

分布式存储与Web服务伸缩性

许式伟

2010-11-25

大纲

- 存储的特殊性和重要性
- 存储即数据结构
- 分布式存储的难度
- 分布式存储与Web服务伸缩性
- Web 建议框架

存储的特殊性和重要性

- Erlang Style Concurrency 理念之一
 - Fail fast（发生错误的模块应立即停止服务）
 - 关键点：一切都可以停止服务，但是存储服务是不能宕的，因为存储服务是系统状态的保持者，模块在重启之后需要从存储中还原出之前的系统状态，继续对用户提供服务。

存储的特殊性和重要性-续

- 可靠存储的代价
 - 存储服务不能宕，并非指提供存储服务的服务器不能宕。
 - 在 **Fail fast** 理念里，没有不宕机的硬件，所以存储的可靠性，是需要多台设备共同保证的。
 - 可靠的存储必须是分布式存储。

存储即数据结构

- 业务无状态模型

- 如果把数据中心的服务比做一个抽象的计算机程序，那么存储类似于数据结构，业务代码类似于算法。

- 这个比喻多多少少有点理想化，他假设业务代码没有持久需要维持的数据，所有状态在存储中保持。

存储即数据结构-续

- 业务无状态模型的优势
 - 系统中业务代码可通过运行多个实例来提升可靠性，而存储则通常由专业的存储团队提供，并且经过了长时间的稳定。
 - 事实上，传统的 LAMP（Linux Apache MySQL PHP）基本符合该模型。MySQL 负责存储，PHP 负责业务逻辑，分工明确。

存储即数据结构-续2

- 存储的多样化需求
 - LAMP 的不足之处在于，它忽略了存储的多样性需要。
 - 存储即数据结构。数据结构的多样性，意味着存储也是多样化的。
 - 在网站压力不足时，这种不足不构成太大的缺陷，所有类型的网站都可以用 LAMP 来搭建一个早期产品。但是，网站压力上来，存储的特殊需求就浮现出来。例如搜索、微博等等应用，都有适用于自身等存储结构。

分布式存储的难度

- 分布式存储的业务需求通常简单
 - 分布式哈希表
 - Put(key, value)
 - Get(key) -> {ok, value}
- 分布式存储的复杂度取决于质量需求
 - 要支撑的数据规模
 - 服务的可靠程度
 - 数据的一致性要求
 - ...
- 同样的业务需求，不同的质量要求，架构设计完全不同。

分布式存储的难度-续

- 质量需求的难度系数
 - 数据规模
 - 跨数据中心
 - 一个机房存不下？
 - 避免单点瓶颈
 - 避免单点IO瓶颈/性能瓶颈
 - 避免单点故障
 - 可靠性要求
 - 多副本
 - 数据自动迁移
 - 临时故障/永久故障
 - 数据一致性
 - 选举机制
- 这些需求是交织在一起的。
 - 难度系数是乘法的关系，而不是加法的关系。

分布式存储的难度-续2

- 研发代价（典型值）
 - 一个高可靠的分布式存储：3年
 - 一个类似DropBox的网盘服务：7个月
 - 一个典型的互联网站点：3 ~ 10个月

分布式存储与Web服务伸缩性

- LAMP 架构，瓶颈通常在 MySQL。
- 存储即系统的状态。在存储达到可伸缩后，整个Web服务通常就可以达到很好的伸缩能力。
 - 当然，某些特殊的Web服务仍然会有存储之外其他方面的瓶颈，例如计算能力上的瓶颈。

Web 建议框架

- 存储层（选择合适的存储服务）
 - MySQL
 - 云存储服务：Amazon S3、etc
 - 开源存储：Hadoop、etc
- 业务服务层（API层）
 - 按照业务逻辑，抽象业务API。
 - API 可考虑对外开放。
- Web 展现/用户交互层
 - 交互界面。
 - 将用户希望转换为API。

Q & A

联系方式:

xushiweizh@gmail.com

网站:

<http://xushiwei.com/>

<http://everbox.com/>