

了解集群

核心系统数据库组 余锋

<http://yufeng.info>

@淘宝褚霸

2012-10-10

集群初感

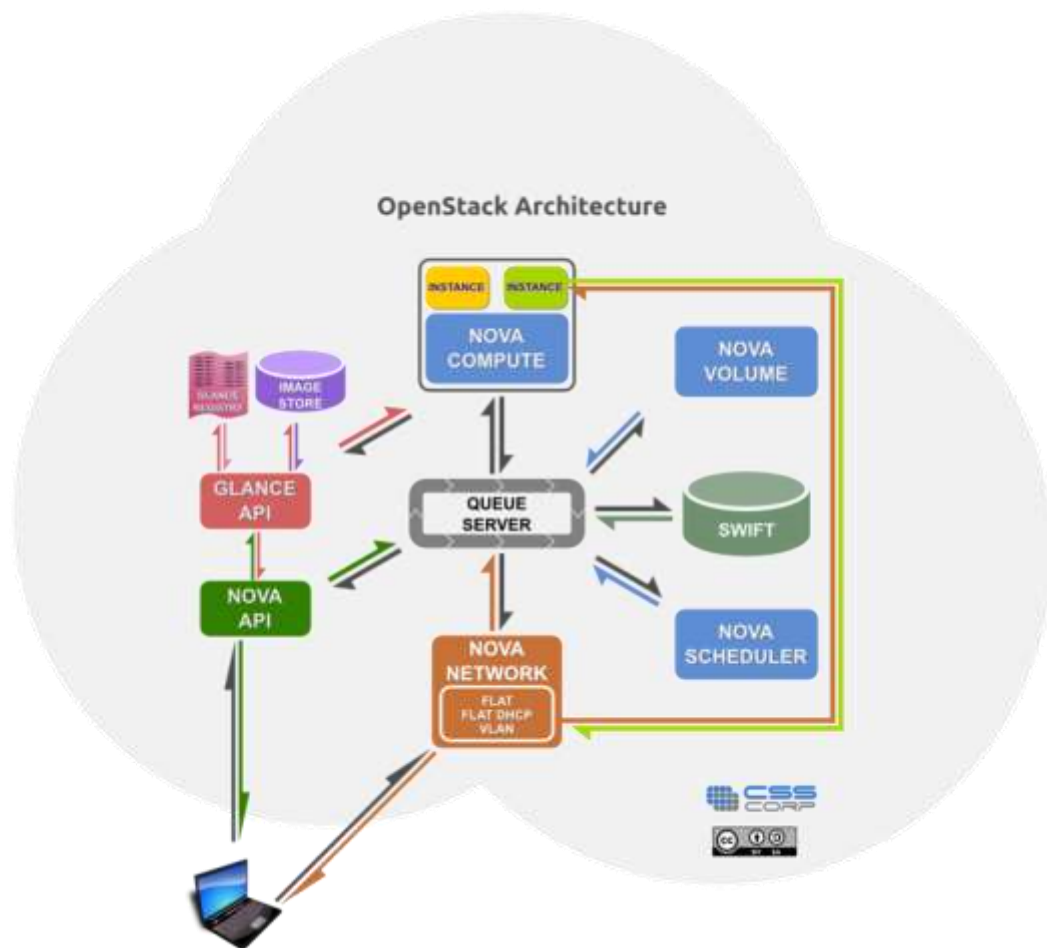
Cloud = "CPU + Storage + Queues + Management"
- *The Amazon Web Services says essentially*



- 同构与异构
 - 区别在于组成集群系统的计算机之间的体系结构是否相同
- 按功能和结构划分
 - 高可用性集群 High-availability (HA) clusters
 - 负载均衡集群 Load balancing clusters
 - 高性能计算集群 High-performance (HPC) clusters
 - 网格计算 Grid computing

