|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Software Design Description**   |  | | --- | | 市场风险管理平台 | | | | |
|  | Version | Status | Date | Reviewed/Approved at / by |
|  | 0.1 |  | 20140313 | cuica |
|  |  |  |  |  |
|  |  |  |  |  |
| Template Version: SDD 4..0b | **INTERNAL / CONFIDENTIAL**  Storage location for temporary versions of this architecture concept document:  Final version: | | | |

**Read before using this template!**

|  |
| --- |
| **How to use this template**  **The purpose of a design doc is**   1. develop design concepts for the target component, 2. enable review of the key concepts, 3. support alignment between stakeholders, 4. record trade-offs and reasons for major design decisions as needed. 5. In addition we use this document as a ‘Guide to the code’ when the implementation is done.   **Consider these guidelines when using this template:**   * Document only what is needed by reviewers and required for development of high quality software * Less is better! Write short / concise and so that it is easy to understand. Focus on key design challenges, not completeness (large documents are a waste and no one will read them!) * Write for ease of understanding but don’t waste time on unnecessary polishing. * ‘Pair Design’ recommended (= develop key concepts together with a colleague) * Document design details & 'guide to the code' after the implementation. Only write what is not contained in the code/system already * A template is a template and not a form. Only fill what is needed and relevant. * This template is a base template. We expect large development units to derive their own versions that are specific to their environment / technology.   **Some guidelines on reviews**   * An Expert Review is mandatory only for chapter 2 * Differentiate between **expert review** (with *few* experts, deep review, issues & changes likely) and **light review / information rollout** (many people, no/few issues expected). * Expert Review Participants: only experts & *directly affected people* (provider and consumers)   + Design for team topic: review with dev team + stakeholders   + (Public) Interfaces / Cross-alignment: Providers and consumers   + Do a real meeting (not just mail / doc) to discuss the key design topics   + Try for no more than 7 people! * Light Review Participants: There can be many participants. Can be mail / doc based, meetings to discuss remaining issues are optional |

**Hidden Text contains important useful hints on how to fill in the template. Activate / deactivate these texts by pressing the following button in MS Word toolbar:** 

Additional guidelines:

* All graphical descriptions of architecture aspects must follow PACTERA standards for modeling:
  + **Technical Architecture Modeling (**[**TAM**](http://ency.wdf.sap.corp:1080/wiki/Technical_Architecture_Modeling)**)**
  + **SOA modeling using ARIS**.
* Do not delete non-applicable/optional items or sections of this template, but mark them with "Not relevant" and give a short reasoning why this is the case. This makes reading easier for readers which were not involved in the design discussions.
* The design document has to be available in **English / Chinese** language.
* Always store your documents in document management systems such as **Redmine** and avoid entering links to documents on shares or servers.
* Describe the **changes and revisions** of this document in the table on the title page.
  + Document versions follow the scheme x.y.z.
  + When you change a document, create a new version and keep the old document version. Make sure that “Track Changes” is turned on in the new version and accept all existing changes that were tracked in the old document version. Then insert a new line in the version table on the title page.
  + There are three types of changes, which affect the version number as follows
    - Status Changes:
      * Draft versions prior to review meeting => starting with 1.0
      * Version after action items of review meeting have been completed => starting with 2.0
      * Version after implementation (starting with 3.0)
    - Major Changes: Increment y-part of document version for complete revisions (e.g. 1.3 to 1.4)
    - Minor Changes: Increment z-part of document version for minor corrections (e.g. 1.3 to 1.3.1)
    - Changes after sign off of the document have to be aligned with the verifier(s) of the documents. If required a new review for the document has to be set up.
  + Consider update of existing central documents, which could be affected by this design (blue book, Architecture documentation, …).
* To update the table of content of this document, mark it and press <F9>
* Template Owner: Xu Ying; last changed: Feb. 2013
* Contact the template owner if you have any ideas to improve this template.

**Contents**

[1 General Information 4](#_Toc349599793)

[1.1 Stakeholders and Roles 4](#_Toc349599794)

[1.2 References 4](#_Toc349599795)

[1.3 IP Compliance and Patents 4](#_Toc349599796)

[2 Design 5](#_Toc349599797)

[2.1 Key Requirements and Design Goals 5](#_Toc349599798)

[2.2 Context 5](#_Toc349599799)

[2.3 Major Building Blocks 5](#_Toc349599800)

[2.4 Interfaces / Communication Handling 5](#_Toc349599801)

[2.5 Design Challenges resulting from Non-Functional Requirements 5](#_Toc349599802)

[2.6 User Interface 5](#_Toc349599803)

[2.7 Used Components and Frameworks 5](#_Toc349599804)

[2.8 Package/Development Component Concept 5](#_Toc349599805)

[2.9 Upgrade / Migration / Compatibility 5](#_Toc349599806)

[2.10 TCO Considerations 5](#_Toc349599807)

[2.11 Analytics / BI Content 6](#_Toc349599808)

[2.12 Compliance to Standards and Guidelines 6](#_Toc349599809)

[3 Design Details Documentation 7](#_Toc349599810)

[3.1 Database Design 7](#_Toc349599811)

[3.2 Testability and Test Environment 7](#_Toc349599812)

[3.3 Complex Algorithms and Applied Patterns 7](#_Toc349599813)

[3.4 Design Alternatives and Trade-Offs 7](#_Toc349599814)

[3.5 Guide to the Implementation 7](#_Toc349599815)

[4 Appendix 7](#_Toc349599816)

[4.1 Glossary 7](#_Toc349599817)

[4.2 Customizing 7](#_Toc349599818)

[4.3 Supportability Considerations 7](#_Toc349599819)

[4.4 Error Analysis 7](#_Toc349599820)

[4.5 Other 7](#_Toc349599821)

# General Information

## Stakeholders and Roles

List all people who were involved in the creation of this document and who should be involved in the final review and sign-off of this document.

|  |  |
| --- | --- |
| **Role** | **Name** |
| Author(s) | 崔长安 |
| Architect |  |
| Product Owner |  |
| Information Developer |  |
| Quality Responsible (QE/QPE) |  |
| Test Developers |  |
| < other relevant roles> |  |

## References

In this table give a list of all documents that are input / assigned to the design scope.

Typical references: ACDs and SRSs (mandatory), UI spec, Architecture guidelines, SOA content, work packages, naming conventions, …

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Title**  Title and version | **Date**  Enter document date | **Link**  document (preferably in cPro) | **Comments**  Enter the author/publishing organization responsible/reason for reference |
|  |  |  |  |
|  |  |  |  |

## IP Compliance and Patents

You must always store architecture and design documents 'IP safe', so that (if needed) PACTERA can prove that a certain idea / concept was invented / designed at a certain point in time.

# Design

Give a brief overview of the component / feature design, i.e. of the rest of the document. A reader / reviewer should be able to decide from this overview if they need to read the rest.

## Key Requirements and Design Goals

IMA系统从上游源系统获取数据，以及向下游目标系统发送数据，原则上，都应当通过银行的批量下传平台(DCDS)实现数据交互的过程。对于下游系统来说，IMA系统确保能够按照下传平台规定的数据规范提供接口数据并发送给数据下传平台进行数据发送。而对于上游系统的接口数据，只要源系统能够通过下传平台提供数据，IMA系统都会通过下传平台接收源数据。对于一些上游系统无法发送给下传平台的接口数据，目前阶段，可以直接发送到IMA系统的指定服务器。在未来合适的机会，上游系统可以将这类数据切换到批量下传平台，IMA系统支持这种接口方式的变化情况。

从整体架构上讲，批量下传平台应作为IMA系统的统一数据接口平台。

* 高效，由于源系统的数据量很大，数据处理规则复杂。因此高效处理、快速提供是系统的首要原则。
* 可扩展性，随着管理不断加强，目前数据分析类应用需求旺盛，源系统、目标系统的数据需求也在不断增加、调整，因此可扩展性是系统生命力旺盛的重要保障。
* 可管理性，作为一个重要的数据处理系统，其系统的可管理性也是十分重要的。
* 灵活性，审计业务要求的功能灵活，架构的设计中在审计系统功能设计上应体现较强的灵活性。

Briefly describe the key requirements and design goals that drive this design. Consider the requirements from the release backlog but also **quality attributes and non-functional requirements** (they should all be in the backlog as well). The reference to the backlog / requirements is given in References above.

