

2011年计算机二级考试JAVA知识点整理（70）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/647/2021_2022_2011_E5_B9_B4_E8_AE_A1_c97_647208.htm 2.6 应用服务器的集群策略

及Java EE 5.0 开源代表的经常是理想主义者，而商业公司代表的经常是现实主义者，两者之间有相互竞争的地方，但从长远来看，更多的是一种是相互补充、相互促进的过程..... 编者按：在中国Java技术界，袁红岗是一个不能忽视的名字。他的观点，及对中间件趋势的看法，是很多人感兴趣的。日前，在金蝶Apusic于广州花园酒店举办的“Java俱乐部”上，记者和这位极少露面的金蝶中间件首席科学家就集群、Java EE5.0等热门话题展开了直率的深入对话。果然，袁红岗出语惊人，带来了很多独特的视角和精彩的观点。不管是一般的观点，还是在平时打单过程中，我们似乎可以感觉到，集群功能一直是国外中间件厂商攻击国内中间件的弱点。而据我们所知，你们金蝶中间件在去年下半年推出了自己的集群功能，并且在宣传中提及，在国家质检总局全行业这个大单中和几个主要国外产品同等测试，测试结果甚至排在前面，这是否表示Apusic的集群功能已经能满足客户的需求？你对集群功能又怎么看，你认为中J2EE集群的本质是什么？首先我可以向你证实，在国家质检总局的核心电子业务系统“大通关”项目中，金蝶Apusic中间件与三家世界主要中间件厂商的产品，在同一平台和环境用国际测试工具进行了全方位的性能测试，经过三轮严苛的点对点、兼容性和性能测试，结果我们成功夺标。在测试结果中，Apusic在集群性能上并不逊色国外同类产品。集群是中间件厂商经常热捧的一个概

念，说只有采取集群策略你的应用系统的性能才能提高。不明就里的用户在付出了数倍的价钱去购买集群设备和软件以后，却往往得不到所应该得到的效果。Apusic作为一家负责任的公司，应当向大家来澄清所谓的“集群悖论”。所谓集群，只有在细粒度计算中其效果才会明显，也就是将计算过程以一定的并行算法进行细分，将计算分布到多个处理机运行，最后再将计算结果合并。有一个很有名计划叫做SETI@home，是一项利用全球互联网的闲置计算机共同搜索地外文明的科学实验计划，只需要下载一个小程序就可以对从射点望远镜得到数据进行分析。这就是一个典型的细粒度计算，所有的参与计划的计算机并行地计算浩如烟海的庞大数据库中的一小段数据，再将计算的结果汇总，从而发现可能的智能信号。而反过来我们看到在J2EE应用中大多数计算都是粗粒度的，再加上事务处理需要在分布式计算中进行协调，更降低了集群的整体处理能力。因此集群并不是解决性能问题的最佳途径，在单机低并发的情况下如果你认为性能不理想，那么请不要指望集群能给你带来性能的提升，相反你会发现性能反而还会有所下降。那么，集群仅是厂商宣传的噱头吗？在以下两种情况下集群是有用的：1. 高并发超负荷运行的主机，例如google这样的网站，它的访问量是相当大的，因此google会采取集群策略来分散客户的请求，以提高整体响应能力。我们接触的很多J2EE应用负荷量都不大，其实每秒访问量在500以下的应用都没有必要采取集群策略。2. 失效转移，其实我认为这才是集群真正有用的地方，使用一台低成本计算设备作为主设备的备份，在主设备发生故障时及时接替，以保证7x24小时不间断服务。综上所述，在准备采用

