

# 单元化与分布式架构的切分问题

Monday, Dec 16th, 2013 by Tim

单元化是将一个系统的架构按某种数据特征维度进行垂直的划分，比如网站有100万用户，如按照用户维度进行划分，则可以分成10个单元，每个单元存储10万用户资料。单元化的一些收益如下

- 由于每个单元数据规模可控，相关维度内的所有资料可放在一个数据库中（如上例中的用户资料），不需要复杂的sharding分库分表逻辑，存储及缓存访问得到极大的简化。同时开发也变得简单，工程师不需要有丰富的“大规模大并发系统”开发经验。
- 同时由于计算离存储更近，也可以让数据离用户更近，比如用户数据存储在地域上靠近用户的位置，数据有了更好的局部性(locality)，因此也会获得更好的访问性能。部署上相关单元的前端、缓存、数据库、数据挖掘等节点可在同一个机柜，架构上让大数据的访问变得低廉，也在部分程度上让大数据更为快速及敏捷。
- 可以自然支持不同用户分片支持不同的功能特性，天然的A/B testing试验场。

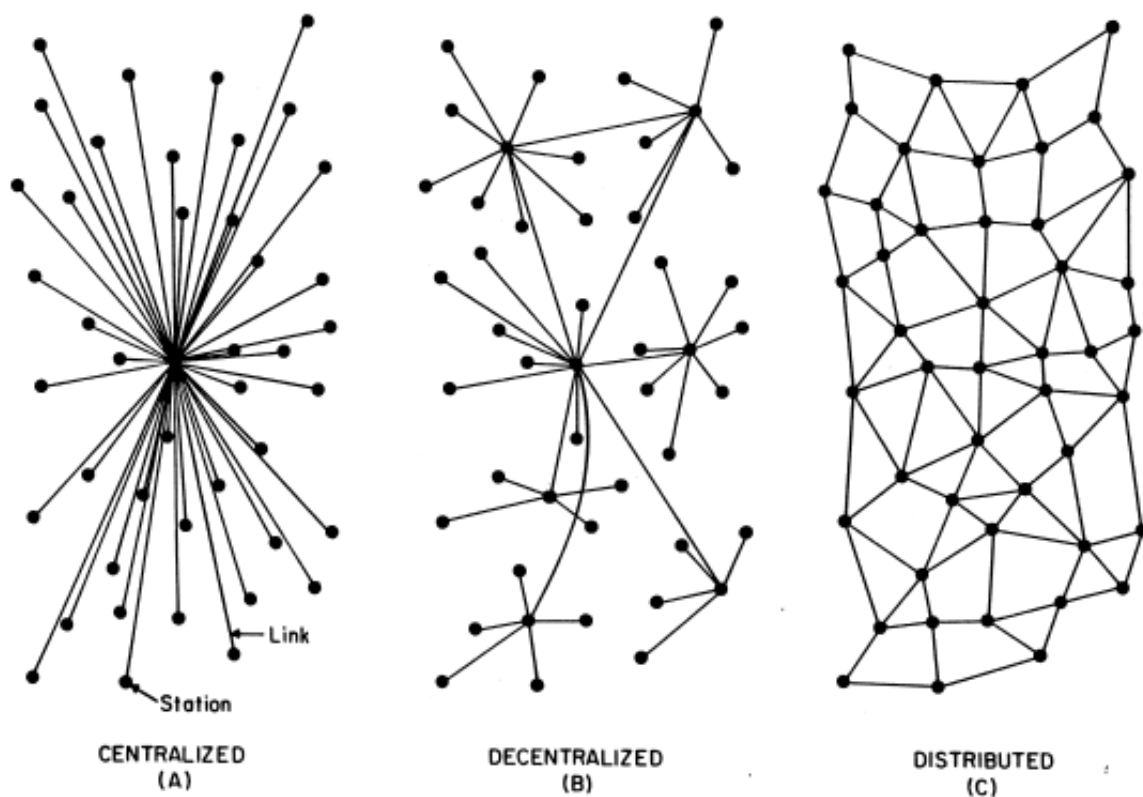


FIG. 1 – Centralized, Decentralized and Distributed Networks

分布式是将一个系统的数据分布到多个单元，以便使系统能够scale out，具有更好的可扩展性。当今大型网站基本上是分布式设计的。分布式系统除了机房内的，考虑到

系统扩展性、用户访问的便捷性、机房规模的物理限制、异地容灾（比如2013年4月及8月的微信机房故障）等需要，大型系统也会考虑地理分布在多个机房。

但是在社交网络中，由于数据的网状访问，单元化会碰到较难选择合适的单元化切分维度的问题，比如按用户或按内容进行单元化不能很好的适应数据访问局部性的问题，同时地理分布式也面临相似问题，由于社交网络中用户的页面需要访问的，所有地理分布的机房都同步需要全量数据，导致部署和维护成本较高。

假定系统中存在一个跨单元的数据访问总线，并且总线的访问满足

- 1. 局部性，单元内的访问，大部分的数据可以在单元内命中。
- 2. 封装性，单元内的应用程序使用统一的方法访问数据，不需要关注数据的具体位置。

那么这个数据总线应该如何设计？放在哪个层级比较合适？比如service layer, cache or storage(e.g. Google Spanner)?

