

对J2EE项目实际应用的一些体会 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/462/2021_2022__E5_AF_B9J2EE_E9_A1_B9_c104_462556.htm

1、认真考虑是否真要使用J2EE 这个很重要，非常重要。J2EE涵盖的内容大而全，但很多不一定就是具体实际项目需要的。象EJB级的权限控制，如果你的表现层（大部分项目就是Web server）和应用服务器不存在信任问题，那么基本上就不用考虑。又比如伸缩性，如果同时在线最多不超过100个，就没什么用处。针对项目的实际情况选择效费比最合适的解决方案，而不要为了应用先进技术而应用先进技术。

2、选择合适的分布模型 提起分布，很多人可能都会有这样的设想：server A处理认证，server B处理订单，server C处理仓储；如果B的负载太大，那么再细分一下：录入、修改部分的EJB部署在server D，统计、分析部分的部署在server E，等等。其实没有必要，我的体会是：除非业务必须（如分支机构统一通过总部的app server来进行权限验证），否则最好将所有的应用全部放在一个app server中，能在一个进程空间内更好（使用home interface），然后进行平行的分布？D？D即集群中的所有app server功能上都是等价的。相比前一种垂直（或者网状）分布，平行分布的可靠性、容错能力、伸缩能力都要更好，同时减少了部署、管理负担。最重要的是，减少了因为业务逻辑层内部跨进程调用引起的开销，提高了整体性能。然而，如果a、一些业务逻辑必须相互独立部署、管理，b、负载较为集中地分布在若干个EJB中，那么，垂直分布还是必不可少的。

3、为Entity bean 选择合适的数据存储方案 首先尽量使用CMP管理数据存储，

尤其是简单的，大部分业务操作都是插入删除修改的实体，不然光insert 0update就够你忙的了，更不用说数据库移植问题。其次对于简单的一对一、一对多关系，如果你的app server没有实现EJB2.0规范，可以考虑使用O/R映射工具帮助开发，象Cocobase，EJB creator等等，可以提高不少效率。对于复杂的对象存储，没办法，老老实实写代码吧.....

4、慎重考虑EJBHome.findByXXX（），listXXX（）的实现 设想对一个百万记录级的表进行检索，结果集很可能是上万条、十万条，这本身就是一个耗费资源的过程；同时对于检索到的每一个记录还要做一次findByPrimarykey，这么多次跨进程调用，开销可想而知。为什么有的人觉得用J2EE实现的程序奇慢无比，缺乏仔细的设计就是主要原因之一。

5、使用抽象数据结构传递数据 例如，listOrder（）返回Collection而不是Vector，insertItems（）也是以Collection为参数而不是LinkedList.当然这个实际上与J2EE本身关系不是很大。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com