想用php实现，博主能不能能个提示？？如果有例子就更好了，先谢谢了！

Re: ohmygirl 2014-12-03 17:23发表

回复XD675652950：计算的hash值（整数值）与 0x7FFFFFFF做 &运算不就完了。
或者%

8楼 展翅飞翔XD 2014-11-30 00:51发表

我想用php去实现，但是想不出来要怎么做，2^32-1就是8个f的16进制数，但是用hash函数，一般都是32位(md5)甚至更多位，这样就不行了啊，难道要自己去写这个hash算法？？博主能不能提示一下？？如果有相关例子就更好了，先谢谢了！

7楼 华为cloudos 2014-04-15 08:57发表

5分钟看不完：）但是楼主讲的很好，赞

Re: cywosp 2014-04-15 12:26发表

回复u012815727：熟悉相关概念之后 5分钟就能看完了

6楼 夏影2014 2014-04-14 10:56发表

感谢分享，现在正在学习中

5楼 守枫竹清 2014-04-12 13:19发表

谢谢了

4楼 soledadzz 2014-04-11 12:17发表

您的文章已被推荐到CSDN首页，感谢您的分享。

Re: cywosp 2014-04-15 12:25发表

回复soledadzz：谢谢

3楼 一只蚂农 2014-04-11 10:21发表

很不错

2楼 5iasp 2014-04-11 09:10发表

非常好，有java的例子吗？ 学习了。

Re: cywosp 2014-04-15 12:26发表

回复5iasp：很抱歉，这个没有

1楼 SpeedMe 2014-04-11 06:41发表

先讲理论，后结合例子讲。总结的非常不错！很容易看懂。