## Context

* **系统中文全称**：市场风险管理平台系统
* **系统英文全称**：
* **系统英文简称**：MRIMA
* **项目的任务提出者**：风险管理总部，项目涉及拟改造系统的主管部门包括金融市场总部、风险管理总部等
* **系统使用用户**: MRIMA系统的主要用户为银行业金融市场业务条线相关部门和集团市场风险管理涉及单位、机构，具体包括：金融市场总部、风险管理总部、运营服务总部、财务管理部等海外机构。
* **系统目前支撑的业务状况：**MRIMA系统原定位为银行市场风险计量和数据处理平台（数据集市），在国内银行中较先实现了集团境内外分行以及各附属机构的全覆盖，系统包括限额管理、外汇敞口计量、内部模型法市场风险计量、压力测试、返回检验和监管资本计算等功能，为银行业实现新资本协议市场风险内部模型法的监管达标奠定了基础，提升了银行业一体化的市场风险管控能力。

State the business context and technical context of the component, sub-component, or function designed in this document. What is its purpose? Refer to requirements defined in the related Software Requirements Specification to make clear, how this design fulfils them.

Explain how the components and functions which are designed in this document fit into the overall architecture (as described in the corresponding ACD) and how they fulfill functional requirements and quality attribute scenarios (as described in Software Requirements Specification). To do so, provide a block/component diagram which shows the main building blocks of your component together with surrounding components, with which it is integrated.

Note: the design has to comply with the boundary conditions given by the corresponding ACD, the corresponding unit architecture guideline(s), and the PACTERA Architecture Guideline.

ERP_Overview_Example

Figure 1: Architecture Context of Component X (please remove example when finalizing this document)

## Major Building Blocks

MRIMA系统的数据和IT功能可以归纳为数据整合、处理、展现和管理四大类，如下图所示：

图表 2 IMA系统功能框架图

### 数据处理

#### 返回检验功能

IMA系统现有理论损益返回检验功能，新增基于资金系统的理论损益返回检验和实际损益返回检验功能。

#### 损益加工

根据实际损益返回检验功能需要，新建各类产品组合的返回检验用实际损益逻辑加工计算引擎；根据金融市场业务条线的利润计算逻辑新建各组织机构下的利润和损益加工计算引擎。

当前IMA系统对每月计算的外币利润进行按日插值的方式获取每日资本计量所需外汇敞口头寸，根据近期监管检查要求需对相关逻辑进行升级改造以满足按日计量的及时性。

使用资金系统产生的模拟损益计算用户自定义层级的VaR及压力测试，每日接受资金系统提供的数十亿条逐笔交易或头寸的模拟损益数据，根据VaR的参数设定进行排序和Percentile数值处理，得到固定口径和用户定制口径的VaR值。

#### 风险资本和绩效

IMA系统需基于资金系统的风险计量结果数据，开发银行内各级机构在内部模型法和标准法下的市场风险监管资本计算引擎，整合金融市场业务信用风险后，并根据RAROC等风险绩效考核指标结合1.4.3.2.2中计算的损益数据形成各业务方向的风险收益分析指标计算引擎。

#### 回归分析

IMA系统需能够接入满足银行需求的用于风险预测预警的所需要的历史数据，如经济宏观数据、金融市场数据、损益数据，需基于各类基础数据（市场数据、损益数据），定制开发中行特色的回归分析计算引擎，用于满足用户提出的回归分析类数据建模需求，具体的回归算法见3.7有关计算方法和计算公式。

#### 数据展现

##### 数据管理

###### IMA现有的数据模型优化

根据银行金融市场业务的特点，市场风险分析领域构建了包含交易头寸、交易对手、参考信息、市场数据四个主体域的数据模型，由于未来资金系统投产将使IMA系统接入更丰富的交易数据源，因此对原有数据模型的优化与扩充才能保证新的业务数据整合到IMA数据集市中，而成熟的数据模型才能支持数据展现层的各类风险指标的多维展示和汇报。

###### 数据存储管理

1. **安全便捷的的数据备份与恢复功能**

随着银行未来金融市场业务的高速增长，IMA系统将接受海量的前台交易数据，因此数据量的增长直接导致频繁的数据存储备份工作，所以根据科学的存储备份与恢复策略来构建数据储存管理的功能非常重要，最终保证系统维护人员能够高效便捷完成数据备份与恢复任务。

1. **高效的数据访问功能**

因为IMA系统未来将含有大量的交易数据查询与分析要求，应该保证在数据层面具备合理的存储设计来保证高效的数据访问。主要包括针对特定数据进行合理的存储分区，构建高效的索引以及数据表的拉链设计等。

###### 数据的安全访问控制

1. **报表信息的访问控制**

整个系统包含报表数量繁多，涉及不同机构、部门，所以应该具备设计统一的权限管理功能，使不同用户访问系统时能查看其自身权限范围内的报表数据。

1. **即时查询数据的访问控制**

单纯的报表数据往往不能完全满足用户的数据需求，因此即时查询数据功能在一定程度上满足了用户的这一要求，因此必须给特定用户开发特定数据表的访问权限，数据访问区域应该参考数据存储的源数据、管控数据、决策数据的类别划分，最终实现特定用户查询特定数据。

###### 管理网站控制功能

1. **管理网站应该具备专业的门户展现效果**

管理网站具备统一的门户登录页面，门户UI设计在参考国际同业专业市场风险管理门户的前提下，展现兼具银行自身特色的门户风格。门户同时应具备将来与集团风险整合管理门户整合的扩展性。

1. **管理网站实现用户单一视图展现功能**

管理网站将成为以后相关部门（金融市场总部、风险管理总部、财管部、运营服务总部等、海外分支机构）使用该系统的唯一的渠道，所以管理网站应该具备向特定用户推送特定信息，让特定用户定制特定信息、特定功能的能力，从而最终达到向特定用户展现特定信息视图的功能。

1. **管理网站实现实时的监控**

管理网站涉及用户、功能众多，所以针对用户的访问行为应该具备统一的记录功能，包括访问者身份、登录时间、登陆次数、执行操作等信息进行实时记录；另外因为IMA升级系统涉及的各业务部门的报表数量繁多，报表报送频度各不相同，因此针对数据批量的要求也各不相同，所以在对批量细分类型的前提下，能从管理网站监控并管理自动和手动批量，管理网站能够完全控制批量执行，并能监控批量执行的进度和状态。

#### 报表清单

经初步梳理，须在本次IMA系统升级改造项目中完成的报表需求共计300余张，在具体实施过程中需进行必要整合以减少固定报表，达到最佳的数据展现方式。为便于界定工作范围，以下仍以固定报表需求为准，具体报表需求以访谈完后签订的SOW为准，各部门报表数量与下列清单所列差异不超过10%。

Deliver a holistic, easy-to-understand description of the design topic. Take an “outside view” and adjust the zoom to show the high-level structure.

Describe how the design topic is structured into major building blocks, which responsibilities each building block has, and how they relate to each other. Typical building blocks on this level are packages or (sub) components, and important classes and interfaces.

It is good design practice to first focus on the most long-lived structures, which is typically the data / entity / business objects level (i.e. show the data model or the changes to the existing data model) and then describe the use cases / functions that use / traverse these objects.

Use TAM block diagrams and/or class diagrams to depict the building blocks and their relations. Additionally use prose to describe their responsibilities.

If helpful, visualize the desired behavior of the system and the interaction between the major building blocks as sequence diagrams or activity diagrams (“who is talking to whom in which order?”).

Describe typical interactions of client components with your component as sequence or activity diagrams and accompanying prose.

If helpful include business object models or entity-relationship diagrams of the most important database tables (table names, important fields only, no data types). A more detailed database design can be included in sections 3.x.2.

After reading this section your stakeholders must be able to understand:

* The scope and the problem described in the document,
* If the described solution appropriately solves the problem,
* Which parts of the component must be kept flexible and variable,
* The core principles of the component.,
* Potential future restrictions for extending the component.

Designers of client components should find all they need in this section.