集群典型例子

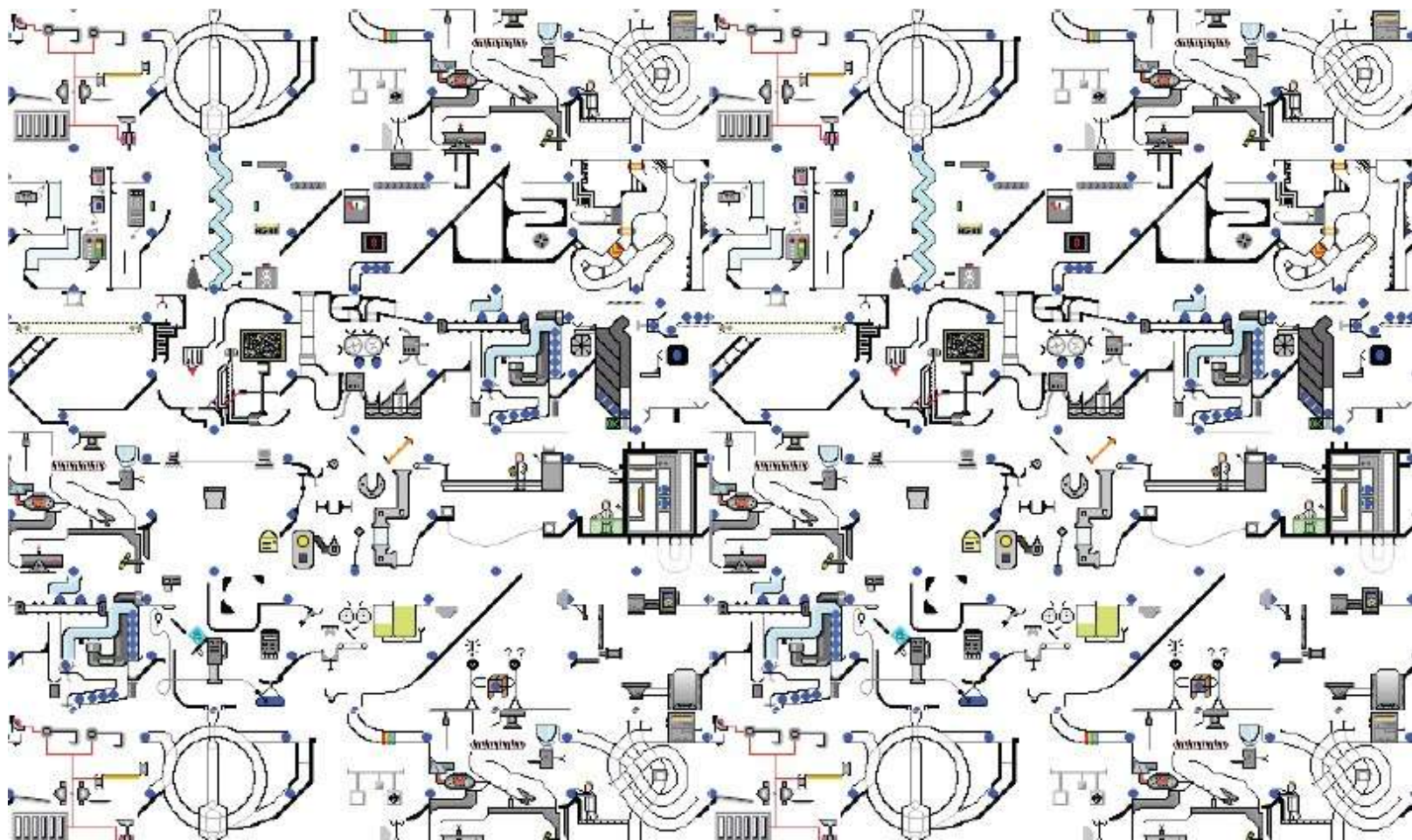
- WEB服务器
- 聊天服务器
- 存储服务器
- 云计算平台



- 通过层次消化复杂性,层越靠后业务越单纯
- 前端状态尽可能的推到后一层
- 后端功能单一、结构简单,性能和可靠性容易做到极致
- 性能不够,通常可引入cache层解决
- 三层是个好的选择