集群之前，一定要仔细分析具体的应用环境，以避免不必要的浪费。作为一种选择，ApuSic同样实现了集群技术，但我们并没有沿用大多数应用服务器厂商所采取的内存复制技术(in-memory replication)，我们知道在集群中需要在各结点之间同步一些状态信息，如果采用内存复制技术，将耗费大量的网络带宽，对性能也有很大影响。这是因为每当一个结点的状态发生变化时，都需要通过多播等方式向其他结点传递状态信息，随着集群内部结点的增多，内存复制将会非常频繁，从而造成广播风暴，严重阻塞带宽。ApuSic所采取的技术是客户端缓存，即直接将状态信息保存在客户端，当服务器失效时将状态转移到可用服务器。其实直到现在，还有人对中国能做出中间件不相信、对产品不信任。你在去年曾说“大家在同一个标准下开发，ApuSic和IBM、BEA的产品没什么本质区别”、对于这句话，你今天能否再解释一下？这个问题其实不需要证明，没有人认为神舟飞船和阿波罗飞船在本质上有什么区别，都是为载人航天而制造出来的工具，并不会因为一个是中国制造、另一个是美国制造，在用途上就存在什么区别。诚然，我们和国外产品还存在一些差距，但在J2EE标准框架之下，我们提供了完全可供用户使用的产品，用户的选择是对我们产品最大的肯定。中国软件起步较晚，基础较薄弱，但在中间件领域我们是及时跟进的，当时站在同一条起跑线上，现在仍然没有被淘汰出局，相反差距还在逐步缩小。我相信凭我们的技术实力，我们完全有资格和国外产品同台竞技。在我参加各种技术大会，包括去年北京Java10周年大会时，跟许多技术人员交流、聊天的时候，他们都反映ApuSic的启动速度非常快，很快就启动了，和同类

产品相比非常突出。看来使用者们对它快速启动的特点非常喜爱。据我了解，ApuSic的代码只是其它产品的几分之一，是因为这个原因吗？你设计时是怎么想的？很多人不理解，为什么ApuSic和其他产品比起来代码规模上要小很多，但使用起来并没有感觉到有什么功能缺失呢？这里要涉及到软件使用上的一个“二八原则”，即80%的使用者通常只会用到一个软件20%的功能。象微软的产品个个都是巨无霸，但对某个产品真正做到完全精通的可以说寥寥无几。以Word为例，平时我们只是用它来写写文档，很多高级功能其实根本用不上。在ApuSic应用服务器的开发上我们也是遵循同样的原则，我们将尽可能地将整个软件产品最重要的20%的功能做好、做完善，以保证大多数用户的需求，剩下的80%功能将根据需要逐步增加。譬如国外产品很早就有的集群功能我们最近才推出来，并不是我们没有能力实现集群功能，而是在我们看来，集群并不是解决性能问题的最好方案，只有在真正大并发请求下集群才会展现它的优势。因此，我们把集群功能归结为低优先级需求，只有在其他方面的性能和稳定性有了很大提高后再来考虑集群。另一个使ApuSic运行轻便的重要原因在于软件架构的设计。架构是一个软件的灵魂，好的架构将延长软件的生命力，轻松应付各种变化。ApuSic的架构在2001年时就已定型，以微内核和多路复用为其核心，历经产品多次重大升级而未影响核心体系，展现了顽强的生命力。相反，如果架构设计不合理，每次升级都要对架构进行调整，势必引入大量冗余代码，使整个产品臃肿不堪。第三个原因在于代码编写的简洁性上。莎士比亚有一句名言：“简洁是智慧的灵魂”，在科学界同样也推崇简洁性，麦克斯韦

方程组简洁深刻，被誉为是上帝谱写的诗歌，爱因斯坦的著名公式 $E=mc^2$ 更是将简洁性发挥到了极致。程序设计语言不仅仅是为计算机运行而设计的，它也是一种思想表达工具，甚至比自然语言更简洁、深刻、无歧义。我平时很少写文档，因为我认为代码本身就已经表达了作者的思想。当我看到简洁优美的代码时，我认为是在读一篇美丽的诗篇，并为作者深邃奔放的思想所折服。相反，当看到混乱、繁复而无章法的代码时，我相信作者的思想同样是混乱的。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