Figure 2: Constituents and relatives of Order (remove example when finalizing this document)



Figure 3: Cancel Order (remove example when finalizing this document)

Use sequence diagrams for simple behavior, activity diagram for complex behavior:



Figure 4: Bar interaction (remove example when finalizing this document)

Restrict entity-relationship diagrams here to the most important database tables and their relationships, depict only the most important fields, omit types:



Figure 5: Order tables overview (remove example when finalizing this document)

## Interfaces / Communication Handling



整个架构由下及上说明如下：

1. 源系统

源系统为所有纳入IMA的数据来源系统，包括：

* MX系统
* RM系统
* Kondor+系统
* OPICS系统
* 中银国际、中银香港及其他舒服机构系统
* GL
* Bloomberg、Reuters、中债登及其他市场数据
* 财务报告系统
* 手工补录数据

业务系统的数据将通过接口数据的方式传递到IMA系统的ETL服务器上，并加载IMA系统的数据库中。或者是通过手工补录方式直接通过WEB界面录入到IMA系统的数据库中。

1. 数据加载

数据加载既利用ETL工具，通过ETL流程将数据按全量/增量、T+1/准实时的方式从源系统中抽取并加载到数据库中。

ETL将来自多个源系统的接口数据文本文件加载到数据库的ODS层（操作型数据存储）；在ODS层的基础上，将数据加工成根据统一数据模型所建立的基础层。

1. 数据库数据逻辑层次

根据数据整合程度,数据应用类型及建模方式的不同，数据库中存储的数据包括如下四个逻辑层次：

* ODS层

以贴源方式保存从源业务系统加载的数据，在本项目中，该层数据保留全量数据。

* 整合模型层

建立面向主题、集成、包含历史变化、符合3NF范式理论的基础数据模型，将ODS层数据通过数据标准映射加载到整合模型层，建立各业务对象的单一视图。

* 汇总加工层

汇总加工层存储面向各类业务应用需要的通用性高、共享性高的轻度汇总数据。建设汇总加工层的主要目的是为了提高通用数据访问效率，所以这一层的模型设计以高效为第一目标，会根据需要进行适当的冗余，综合考虑空间和时间因素选择最佳设计方案。

* 应用展现层

应用展现层是用于分析特定业务问题或者业务目标、服务特定业务人群（如某个部门）、按某种高效方式组织及访问的数据集合。存储按照分析主题划分、面向具体应用的结果型高度汇总数据或者某一个专题的所有粒度汇总数据。

关键指标、OLAP分析、固定报表、灵活查询集市存储相关类型应用展现所需的结果数据，是基于汇总加工层和基础数据层，根据各应用的业务逻辑计算出来的面向具体应用的高度汇总数据。专题分析集市存储面向一个分析专题所需的所有数据，可能包括各种粒度的数据。

1. 应用展现

根据展现形式的不同，IMA系统应用包括：

* 组合分析
* 固定报表
* 灵活查询
* 多维分析

1. ETL过程及作业调度

调度监控工具为ETL作业提供任务调度与监控功能。ETL作业完成数据的层层加工和转换逻辑。为最终的用户查询，报表服务和下游数据消费平台提供数据服务。

1. 统一门户

IMA系统的统一门户，以菜单项或者超链接的形式提供访问入口集成数据库的应用展现和数据管控体系工具，相关用户可以通过此入口单点登录并进行访问操作。

### 系统网络通信要求

系统数据交换集中在源数据系统和数据库服务器之间以及数据库服务器和分析服务器之间。根据同类项目实施经验，系统要求这些服务器部署在同一局域网中，带宽要求建议为不低于100Mbps，如果构建内部数据交换机制，建议使用千兆以太网交换机。而在客户端与服务器之间的带宽建议为2M。在客户可承受前提下，更高的带宽可得到更好的效果。以下是网络带宽的估算方法：

网络带宽的计算涉及以下两种情况：

* 服务器之间的传输带宽；

因为服务器部署在同一局域网中，局域网的带宽可以满足服务器之间访问的流量要求。

* 客户端与服务器的带宽；

调度，ETL加工脚本和报表生成是在局域网的环境下运行的，我们认为系统访问的流量要求在报表查询部分。下面我们以报表访问的带宽情况来计算客户端与服务器的带宽：

带宽计算公式：

C:\Documents and Settings\Crane\桌面\公式.gif

我们在假定打包损失在10%，交换机最佳效率70%情况下，考察不同大小报表，不同响应时间，在50个并发用户的情况下的流量情况如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 类  型 | 报表大小  (K） | 最大并发  用户（个） | 响应时间 | 服务端带宽要求 | 服务端考虑打包损失率（10%） | 服务端考虑交换机效率（70%） |
| 最小 | 35 | 50 | 10秒 | 1.400M/s | 1.556M/s | 2.222M/s |
| 最大 | 101 | 50 | 30秒 | 1.347M/s | 1.496M/s | 2.138M/s |

为了安全性，我们建议在服务器和客户端之间设立防火墙，但这不是必需的。如果分析层和表现层之间的确设立了防火墙，那么指定的TCPIP端口必须打开。客户端通信全部通过标准TCPIP执行，端口号在安装时设定，可在维护界面中调整。

Special consideration should be given to APIs for the designed functionality. A major customer-public API is worth designing in a separate design. **Do not make incompatible changes to existing APIs** without approval from the program architect.

Always include the potential clients/users of the API in the initial design discussions. It is a good practice to also define code examples, in the best case as test code, for any public API.

The interface description has to include the following information:

* Type of interface: Enterprise Service (B2B, A2A, A2X), API, BAPI, RFC, IDoc, Report-Report-Interface, …
* Interface structure or reference to separate document(s)
* Which generic parts (e.g. generated structures or name-value containers) does the interface contain?
* Show usage examples of the interfaces that show key arguments, sequence of calls etc.

## Design Challenges resulting from Non-Functional Requirements

### 非功能性需求(业务)

| **非功能需求项类型** | **非功能需求项说明** | **非功能需求内容描述** | |
| --- | --- | --- | --- |
| 联机查询 | 系统用户群说明 | 中银集团辖内各级机构中的市场风险管理决策者、市场风险管理参与人员、金融市场业务条线的各级用户、业务功能维护人员和IT系统维护人员 | |
| 每日查询时间分布图 | 由于集团内全球范围内的用户时区不同，该系统将根据各机构时区分时段跑批。 | |
| 新系统请估算三年之后的总业务量； | 全集团每日全量头寸估计数：  MX：约200,000笔，  每日市场数据估计数：  约20，000笔，  报表数量：  约200-300张  ETL和数据备份，需要将上述数据量乘以2  目前产品数为71个，三年后产品数估计为不超过100个 | |
| 业务高峰时间总查询量 | 2笔/秒 | |
| 业务高峰时间段每只查询量 | 10笔/秒 | |
| 每只业务查询平均系统响应时间 | 3秒 | |
| 容忍的最大系统响应时间 | 10秒 | |
| 用户平均操作时间 | 60秒 | |
| 每只业务交易成功率要求 | 99% | |
| 批量交易： | 日、月底、季末、年底、特殊时间批量时间窗口、数据量以及批量开始时间； | 以总行时区为例（全行分为4个时区）：  (T-1)19：00-(T)5：30，各源系统提供头寸、损益和风险计量明细数据（其数据需要与前台传输到后台的数据一致）；  5：30-6：30，数据平台通过FTP取得原始明细数据并对原始数据进行ETL  6：30-7：30，完成数据核对并生成数据质量报告；  7：30-9：00，生成各类风险报告；  9：00-11：00，按数据质量与出错报告对数据进行修改；  11：00-12：00，厂商数据更新  12：00-13：30，等待市场数据第一次更新后，生成各类风险报告；  12：00-13：30，等待市场数据第一次更新后，运行、下载并存储进行返回检验需要的风险数据；  13：30-15：00，返回检验完成后，如检验结果正常，向市场风险计算引擎上传头寸和必要市场数据。如返回检验结果异常，需要在计算引擎中计算相应维度细化表，完成初步结果分析后，再向市场风险计算引擎上传头寸和必要市场数据；  14：00-15：00，厂商数据再更新  15：00-16：00，等第二次市场数据更新后，从计算引擎运行、下载风险日报、压力测试和限额管理等需要的各项风险管理业务数据；  15：00-16：00，生成各类风险报告；  17：00以后，可以进行数据备份、升级、维护等其他工作。 |
| 对硬件、系统、网络层面的安全要求 | 保证有关头寸数据和风险数据必须满足中行各项保密协议规定 |
| 可用性 | 使用和投产范围：要开发系统的应用范围，包括是全辖使用还是在某一局部使用；使用时系统覆盖到哪一级别等。 | 系统全部建成投产后，应为集团全辖范围内各开展金融业务的机构所使用，第一阶段使用机构为境内法人和集团层面使用。 |
| 运行时间模式（例如7X24、批量时间要求）。 | 7X24小时，批量处理时可允许查询，但不能修改 |
| 普通用户和高级用户要高效地执行特定操作所需的培训时间。 | 待定 |
| 数据需要在系统中保留的时间，可以分数据类型描述（含非长期保留的数据以及历史数据的在线保留时间）。 | 在线保留3年历史数据，并离线保留全部历史数据 |
| 可靠性 | 数据恢复机制，故障恢复时间的数量级。 | 需要热机备份,故障冷恢复时间不超过半日 |
| 数据备份需要保存时间。 | 10年 |
| 性能 | 吞吐量的数量级（例如每秒处理的事务数）。 | 最复杂报表的查询响应时间为5秒内 |
| 说明现在的业务量及业务量的增长幅度等，提出系统应能处理的最大量。业务量用CIF客户量、账户量、交易量等具体参数描述。 | 市场风险管理内部模型IT系统用户账户数经初步估计需要200个，包括风险管理总部40个，金融市场总部60个，运营服务总部20个，财务管理部10个，海外分行40个，附属公司30个。随着业务规模的增加和监管要求等外部环境的变化，账户数目在未来可能扩展至500个以内。  其余待定 |
| 系统应支持的终端数。 | BS结构，用户数不限 |
| 应支持的并行操作的用户数。 | 50-100个用户同时在线 |
| 按功能说明应支持的特别交易的最大并发量。 | 50个用户同时在线使用 |
| 用户权限管理 | 用户角色的设置和管理 | 系统必须能按照用户的职能权限，对登录、系统特定功能应用、数据存取等进行约束，如：   * 只准许指定的用户和部门利用系统； * 只准许指定的用户和部门执行数据的输入或修正； * 只准许指定的用户利用指定的系统功能。 * 只准许指定的用户查询其授权范围内的报表。   系统内可定义拥有不同权限的用户角色，并根据用户的具体工作职责配置 (授权)适当的用户角色。 |
| 安全与  稽核轨迹 | 系统安全与稽核轨迹功能 | 系统对所有安全违规的情况 (包括真正的和尝试的) 必须能够提供警报。  系统必须提供全面的稽核轨迹，包括钻取到交易细目以了解交易的来源。系统必须具备控管用户改变各类参数和用户信息的功能。所有对参数和用户信息的修改，包括修改者的信息、日期、时间、修改内容等，都必须有所记录并保存。 |

