

【软考达人】

# 软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



微信扫一扫，立马获取



6W+免费题库



免费备考资料

PC版题库: [ruankaodaren.com](http://ruankaodaren.com)

## 论软件三层结构的设计

### 摘要：

我所在的单位是国内主要的商业银行之一，作为单位的主要技术骨干，2009年1月，我主持了远期结售汇系统的开发，该系统是我行综合业务系统的一个子系统，由于银行系统对安全性，可靠性，可用性和响应速度要求很高，我选择了三层C/S结构作为该系统的软件体系结构，在详细的设计三层结构的过程中，我采用了字符终端为表示层，CICS TRANSATION SERVER为中间层，DB2 UDB 7.1为数据库层，并采用了CICS SWITCH组，并行批量的办法来解决设计中遇到的问题，保证了远期结售汇系统按计划完成并顺利投产，我设计的软件三层结构得到了同事和领导的一致认同和称赞。但是，我也看到在三层结构设计中存在一些不足之处：比如中间层的负载均衡算法过于简单，容易造成系统负荷不均衡，并行批量设计不够严谨，容易造成资源冲突等。

### 正文：

我所在的单位是国内主要的商业银行之一。众所周知，银行的业务存在一个“二八定理”：即银行的百分之八十的利润是由百分之二十的客户所创造。为了更好地服务大客户，适应我国对外贸易的蓬勃发展态势，促进我国对外贸易的发展，2009年1月，我行开展了远期结售汇业务。所谓的远期结售汇就是企业在取得中国外汇管理局的批准后，根据对外贸易的合同等凭证与银行制定合约，银行根据制定合约当天的外汇汇率，通过远期汇率公式，计算出交割当天的外汇汇率，并在那天以该汇率进行成交的外汇买卖业务。远期结售汇系统是我行综合业务系统的一个子系统，它主要包括了联机部分、批量部分、清算部分和通兑部分，具有协议管理、合约管理、报价管理、外汇敞口管理、帐务管理、数据拆分管管理、报表管理、业务缩微和事后监督等功能。我作为单位的主要技术骨干之一，主持并参与了远期结售汇系统的项目计划、需求分析、设计、编码和测试阶段的工作。由于银行系统对安全性，可靠性，可用性和响应速度要求很高，我选择了三层C/S结构作为该系统的软件体系结构，下面，我将分层次详细介绍三层C/S软件体系结构的设计过程。：1、表示层为字符终端。我行以前一直使用IBM的VISUALGEN 2.0附带的图形用户终端来开发终端程序，但在使用的过程中，分行的业务人员反映响应速度比较慢，特别是业务量比较大的时候，速度更是难以忍受。为此，我行最近自行开发了一套字符终端CITE，它采用VISUAL BASIC作为开发语言，具有响应速度快，交互能力强，易学，编码快和功能强大的特点，在权衡了两者的优点和缺点之后，我决定选择字符终端CITE作为表示层。2、中间层为CICS TRANSATION SERVER(CTS)。首先，我行与IBM公司一直保持着良好的合作关系，而我行的大部分技术和设备都采用了IBM公司的产品，其中包括了大型机，由于CICS在IBM的大型机上得到了广泛的应用，并在我行取得了很大的成功，为了保证与原来系统的兼容和互用性，我采用了IBM的CTS作为中间层，连接表示层和数据库层，简化系统的设计，使开发人员可以专注于表示逻辑和业务逻辑的开发工作，缩短了开发周期，减少开发费用和维护费用，提高了开发的成功率；其次，对于中间层的业务逻辑，我采用了我行一直使用的VISUALAGE FOR JAVA作为开发平台，它具有简单易用的特点，特别适合开发业务逻辑，可以使开发人员快速而准确地开发出业务逻辑，确保了远期结售汇系统的顺利完成；最后，由于采用了CTS，确保了系统的开放性和互操作性，保证了与我行原来的联机系统和其他系统的兼容，保护了我行的原有投资。3、数据层为DB2 UDB7.1。由于DB2在大型事务处理系统中表现出色，我行一直使用DB2作为事务处理的数据库，并取得了很大的成功，在DB2数据库的使用方面积累了自己独到的经验