- 不中断服务(公路的例子)
- 容量规划(高速公路5车道例子)
- 预警机制 (依据是什么)
- 流量调度/排队机制(国庆高速公路不发卡例子)
- 降级服务 (高速绕行国道例子)

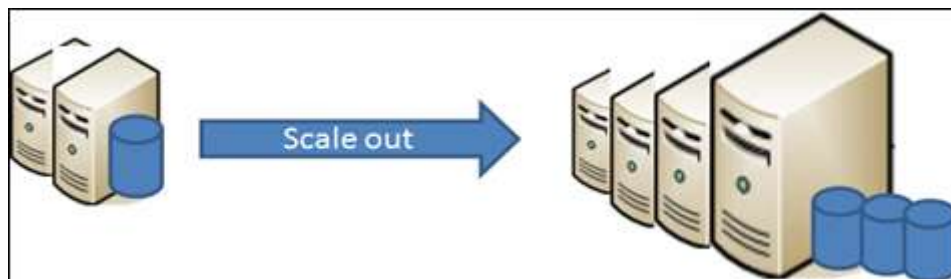
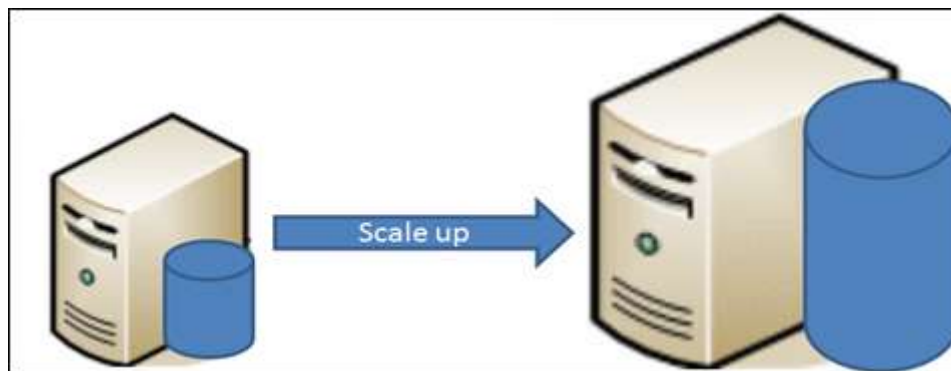
集群并发活动



- 再小的事情乘以13亿人口都不是小事情
- 避免群体事件
- 配置管理
- 资源池调度

- 流量切割
 - 4层和7层
- 角色分工明确
- 名称服务
 - 引入间接层
 - 其他服务如全局锁的基础
- 数据冗余
- 故障隔离/恢复

- scale out 还是 scale up
- 按照现实社会建模
- 分而治之
- 数据复制方式
 - 强同步
 - replica



- 逻辑单点
 - 单一视图简化系统模型
- 物理单点
 - 潜在风险点, 尽量消除
 - 候补队员

- 网络通信问题
 - 丢包延时/带宽问题
 - 可靠通信
 - 防火墙友好
- 消息队列(MQ)
- 网络风暴

- 系统实现足够简单清晰
- 开发效率/运行效率/维护效率方面的考虑
- 平台和工具方面的考虑
- 高并发带来的风险

- 采用成熟主流的技术
 - Massive adoption leads to massive investment.
 - Massive investment leads to better tools, better and faster VMs.
- 复用工业强度的部件,避免造轮子

- 高可用, 为失效设计
 - 硬件/软件都会失效
 - 节点自我保护(拒绝服务/服务降级)
 - 处处维稳 (非典期间人员监控的例子)
- watchdog/heartbeat 系统
- failover/takeover 机制



- 系统健康检查
- 双集群保证平滑切换(公路修理例子)
- 系统升级/热部署
- 节点添加/退出

- 为诊断而开发
- 日志/收集分析融入到集群
 - 信息尽可能的能自描述所做的事情
- 监控系统
 - snmp协议/web portal
 - 图形说话
- 主动诊断工具
 - 系统环境随着时间在变化
- 事后审计（超速例子）

谢谢大家！