图表 4 非功能性业务需求

### 非功能性需求(监控)

* 系统运行管理员（信息中心）能够登录系统查看批量处理执行的结果，该用户登录系统后只有系统监控操作界面，只能选择菜单上的功能选项，不能做任何其他操作，用户签退时直接退出系统。对于业务人员经常会手动操作的批量，必须有相应的监控界面显示批量运行状态，并可操作相应批量
* 应用管理员能够监控系统资源包括CPU、RAM等的使用情况。

Non-functional requirements have a major influence on design decisions and must be considered early on. Describe briefly the non-functional requirements / Quality Attribute Scenarios (e.g. from higher level designs / ACD) that have to be satisfied with this design. Add a reference to detailed descriptions / the backlog if this is needed / useful. Describe how these design challenges were solved and how the solution was verified (e.g. by a prototype, …).

Example: to achieve the goal of processing 100 million records/hour, parallelization is used. The prototype on the performance test system achieved already 85 million records/hour.

## User Interface

*\*\* UI and usability have a high priority at PACTERA. The UI is visible to all end-users. The UI description in the design shall enable all participants to judge if the UIs to be developed will be user friendly and easy to use\*\**

UI design is very important for usability and the end-user perception of your software. Reviewing the UI of a feature is just as important as architecture / design topics. For major development / changes in UI, consider having a professional UI designer do or at least review the UI parts.

You should follow the UCD (user centered design) process and their templates and process. Refer to the UCD documents here.

If you cannot follow the full UCD, provide at least UCD use cases and low-fidelity mock-ups for the user interactions (transactions, pop-ups, config screens, ... ) e.g. with Powerpoint or the like. Focus on planned interaction steps, layout, texts, ... that will be visible to end-users.

## Used Components and Frameworks

State, which components and frameworks are re-used and give the reason. If possible indicate the components and frameworks in the block/component diagram of chapter 2.1 or include a separate one.

Technology decisions have to be documented in ACD.

Consider, if any software components can be reused (e.g. from AP, “make or buy”). Refer to the corresponding chapter “Reuse of Existing Components” in the related SRS document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name**  Enter name of the reused component | **Description**  Provide a detailed description of the functionality used of the reused component | **Software Component**  Specify the software component of the reused/dependent component | **Implications**  Describe the impact for configuration, data, shipment, etc. |
|  |  |  |  |
|  |  |  |  |

## Package/Development Component Concept

### 主要软件产品特性介绍

数据处理/ETL在IMA系统中有其独有的特征：

* 需要处理的数据范围广，涵盖了整个银行业务头寸数据，同时还包括大量的市场数据以及损益数据；
* 需要处理的数据量大，IMA系统需要处理的数据是账户级数据，银行到目前为止账户处理的数量已经达到4亿，因此在设计数据处理/ETL机制和实现时，必须根据银行的实际数据状况来进行考虑；
* 需要处理的ETL逻辑关系复杂，IMA系统需要为多个单位提供报表服务，同时要为MX系统提供计算基础数据并获取返回的数据进行后续加工保存。再加上IMA系统是面向境内海外所有分行的业务数据，也面临着多时区多业务流程的复杂作业调度问题。为保证整体效率的提升，自动化作业和作业流的设计，实现和管理也是一个难度非常大的挑战。

根据文思海辉在多家银行客户的数据处理/ETL实践经验可知，数据处理/ETL的处理效率主要由处理环境的存储能力、IO能力和内存大小来决定，根据银行数据量大的特点，文思海辉建议硬件配置方案选择稳定性能高、IO能力强的小型机环境作为硬件处理环境，同时选择了对大数据量存储能力支持较好的Oracle数据库，稳定性较好的UNIX操作系统。

#### *Oracle数据库*

* 可伸缩性

真正应用集群（可选）：跨多个相互连接或“集群的”服务器运行任意未做更改的打包或定制的管理软件。

集成的集群件：利用一组通用、内置的集群服务创建和运行数据库集群。

自动工作负载管理：将服务连接请求发送给拥有最低负载的适当服务器；一旦发生故障，自动将幸存的服务器重新分配以用于服务。

Java和PL/SQL的本地编译：用Java和PL/SQL语言编写部署在数据库中的程序。

* 可用性

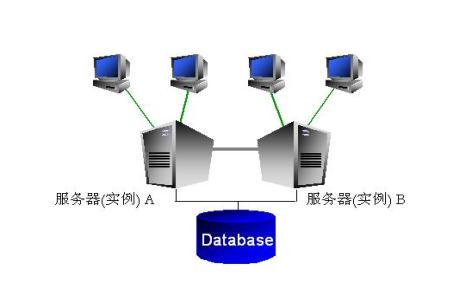
Oracle 数据库提供了丰富的特性和工具，以确保关键任务应用的数据访问支持和可用性。Oracle 数据库 11g 包括重要和创新的可用性特性，以确保您的数据和数据库随时随地可用。同时它提供的网格能力也可帮助您确定部署数据库环境所需的成本，同时能够根据业务需求进行扩展。

最高可用性体系结构(MAA)

Oracle 的最高可用性体系结构 (MAA) 是完全集成并经过验证的构建高可用性系统的蓝图。在 MAA 上构建其系统体系结构的企业发现，它们可以快速高效地设计并部署能够满足其系统可用性业务需求的应用程序。MAA 包含了详细的设计和配置建议。这些建议经过了广泛的审核和测试，以确保获得最佳的系统可用性和可靠性。Oracle 数据库的主要特性，包括真正应用集群（RAC）、Data Guard、Recovery Manager 和企业管理器解决了高可用性系统其他关键组件的配置和集成问题，包括服务器、存储、联网和应用服务器。所有这些特性都分别为 Oracle 客户的解决方案提供高可用性。这些特性的正确组合是完全集成的且遵循最高可用性体系结构的原则，为系统提供了最高可用性，因而是一个坚不可摧的解决方案。

真正的应用集群（Real Application Cluster）

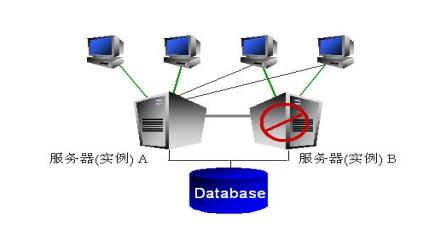
Real Application Cluster（RAC）为大多数关键业务要求的数据库环境提供了极高的性能和完善的纠错功能。Real Application Cluster允许集群系统或大型并行系统中的多个节点共享同一物理数据库。Real Application Cluster可以自动进行负载平衡、故障修复和规划停机时间，以支持高可用性应用程序。它还显著地提高了大型数据库和决策支持系统的性能。通过与并行查询选件结合，它还提供了节点间的并行性和节点内的并行性，以得到更高的性能。



