2010年的技术架构建议

Thursday, Dec 31st, 2009 by Tim | Tags: 2010, app engine, key value store

在 2009年最后一天，根据自己小小的视角提出一些技术建议，供同行参考。

编程语言

首先要能跳出语言之争及语言偏见，架构师需要在中立的角度选择最合适团队的语言，避免在技术决策中加入过多个人喜好。在系统语言层面，主要可关注以下几种

Erlang, 会继续在小圈子内流行，业界应用Erlang技术最大的障碍不是Erlang技术本身，而在于缺乏这方面专业人才。 余锋（淘宝褚霸）对Erlang技术的深度探索

Scala, 和Erlang不同，Scala有成熟JVM及丰富的周边library，在异构系统中集成也很容易，新项目使用Scala风险很小，所以Scala在新语言中应该有较大的提升优势。

Go, 由于刚开始推出，不适合正式项目使用，2010年会稳步上升，可适当关注。

其他语言基本保持现状。

架构

LAMP中的Linux, Apache, MySQL会受到云计算中的App Engine模式的冲击，因为App Engine在分布式处理，可扩展性，稳定性方面都有很大的优势。 在App Engine模式中，MySQL作用会降低，退化成一种存储服务。而且App Engine的存储服务含义会更广泛，传统架构中的MySQL, Memcached, 及key value store在App Engine框架下都会以底层的服务方式提供。存储不再是软件，而是一种可靠服务，因此也会带来分布式存储相关技术的繁荣。

Web 2.0的设计中，Cache会成为一个中心元素。传统的web应用cache只是一个可选的锦上添花层，即使去掉，PHP + MySQL这种模式也可正常运行。但随着未来应用social化及realtime的趋势，从facebook及twitter的设计来看，cache已经从可选层成为核心层。cache设计的好坏直接决定架构的成败。

由于web发展的趋势会使应用更realtime化，体现到技术层面是HTML5(websockets)及类似技术具有更高的价值。但由于阻碍生产力的IE存在，HTML5无法一步到位。建议关注能解决HTML5及旧ajax自适应的框架。

网络模型方面，由于多核的硬件环境，轻量级的进程模型值得采用。如传统的C++ boost的asio, 各公司自己实现的coroutine, Erlang的process, go的goroutines, Java/Scala的Netty/Mina框架等。但C++框架的代码优雅性可维护性欠佳，性能也没有突出的优势，可关注后面几种方案。

分布式方面，Dynamo及Chubby的思想会逐渐在国内的项目等到更广泛的应用，架构师会逐步丢弃双写，双机心跳等山寨式的容错设计思想，可靠的分布式设计思想会更普及。

存储

2009是key value/nosql产品百花齐放的年代。到2010年，它们之中优秀的会脱颖而出逐步主流化，主流化的产品周边的工具会更丰富，运维相关经验也会更成熟。目前阻碍很多key value产品推广很大一个障碍是运维的顾虑，而不是它们本身的性能。究竟会是Memcachedb/Tokyo Cabinet/Redis这样的小巧软件走向主流，还是Cassandra这样的巨无霸更受欢迎，我们拭目以待。