注:

笔记时间:2016-7-2

# (一)定义

A distributed system is one in which components located at networked computers communicate and coordinate their action only by passing messages

# (二)意义

1.升级单机处理能力的性价比越来越低.

2.单机处理能力存在瓶颈.

3.出于稳定和可用性的考虑.

# (三)服务器集群Session问题

1.问题来源:HTTP协议本身是无状态的.需要基于HTTP协议支持会话状态(Session State)的机制。这样的机制可以使得Web服务器从多次单独的HTTP请求中查看“会话”，从而获得访问信息。

2.解决方法：

1).Session Sticky:同样的Session请求都会发送到同一个服务器.

2).Session Replication:所有的服务器集群都保存一份Session.

3).Session数据集中存储:把Session数据集中存储到数据库中,然后服务器集群都从存储Session的数据库中获取Session数据.

4).Cookie Based:把Session数据存放在Cookie上,然后服务器解析Cookie生成对应的Session数据.

3.小结:一般来说采取Session Sticky和Session数据集中存储两种解决方案来解决Session问题,这个选择上主要是根据集群的大小.

# (四)数据库读写压力问题

1.问题来源:服务器集群的引入解决了系统的数据处理能力问题,但是没有解决数据存储的问题.随着系统的数据量与访问量的增长,数据库会存在比较大的压力,这时候需要提出解决方案来解决数据读写压力问题.

2.解决方法:

1).读写分离:一般来说大型系统到后期的一些业务都会出现读多写少的情况.当分离出一个读库的时候自然的也会带来两个问题:数据复制问题/应用对于数据源的选择问题.一般来说数据复制问题都可以使用数据库自带提供的数据复制功能.

2).善于利用缓存.数据缓存/页面缓存

3).引入分布式存储系统.分布式文件系统/分布式key-value系统/分布式数据库系统

4).专库专用,根据不同的业务对数据进行垂直拆分.难点:分布式事务

5).面向服务编程

# (五)中间件

1.定义In its most general sense,middleware is computer software that provides services to software applications beyond those available from the operating system . Middleware can be described as “software glue”. Thus middleware is not obviously part of an operating system, not adatabase management system, and neither is it part of one software application. Middleware makes it easier for software developers to perform communication and input/output,so they can focus on the specific purpose of their application.

2.三个领域的中间件:

1)远程过程调用和对象访问中间件:支撑应用服务化的基础。

2)消息中间件:解决应用质检的消息传递\解耦\异步问题.

3)数据库访问中间件:主要解决应用访问数据库的共性问题的组件.

# (六)大型服务器一致性基础-CAP

1.在分布式多服务器的情况下要服务器像单机系统一样做到强一致性,需要付出很多代价. CAP理论:

1)Consistency: All nodes see the same date at the same time.

2)Availability: A guarantee that every request receives a response about whether it was successful or failed.

3)Partition-Tolerance: The system continues to operate despite arbitrary message loss or failure of part of the system.

# (七)消息中间件(实践使用JMS+ActiveMQ)

定义与意义:Message Oriented Middleware is software infrastructure focused on sending and receiving messages between distributed systems.消息中间件带来了异步的特性,并且对分各系统进行了解耦。

# (八)分布式缓存

1.分布式缓存的特性

1) 高性能:当传统数据库面临大规模数据访问时,磁盘I/O 往往成为性能瓶颈,从而导致过高的响应延迟.分布式缓存将高速内存作为数据对象的存储介质,数据以key/value 形式存储,理想情况下可以获得DRAM 级的读写性能;

2) 动态扩展性:支持弹性扩展,通过动态增加或减少节点应对变化的数据访问负载,提供可预测的性能与扩展性;同时,最大限度地提高资源利用率;

3) 高可用性:可用性包含数据可用性与服务可用性两方面.基于冗余机制实现高可用性,无单点失效(single point of failure),支持故障的自动发现,透明地实施故障切换,不会因服务器故障而导致缓存服务中断或数据丢失.动态扩展时自动均衡数据分区,同时保障缓存服务持续可用;

4)  易用性:提供单一的数据与管理视图;API 接口简单,且与拓扑结构无关;动态扩展或失效恢复时无需人工配置;自动选取备份节点;多数缓存系统提供了图形化的管理控制台,便于统一维护;

5) 分布式代码执行(distributed code execution):将任务代码转移到各数据节点并行执行,客户端聚合返回结果,从而有效避免了缓存数据的移动与传输.最新的Java 数据网格规范JSR-347中加入了分布式代码执行与Map/reduce 的API 支持,各主流分布式缓存产品,如IBM WebSphere eXtreme Scale,VMware GemFire,GigaSpaces XAP 和Red Hat Infinispan 等也都支持这一新的编程模型.

2.分布式缓存的典型应用场景

1) 页面缓存.用来缓存Web 页面的内容片段,包括HTML、CSS 和图片等,多应用于社交网站等;

2) 应用对象缓存.缓存系统作为ORM 框架的二级缓存对外提供服务,目的是减轻数据库的负载压力,加速应用访问;

3) 状态缓存.缓存包括Session 会话状态及应用横向扩展时的状态数据等,这类数据一般是难以恢复的,对可用性要求较高,多应用于高可用集群;

4) 并行处理.通常涉及大量中间计算结果需要共享;

5) 事件处理.分布式缓存提供了针对事件流的连续查询(continuous query)处理技术,满足实时性需求;

6) 极限事务处理.分布式缓存为事务型应用提供高吞吐率、低延时的解决方案,支持高并发事务请求处理,多应用于铁路、金融服务和电信等领域.