图表 10 Oracle RAC架构示意图

在Oracle 11g的Real Application Cluster技术中，全面采用了Oracle独特的Cache Fusion(缓存熔合)技术，消除了多台服务器争用数据时产生的碰撞(Ping)现象，极大的提高了Cluster系统的可扩展性。使Cluster系统可以支持更多的节点，而且数据库应用完全不需要做任何复杂的修改或特殊设计就可以良好的运行在Cluster系统上，并且充分发挥多节点的处理性能，有效地提高主机的利用率，充分利用了计算资源。采用Oracle 11g RAC，正常时两个节点同时提供对数据库的访问，分担负载，处理能力将达到原来的1.8倍；

当集群服务器中某节点失效，透明的应用程序容错能够把用户自动转接到另一节点上继续运行，应用程序在用户没有察觉的情况下继续执行。这使周期性和非周期性发生故障的系统增大了连续可用性。进程的失效可以完全透明地转移到另一节点上去，通过适当地配置，可以指定所有查询都在客户端进行缓存，这样它们便可以在转移后的节点上重新设置。同时，还可以在没有失效时预先与容错节点建立一个连接，这样可以减少容错时在连接所花的时间。并且当失败节点修复后，失败节点可自动加入到集群，对应用提供服务。有非常高的可用性。



图表 11 Oracle RAC架构高可用性

优秀的分区( Partitioning) 技术

为了保证系统对于大数据量表的可用性和访问时的性能，可以使用Oracle的分区( Partitioning) 技术。Oracle Partitioning 把一个大表分割为独立管理的多个小分区，当用户访问该大表时，Oracle将根据SQL的where条件只访问对应的分区，其它的分区不作扫描（称为分区忽略），从而提高响应速度，并且分区忽略对于应用层面来说是透明的，不需要修改应用。Oracle Partitioning性能稳定，这已在大量的成功应用案例中得到证实。

方便的备份和恢复

Oracle提供的在线备份和在线恢复功能，允许在联机环境下，不中断正常的业务处理，对数据库和日志进行备份和恢复。Oracle 数据库服务器中的Recovery Manager(RMAN)模块提供了Oracle Server内部的高级备份和恢复功能，并可以和各种主流厂商的存储介质(如磁带库、光盘库等)配合工作

* 可扩展性

随着业务的不断发展，系统的用户数和处理交易量势必会不断上升。因此，一定要保证系统具有很强的扩充能力，并且能够伴随业务的增长逐步升级和扩充，以保持系统的连续性，避免出现重构数据库系统的事件。

Oracle数据库系统具有很好的扩展性，Oracle数据库可运行在各种SMP、Cluster、MPP硬件平台上，当硬件平台扩展时，可充分利用硬件平台的处理能力。建立在Oracle数据库上的数据库随之具有扩展能力。

Oracle数据库对于系统的数据容量基本没有限制，但数据大量扩展时，要考虑系统的I/O性能；通过利用高I/O性能的存储系统及合理的数据库结构的设计，满足系统的数据扩展性要求。

基于共享磁盘体系结构的 RAC 能够按需要增长和收缩。这对来自其它供应商的数据库是不可能的，因为它们基于非共享的体系结构，该结构不能提供这种灵活性。由于不进行共享，数据被人为地分区。当增加了更多的服务器时，所有的数据都需要重新分区，并将数据分配给新的服务器。类似地，当需要删除服务器时，在删除前需要重新对数据分区。

当资源不足时，RAC允许向数据库动态添加节点，有效地提高系统处理能力，同时也保护了投资。在添加节点时非常容易，不用手工对数据进行分区。对于所有的应用即时提供集群的可伸缩性－－不用修改应用程序。

* 可管理性

Real Application Clusters实现了真正意义上的一个单系统访问数据库，它提供了从任何节点到所有磁盘设备和远程高速缓存进行无缝数据访问的能力。此单系统映像延伸到所有数据库管理操作。安装、配置、备份、升级以及监控等操作只需进行一次，然后会自动发布到集群中所有节点上去。各种Oracle工具（如Oracle Universal Installer、Database Configuration Assistant以及Recovery Manager）将发现集群数据库中所有不同的节点并以它们为目标分配给想得到的任务。

通过采用Oracle Enterprise Manager网格控制，实现了对整个网格环境的系统在统一窗口下的管理。网格控制提供了一个简化的、集中化的管理框架，用于管理企业资源和分析网格的性能。利用网格控制，网格管理员能够在整个系统的软件生命周期内、从前至后、从网络上的任何位置通过 Web 浏览器来管理整个网格环境。

利用网格控制，管理员能够启动和运行任意数量的集成 Oracle 数据库特性（如 Data Pump、Resource Manager、Scheduler、Transportable Tablespaces 等等），管理员还能够通过相关的管理工具包（Diagnostics Pack、Tuning Pack、Change Management Pack和Configuration Management Pack），实现、监控、诊断、修改、调整、升级、补丁网格中的数据库。同时，管理员能够以不同的时间间隔调度多个系统上的任务，与其他的管理员共享任务，以及将相关的服务集中在一起，这为用户在系统管理方面带来了极大的方便。

因此，采用Oracle Enterprise Manager网格控制，可以将系统中的所有数据库进行统一的管理，大大简化管理的复杂度。

* 安全性

客户的数据库系统必须提供一套广泛而严密的安全机制，确保数据在存放、备份、传送和使用等环节上的安全性；要有大量的备份及恢复程序来减少数据丢失的可能性，具有灾难备份的功能；建立完善的操作人员授权机制把数据通过授权的方式保护起来，设立数据存取等级；对敏感数据的传输要进行加密，以防止数据被破坏、窃取；要有严密的措施来防范非法入侵。另外，对数据的操作在日志文件里也都要有记录以便于日后进行跟踪。

系统的安全性首先包括对系统数据访问的安全性。Oracle基于角色的安全控制通过灵活及易于管理的授权方式，保证了数据访问的安全性；另外还可以通过视图、数据库触发器等实现特殊的数据安全性；Oracle数据库的审计功能还可跟踪记录系统的各种操作，用于查找可疑的操作。

对于因系统、介质故障引起的数据损坏，通过Oracle提供的各种备份恢复方法来恢复数据，如数据库物理备份、逻辑备份、容灾备份、闪回查询等等。

为了实现数据库中数据的更高的安全防范， 从根本上防范由于数据库管理员过高的权限导致的数据风险， 可以采用Oracle Database Vault（数据保护）选件，防止利用数据库管理员来进行非法数据修改和访问的可能，把系统中的数据安全防范进一步提高。

另外数据库系统还可以利用网络、操作系统的安全控制来实现其综合的安全性要求。

* 快速响应时间

数据库系统要求数据要以高度规格化方式进行组织，通过智能合计和查询间并行性使数据库变得更小来能提高查询性能。要充分考虑到统计、查询、分析、决策等各项功能的需要，要为复杂的查询提供最快的反应时间。

Oracle数据库中的多种处理优化机制，如并行查询、并行插入更新和删除操作、并行装载、并行索引建立、分区技术等等，加上具有丰富经验的数据库整体应用结构和硬件结构的合理设计、数据逻辑结构和物理结构的合理设计，保证了系统的快速响应时间。

### 系统各环境的关系

IMA系统升级项目技术解决方案支持对开发环境、UAT测试环境、投产演练环境、海外推广投产演练环境、准生产环境和生产环境的版本输出。

开发人员在开发区实施客户化和报表制作，完成后提交到测试区供测试人员测试。通过测试的配置文档和报表程序将发布到生产区和准生产区。发布过程为授权操作，由发布过程管理员监控和记录。

ETL整理完成的数据集中在数据库服务器。测试区将抽取样本数据以满足测试要求。而生产区和准生产区将根据实际报表需要，定时从数据库服务器加载生产数据。

访问测试区同样需要账户口令。而且用户必须申请和生产区不同的账号以访问测试区。登录账号会有特别标示，说明此账户仅用来访问测试区。所有在测试区产生的报表将在标题栏注明，该报表为“测试报告”，该标示不能被用户手工修改。程序文档在测试区和生产区将采用不同的命名方式以区别。

准生产区的软硬件环境是和生产环境完全一致的环境。当应用系统测试验收通过后，由行内人员将系统移植到跟生产环境完全一致的准生产环境进行最终测试。成功后即可升级到生产区。

客户工作站通常连接到生产区以访问最终报表。在生产区和客户端之间要设立防火墙，生产区和测试区之间也需要有防火墙保护。

Refine the package concept of the ACD for the part of the application this design document focuses on:

* What packages/software components/development components does it consist of?
* What is the relation of these entities to the overall structure?
* How are these structured internally?

