

分布式服务治理

ESB

- 服务全汇聚到ESB，通过ESB暴露和调用
- 增强和中介
- 统计和监控

服务是业务语义，偏向于业务与系统的集成

分布式服务化作为SOA的另一种选择，以不同方式把ESB的一些功能重做了一遍

直连调用，侧边增强

配置中心、注册中心。有状态的部分，放到配置中心、注册中心、元数据中心。无状态的部分，放到应用侧（具体来说是框架和配置部分，尽量不影响业务代码）

配置中心（ConfigCenter）：管理系统需要的配置参数信息

解决问题

- 大规模集群下，管理配置信息，批量更新
- 开发、测试、运维分离（物理隔离）
- 运行期的一些开关控制

注册中心（RegistryCenter）：管理系统的服务注册、提供发现和协调能力

解决问题

让消费者能动态知道生产者集群的状态变化。这些信息很重要，后续的集群管控，分布式服务治理，都要靠这个全局状态

元数据中心（MetadataCenter）：管理各个节点使用的元数据信息

定义了所有业务服务的模型

常用中间件

- Zookeeper
- etcd
- Nacos
- Apilo

- 1、有存取数据的能力，特别是临时数据的能力。
- 2、有数据变化的实时通知机制，全量 或 增量。

服务注册一般流程

- 将自己注册到注册中心（比如zk实现）的临时节点
- 停止或者宕机时，临时节点消失。

服务发现一般流程

- 从注册中心代表服务的主节点拿到多个代表提供者的临时节点列表，并本地缓存
- 根据router和loadbalance算法从其中的某一个执行调用
- 如果可用的提供者集合发生变化时，注册中心通知消费者刷新本地缓存的列表

服务集群

服务路由

- 比如基于IP段的过滤
- 再比如服务都带上tag，用tag匹配这次调用范围

服务负载均衡

- Random（带权重）
- RoundRobin（轮询）
- LeastActive（快的多给）
- ConsistentHashLoadBalance（同样参数请求到一个提供者）

服务流控

为何需要流控

- 本质原因是，输入请求大于处理能力

三级别

- 限流（内部线程数，外部调用数或数据量）
- 服务降级（去掉不必要的业务逻辑，只保留核心逻辑）
- 过载保护（系统短时间不提供新的业务处理服务，积压处理完后再恢复输入请求）

都需要保存和读取数据/状态，变更通知。配置是全局非业务参数，注册中心是运行期临时状态，元数据是业务模型