和大量的人才，为了延续技术的连续性和保护原有投资，我选择了 DB2 UDB7.1 作为数据层。但是，在设计的过程中我也遇到了一些困难，我主要采取了以下的办法来解决：

1、CICS SWITCH 组。众所周知，银行系统对于安全性，可靠性，可用性和响应速度要求很高，特别是我行最近进行了数据集中，全国只设两个数据中心，分别在 XX 和 YY 两个地方，这样对以上的要求就更高了，为了保障我行的安全生产，我采用了 CTS SWITCH 组技术，所谓的 CICS SWITCH 组，就是一组相同的 CTS，每个 CTS 上都有相同的业务逻辑，共同作为中间层，消除了单点故障，确保了系统的高度可用性。为了简化系统的设计和缩短通讯时间，我采用了简单的负载均衡算法，比如这次分配给第 N 个 CTS，下次则分配给第 N+1 个 CTS，当到了最后一个，就从第一个开始；为了更好地实现容错，我采用了当第 N 个 CTS 失效的时候，把它正在处理的业务转到第 N+1 个上面继续处理，这样大大增加了系统的可用性，可以为客户提供更好的服务；此外，我还采用了数据库连接池的技术，大大缩短了数据库处理速度，提高了系统运行速度。

2、并行批量。银行系统每天都要处理大量的数据，为了确保白天的业务能顺利进行，有一部分的业务处理，比如一部分内部户帐务处理，或者代理收费业务和总帐与分户帐核对等功能就要到晚上批量地去处理，但是，这部分数据在数据集中之后就显得更加庞大，我行以前采用串行提交批量作业的办法，远远不能适应数据中心亿万级的数据处理要求，在与其他技术骨干讨论之后，并经过充分的论证和试验，我决定采用了并行批量的技术，所谓的并行批量，就是在利用 IBM 的 OPC (Tivoli Operations, Planning and Control) 技术，把批量作业按时间和业务处理先后顺序由操作员统一提交的基础上，再利用 DB2 的 PARTITION 技术，把几个地区分到一个 PARTITION 里面分别处理，大大提高了银行系统的数据处理速度，确保了远期结售汇系统三层结构的先进性。在并行批量的设计过程中，我考虑到批量作业有可能因为网络错误或者资源冲突等原因而中断，这样在编写批量程序和作业的时候必须支持断点重提，以确保生产的顺利进行。由于我软件三层结构设计得当，并采取了有效的措施去解决设计中遇到的问题，远期结售汇系统最后按照计划完成并顺利投产，不但保证了系统的开发性开放性、可用性和互用性，取得了良好的社会效益和经济效益，而且我的软件三层结构设计得到了同事和领导的一致认同与称赞，为我行以后系统的开发打下了良好的基础。在总结经验的同时，我也看到了我在软件三层结构设计中的不足之处：首先，负载算法过于简单，容易造成系统的负荷不均衡：由于每个业务的处理时间不一样，有的可能差距很远，简单的顺序加一负载分配算法就容易造成负载不均衡，但是如果专门设置一个分配器，则增加了一次网络通讯，使得系统的速度变慢，这样对响应速度要求很高的银行系统来说也是不可行的，于是我决定采用基于统计的分配算法，即在收到请求的时候，根据预先设定的权值，按概率，直接分配给 CTS。其次，由于批量作业顺序设计得不够严谨等各种原因，容易造成资源冲突：在远期结售汇系统运行了一段时间之后，数据中心的维护人员发现了，系统有的时候会出现资源冲突现象，在经过仔细的分析之后，我发现，由于每天各个业务的业务量大小不一样，顺序的两个作业之间访问同一个表的时候便会产生资源冲突，另外，在 OPC 作业运行的过程中，操作员提交的其他作业与这个时间的 OPC 作业产生也有可能产生资源冲突。对于第一种情况，可以在不影响业务的情况下调整作业顺序或者对于查询作业运用 DB2 的共享锁的技术，而第二种情况则要制定规范，规定在某时间断内不允许提交某些作业来解决。为了更好地开展系统分析工作，我将在以后的工作实践中不断地学习，提高自身素质和能力，为我国的软件事业贡献自己的微薄力量。