Use package diagrams to show the usage dependencies. If necessary/applicable, distinguish between design time, configuration time, and run time dependencies. Describe which package interface is involved in which usage dependency. If you use a general pattern (e.g. each package has an interface with identical name), describe only the pattern and omit a repetitive list of package interfaces.



Figure 7: Layer Concept (remove example when finalizing the document)



Figure 8: Structure of H (remove example when finalizing the document)

## Upgrade / Migration / Compatibility

### IMA项目安全设计

国标关于信息系统安全等级保护基本要求中对第三等级的信息安全做出了一系列的要求，以确保数据在存储、传输、处理过程中不被泄漏、破坏和免受未授权的修改。同时，要保护系统连续正常的运行，避免因对系统的未授权修改、破坏而导致系统不可用的情况发生。

在信息系统安全三级保护要求中，对提供的用户的访问控制，安全审计，主机安全，网络安全访问，参数配置，数据库权限等多方面提出了各种要求。

为满足信息安全保护要求，对于IMA系统采用以下的安全设计

1. 日志分析功能

充分利用各类日志功能，尽可能地从多个角度和其中既包括了对操作系统自身层面的日志分析，也包括了在数据库内总层面自动登记用户登录失败及越界操作失败的信息。在满足系统运行和用户的使用性能的前提下，尽可能多的进行各项日志分析的管理和保存，提供更详尽的信息安全记录和分析功能。

1. 用户访问控制

用户安全访问控制层面涉及每个需要用户认证访问的每个环节，包括了操作系统连接用户，数据库系统中每个数据库连接用户，使用各种前端应用的用户。

以上各类用户在项目需求分析及详细设计阶段还需要进一步的梳理和归纳，以确保每类及每个用户的权限在功能上不缺失，且在整理系统层面安全、合理，相互间不冲突。

1. 安全审计

为配合整个数据库平台的数据安全访问控制，在平台中将充分利用各类日志功能，其中既包括了对操作系统自身层面的日志分析，也包括了在数据库内总层面自动登记用户登录失败及越界操作失败的信息，另外在整个ETL数据过程中的关键环节（抽取、下载、入库）设置关键操作信息的记录功能，实现全流程的监控，确保数据在从源系统抽取至数据库内的安全与保密。通过以上几类详细的日志记录功能，可以有效地在事后通过详细日志记录来进行审计。

1. 主机安全

对于登录系统内各主机服务器的用户实行注册管理机制，针对用户/用户组设置是否允许登录，以及允许从哪些客户机（IP地址）登录；可以设置口令格式、口令过期规则、口令重复使用规则、允许口令错误次数等。

对于生产系统中所使用的操作系统为Unix/Linux，应依托服务器的安全体系结合数据库系统的安全策略，使得在操作系统层面的用户及目录的创建、授权、删除等管理，即用户、用户组可以访问的目录、文件以及目录的属主、读、写、执行权限，仅接受指定用户或用户组，在已授权范围内的操作。

1. 网络安全访问

系统部署在局内业务网服务区，同时会向外部机构和互联网用户提供相应的数据服务功能。网络安全应依托银行内部网络安全策略，对防火墙进行配置，使得数据库系统相关的所有服务器及相关查询访问服务器，在指定的网络环境中进行授权。互联互通方面应在有效的在防火墙层面就屏蔽未受权连接和访问；

相关服务器所需要必需起动的服务（即所起用的端口服务）也需要在防火墙中进行配置，使用与系统正常运行的相关服务可以正常提供服务，并满足指定范围内机器设备和用户的互联互通要求。

WEB应用开发方面应有针对性地防止WEB访问攻击，防止会话及Cookie劫持的跨域操作、SQL注射等危险事件的发生。

应用门户的权限管理则通过用户管理、机构管理、角色管理、资源授权在内的权限管理功能，实现数据一级的访问控制，支持分级授权。

1. 参数配置

在数据库系统中，所有参数的配置文件，包括有关用户名及密码项，不得以明文存储，而是采用一定的加密算法进行加密存储，以保证配置文件中关键参数的信息安全。

1. 数据库权限

通过安全措施的建立和设置，建立完善的权限管理，用户只能访问到允许访问的数据，从而保证数据不被非法使用。

数据库用户权限设置是由DBA来完成的。

对于数据的存取应以“需要知道”为基础，没有限制，可能会造成数据不当使用；但是如果限制太严格与繁琐，就会使信息难以利用，最终用户和IT用户在实际使用中感到过分繁琐，安全效果不好，因此需要DBA进行权衡。

要建立一定的安全机制措施来确保数据存取的过程是在一定的安全规则下执行，而且系统日志能够记录哪些用户，哪些时间，在访问哪些数据，以便对系统进行审计。在此同时，程序应尽可能有效率，不要对DBA造成管理负荷。因此对用户进行分级的权限管理十分必要。

If the present software design does not deal with a completely new functionality, it is usually based on an existing design that is enhanced, adjusted, or corrected. “Migration” should describe how any kind of content of the previous implementation can be moved into the present design without feature loss.

In any refinement case provide in this chapter a comparison how the current design relates to an existing implementation of a previous design, that is code, database content in either customer data or configuration case. Also reflect how the new solution may be deployed on an existing implementation. For this purpose the **compatibility and interoperability rules** of the architecture guideline must be recognized. Consider dependencies within the previous implementation that might imply further activities. Also rules and recommendations of the TICM product standard must be considered. From the total cost of ownership (TCO) perspective **XPRAs and manual upgrade/migration steps must be avoided**, but if inevitably required shall be described here as well.

## TCO Considerations

TCO (total cost of ownership) is a major consideration for customers. Some of these aspects (technology selection, systems / servers, etc) are already handled at the ACD level.

Describe how low TCO is achieved by your design for these and other relevant aspects on the design level: these aspects: Admin / config complexity, need for monitoring, dependency on specific versions of other components – and finally end-user usability.

## Analytics / BI Content

If you create new data or change existing data (entities, DB tables, …) consider if this data / change must be reflected in central BI reporting / content, Check if the BI reporting for this data is defined as a requirement in the backlog.

## Compliance to Standards and Guidelines

Consider PIL product standard requirements, industry standard requirements, as well as guidelines which are relevant for this design.

### Applied Architecture / Design Guidelines

Maintain version and storage location of the architecture guidelines, development guidelines, and programming guidelines relevant for this project. In case of deviation from the guideline, apply for an exception approval from the corresponding responsible organization (for example unit architecture board, ATG) and document it here (guideline, from which design deviates; reason for deviation; approved by n.n.)

|  |  |  |
| --- | --- | --- |
| **Name** | **Version/Date** | **Link** |
| Architecture Guideline of …. |  |  |
| BOPF Development Guide |  |  |

### Approved deviations

Please document here any deviations from product standards or architecture guidelines for this design. These can be deviations that you already know from the program / architecture level (that apply to you) or ones that have surfaced during development.

In case of new deviations arising during development, you must contact your program architect and get an approval.

Note if these deviations are temporary or long-term.

# Design Details Documentation

This section is not part of the review but part of the developer's / team's documentation of relevant details. **Write only what is needed to understand non-obvious details and what is not documented / contained in the code/system itself**.

## Database Design

### 数据流向



图表 12 数据流向示意图

### 处理步骤

1. 设计源数据的规则及ETL的抽取规则，从源系统获取的接口文件加载到ODS层，数据结构跟源数据相同；
2. 建立ODS层到业务主题数据的映射关系，并根据数据映射规则将ODS层数据加工到当前层；
3. 生成数据汇总规则和报表配置规则，将IMA各主题域数据，经过计算加工形成初步汇总数据保存在汇总层
4. 最终依据基础层数据和汇总层数据生成支持多种报表工具的报表数据，数据立方体和汇总数据
5. 生成备份规则，并将IMA各层数据按时间频率备份为历史数据集，历史数据立方体和历史汇总数据；

### 总控调度方案

为符合企业的商业目标，一个可靠、可扩展的自动任务调度系统至为重要。对于分析型系统平台而言，ETL是后台批量作业处理的重要技术手段，而调度不仅是后台处理中一个重要的概念，也是ETL中必不缺少的重要技术，它担任着后台众多批量处理程序管理者、执行者。

调度的本质就是各个批量任务的管理者、执行者。通过调度，众多的任务作业才能有序地展开工作，并协同完成庞大复杂的ETL处理。

使用调度软件实现作业调度。作业可以分布在多个服务器平台上，能够设定作业定义、依赖关系、顺序关系、工作组关系等，方便地对作业进行自动调度、运行和管理。调度监管平台可以图形方式动态监视和控制作业的运行，对作业执行中出现的错误/警告提供详细的信息。

* 有利于规范ETL开发流程：通过规范实施工艺，围绕用户需求信息，完善与用户数据需求相关的各种实现信息，统一管理与维护用户数据需求的全生命周期，完善实现过程与用户数据需求的关联关系，统一维护与管理ETL过程运行过程中的各种作业信息，减少需要人工干预的过程，形成作业指导规范。
* 有利于调度和监控产品化：调度和监控产品在项目实施中广泛使用，形成产品后避免重复开发，节约公司成本。
* 有利于提高实施ETL的效率，缩短实施ETL周期：通过统一的作业调度和监控，让开发人员把精力只放在具体的业务作业的开发上，减少ETL开发人员开发调度等功能，通过统一的调度和监控可统一作业的开发规范，可把作业的公共部分可通过产品化形式，最大化的节约开发工作，缩短开发周期
* 有利于降低实施ETL的技术门槛：系统提供统一的ETL平台，把ETL过程中繁多的技术对开发人员屏蔽起来，把众多工具的安装统一起来，减少安装步骤，清晰使用流程，让开发人员专一关注ETL作业的开发。

***Data*** ***Views*** describe the database layout including: tables, key fields, additional fields, and relations between tables. Consider the object-relational mapping between classes/objects and the database tables.

Relevance: needed when database tables are created or extended.

Use entity-relationship diagrams (UML class diagrams using attribute stereotypes to mark key fields).

Distinguish between system, customizing, and application tables, e.g. by using stereotypes for the delivery class.

**Archiving**: Do not forget archiving. Details can be found in the Information Lifecycle Management

## Testability and Test Environment

Briefly document the design elements that support isolated testability of this component / feature (e.g. mocking interfaces, test approach).

From a development perspective describe what should be tested and how (manual vs. automated, which unit test framework is used, which mock object framework or generator, or which capture-replay mechanism).

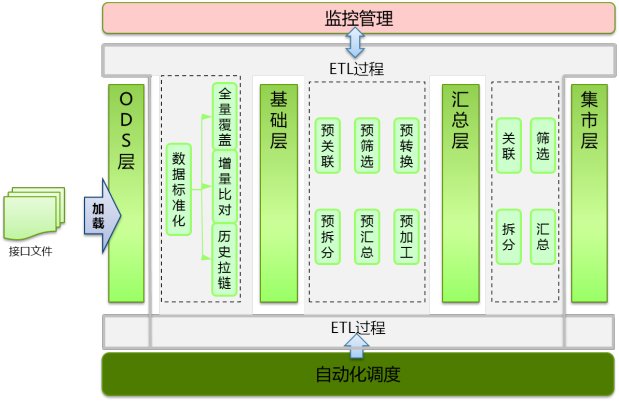
If there are special requirements for test execution, describe the test environment that is required to run the tests:

* Which system landscape is necessary?
* Which customizing settings (standard customizing, business content) are prerequisites?
* Which application data (master data and transaction data) is required? How is it generated?

Note: all implemented unit tests should be self contained. That means unit tests should not rely on any external context e.g. certain customizing or master data. Standard delivery customizing or business content can be used. If any external data or settings are needed they should be created in the test class itself or faked via mock objects.

## Complex Algorithms and Applied Patterns

银行的业务逻辑复杂，数据量巨大， ETL过程需具备较强的流程调度管理能力，能按照不同调度方式启动ETL系统调度流程，自动地操作和调度控制各个数据抽取、转换、加载等作业任务，生成满足计算引擎所需的数据，并提供数据质量监控及相关报告。



图表 15 ETL架构图

1. 数据加载(LOAD)

数据加载就是将数据文件直接加载到数据库，经过数据转换生成的数据文件的结构与数据集市数据表的结构完全一致，可以直接向数据集市加载，为了加快加载的速度，可以直接通过数据加载工具(如Oracle SQL Loader或者Oracle Enterprise Stage的Load方式)，以Bulk Load的方式进行加载。

数据加载工作可能分为以下3步进行：

* Pre-Load（非必须）

在真正进行数据加载之前根据实际要加载的表的情况，主要是针对detail及fact大表，基于性能考虑，可能需要完成以下准备工作：

* 删除数据集市中数据表的索引。
* 删除RI Key。
* Load

Load主要完成将数据文件的数据加载到本系统的表中，需要用到的加载方式有三种：

* Insert：只需要将数据文件所有数据完全Insert到目标表中。
* Upsert：需要对目标表同时做Update及Insert操作，根据primary key，对于已有的记录进行Update操作，对于不存在的记录做Insert的操作。
* Refresh：即将目标表的数据完全更新，一般的做法是先清空目标表的数据，然后再完全Insert要加载的记录
* Post-Load

Post-Load阶段主要完成在数据加载完成以后的相关的环境整理工作，主要包括如下一些工作：

* 重新生成索引：如果在Pre-Load阶段做了删除数据表索引动作，则在Post-Load阶段需要重建。
* 重新创建RI Key：如果在Pre-Load阶段做了删除RI Key动作，则在Post-Load阶段需要重建。
* 抽取数据文件：如果所加载的数据表后续被其他表加载时大量lookup，可以考虑抽取出文件。

1. 基础层ETL过程

基础层的ETL加载主要是将临时区中贴源结构的数据，按照一定的加载算法，加载到目标模型中。基础层的加载算法主要可以分为以下几类：

| **算法名称** | **描述** |
| --- | --- |
| 全删全加（Delete all/Insert） | 先全部删除目标表的数据，然后将新来的源数据全部加入目标表。此算法要求源表数据必须以全量提供。 |
| 直接更新（Upsert） | 使用源表数据更新目标表,如果新来的源数据已经在目标表中存在，则使用新来的源数据更新目标表中的相应数据，如果新来的源数据在目标表中不存在，则直接将其加入目标表中。针对此算法，源表选择全量或增量方式提供数据均可以。 |
| 历史拉链 | 在目标表中增加start\_date（开始时间）和end\_date（结束时间）字段，用来保存数据的历史变化。相对于历史数据的切片累加方式，此算法可以减少历史数据存储的空间要求并方便对历史数据的访问。针对此算法，源表选择全量或增量方式提供数据均可以。 |
| 历史拉链（带删除） | 在“标准“历史拉链算法的基础上，对源数据中已经删除的记录打标记，根据需要会修改end\_date来关闭该数据当前的时间链。此算法要求源表提供全量数据或能表示出删除数据的增量数据。 |
| 增量追加（Append） | 将源数据打上时间戳直接追加到目标表中。此算法要求源表数据必须以增量提供。 |

表格 8 基础层ETL算法

根据源表的类型、处理特征的不同，上述算法的适用情况如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **源表类型** | **更新方式** | **适用算法** | **说明** |
| 参数表 | 新增、修改和删除 | 全删全加 | 参数表的数据量一般不大，因此采用全删全加的加载算法。 |
| 普通信息表（账户主表、客户信息表） | 新增和修改 | 直接更新**/**历史拉链 |  |
| 普通信息表 | 新增、修改和删除 | 直接更新**/**历史拉链带删除 |  |
| 交易表 | 新增 | 增量追加 | 目标表一般为事件主题的表。通过加时间戳直接插入。 |

1. 汇总层和集市层ETL

按照目标表的数据结构，对一个或多个源数据的字段进行翻译、匹配、聚合等操作得到目标数据的字段

数据转换主要包括格式和字段合并与拆分、数据翻译、数据匹配、数据聚合以及其他复杂计算等。

* 字段合并与拆分，字段合并是指源数据的多个字段合并为目标数据的一个字段。字段拆分是指将源数据中一个表的一个字段拆分为目标数据的多个字段。
* 赋缺省值，对于数据集市中有的字段，在源系统中并没有相对应的源字段，这时根据模型的设计，可能需要缺省赋一个值。
* 数据翻译(Lookup)，将源系统中一些表示状态、类型等的代码直接翻译为其所表达的意思，或反之。数据翻译需要用到参考表（Reference Table），数据参考表一般是字典表或根据源数据与目标数据的定义手工产生，如果数据翻译时在参考表中找不到对应的对照，根据业务规则，需要将对应的记录Reject出来或赋缺省值。
* 数据合并(Merge)，按一定条件（一般是key值相等）对数据进行合并，找出描述同一对象的分布在不同数据表中的记录，并把这些记录联系起来。数据合并其实是数据翻译的一种特殊情况，主要用于数据量特别大的情况，数据合并在实现方式上一般先对要合并的两个表分别排序(Sort)，然后顺序对两个表的记录进行匹配合并，这样可以大大加快处理的速度。
* 数据聚合(Aggregate)，对数据按照不同分组进行汇总等统计计算，一般是用于汇总表的计算，主要的聚合种类有：
* 求和
* 求平均值
* 求记录数
* 求最小值
* 求最大值
* 取第一行
* 取最后一行
* 其他复杂计算，在数据集市中定义的某些字段需要按照业务规则进行复杂计算才能得到，主要针对一些在数据源中没有直接对应的字段，需要在ETL过程中通过相关字段计算才能得出的字段。

If this design involves complex logic / algorithms, sketch the algorithms you are using. Make clear which parts of your development are driven by used frameworks or APIs and which parts really need new concepts which must be described here.  
Try to structure your algorithms according to software design pattern. Explain why you think that this pattern matches the requirement, Consider the aspects of possible future extensions to your code, supportability, testability and simplicity.

## Design Alternatives and Trade-Offs

\If there were significant design alternatives and trade-offs, describe them briefly here. The purpose is to later remember why a certain design decision was made. When a decision between two alternatives was not trivial, it is usually worth documenting.

It is a good practice to document design alternatives as follows:

* Definition of the trade-off / topic
* What are the competing 'forces' (that create the design tension / trade-off)
* Alternatives
* Chose alternative with reason.

本项目主要实施风险及应对策略如下：

| **风险类型** | **风险描述** | **可能**  **性** | **影响** | **应对方案** |
| --- | --- | --- | --- | --- |
| 管理类 | 1、对MX系统了解不足，对前期技术方案及源数据分析工作造成一定影响 | 高 | 高 | 1、提前获取MX相关资料，并于MX项目组建立长效沟通机制，确保问题解决效率；  2、获取MX开发环境访问渠道，加强对系统深入理解；  3、关注MX系统开发进展，必要时参与相关方案讨论及评审工作。 |
| 2、项目周期较短，开发进度、质量压力较大 | 高 | 低 | 1、采取并行开发模式，缩短整体项目周期  2、提高高端人员比例，及模型设计、业务分析人员数量适应并行、快速开发模式  3、细化项目计划到2天粒度，及时监控项目进展  4、采取必要的赶工和加班策略。 |
| 技术类 | 3、多时区、多字符集等特点，对模型设计、总控调度等工作要求较高 | 高 | 中 | 1、采用标准模型设计方法论，构建架构稳定已于扩展的数据仓库级基础模型  2、借鉴ALM系统设计思路及经验设计总控调度方案  3、选择灵活性更好的调度工具 |
| 4、VaR值加工等复杂计算逻辑性能及灵活性要求均较高 | 高 | 中 | 1、构建汇总层，并充分平衡前后端数据处理量；  2、采用参数驱动模式简化脚本逻辑、实现数据展现的灵活性  3、建议采购尽量高性能的数据库主机设备。 |
| 5、对现有IMA系统功能及数据结构了解有限，对后期历史数据迁移有一定印象。 | 高 | 中 | 1、与现实施商建立有效的沟通合作机制，共同完成迁移工作。  2、前期实施过程，加强对现有系统的了解。 |

### 其它要求

* 各需求部门
  + 负责在业务访谈期间提出明确的业务需求，并对相关业务逻辑给出清晰的定义；
  + 评审项目组完成的需求规格说明书，并签字确认；
  + 提供数据核对依据，并配合完成系统集成测试工作。
* Murex项目组
  + 根据项目需要及时提供数据字典及可用测试数据，并在项目期内提供必要的讲解与技术支持；
  + 指派专人配合完成该系统相关数据接口方案制定及联调测试。

## Guide to the Implementation

This section should be written after the implementation is completed. Briefly describe ‘where to find what’ in the code, DDIC, or other tools / repositories that are relevant.

This information is very useful for yourself after a few months but also for handovers of code responsibility.

从项目实施工作内容来看，具体体现为总-分-总形式：



如上图，本项目实施将分为总体规划设计、迭代实施（总行投产、海外推广）及系统整合三个部分。

同时为确保在较短周期内完成项目，在迭代实施过程中，通过进一步细分工作内容，实现数据加载、数据整合、数据汇总加工、数据展现加工等ETL工作以及报表界面开发的并行推进。

### 实施阶段说明

根据中行相关规定，本项目建设通常分为准备、需求分析、设计、实施、系统测试、验收测试、试点投产及推广几个阶段，以下就本系统两期工作中的各阶段工作分别进行说明。

#### 总行投产期阶段各项工作说明

* 项目准备工作
* 工作重点：

1、确定项目总监、项目经理、技术经理、业务经理等核心成员，制定项目组织结构；

2、根据客户需求制定高阶（双周）实施计划，并根据计划申请项目资源；

3、评估项目实施风险并制定应对计划；

4、获取项目实施所需资料，开展相关学习；

5、提出项目场地、环境要求。

* 工作目标及要求：

1、组建项目核心团队；

2、完成项目计划及风险评估；

3、做好项目开展各项准备。

* 需求分析
* 工作重点：

1、分析MX系统数据接口并与现有数据接口进行差异性分析；

2、分析金融市场总部报表、运营服务部报表及财务管理部报表相关需求；

3、确定各报表所需数据范围及来源。

* 工作目标及要求：

1、充分理解客户需求及源系统数据情况；

2、为数据接入、数据模型、ETL调度等设计工作提供必要信息。

* 功能设计
* 工作重点：

1、总体架构设计及总体实施方案制定；

2、数据模型框架设计及评审，并结合数据源情况逐步完成数据模型设计工作；

3、一期建设所需数据接口内容、格式及传输方案制定；

4、接口数据加载与整合算法设计；

5、接口数据与模型映射规则设计；

6、数据库结构及存储、访问机制设计；

7、报表样式及开发模式设计；

8、报表数据汇总、加工逻辑设计及相关指标梳理；

9、门户及用户权限设计；

10、数据补录界面设计；

11、数据处理总控调度设计；

12、各模块测试计划及测试用例编写。

* 工作目标及要求：

1、结合需求分析成果，制定符合客户要求系统架构及实施方案；

2、各模块的概要、详细设计须满足客户提出的各项，并在必要时进行相应的验证性原型开发；

3、完成各模块测试计划及案例制定。

* 编码实施
* 工作重点：

1、开发、生产环境部署；

2、物理数据模型建设与部署；

3、数据处理总控调度模块开发；

4、数据处理脚本（加载、整合、汇总）开发；

5、报表功能开发；

6、门户功能开发；

7、数据补录功能开发。

* 工作目标及要求：

1、完成系统各模块开发；

2、总控调度模块须满足全球分行数据处理任务多时区、多币种、多语言等特点的要求；

3、数据加工脚本满足模板化、模块化设计要求，保障脚本一致性和质量。

* 集成测试
* 工作重点：

1、各模块单元测试；

2、数据接口联调测试；

3、系统集成测试；

4、处理性能测试。

* 工作目标及要求：

1、完成系统单元、集成测试以及与相关系统的联调测试；

2、优化系统性能。

* 用户验收测试
* 工作重点：

1、配合客户进行验收测试；

2、对测试问题进行修正和反馈；

3、进行系统运行相关准备。

* 工作目标及要求：

1、完成验收测试；

2、完成系统试运行准备。

* 投产演练
* 工作重点：

1、部署生产环境；

2、系统数据初始化及日常数据加载启动

3、系统使用培训；

4、监控试点地区运行情况；

5、对问题及故障进行分析和处理。

* 工作目标及要求：

1、完成试点投产。

# Appendix

## 资源估算逻辑

在本系统中最主要的资源是CPU、内存、I/O和存储，以下对这些资源配置的估算逻辑作说明。其中对于存储的容量估算参见3.4节的数据量估算。

估算的基础依据是按照能力评估场景下的环境进行推算。

能力评估时的环境配置如下表所示。

|  |  |
| --- | --- |
| 项目 | 配置 |
| CPU | 1\*4核 2.13GHz |
| 内存（GB） | 4 |
| 内置硬盘（GB） | 300 |

能力评估测试基本数据情况如下表所示。

|  |  |
| --- | --- |
| 项目 | 数量 |
| 测试交易数量 | 10,000 |
| 报表数量 | 6 |
| 复杂报表数量 | 1 |

## VaR计算引擎服务器生产环境配置估算

在VaR计算服务器上，部署的软件是定制开发的VaR计算引擎服务器。服务器上负载主要是由于VaR的原始数据加载和VaR的查询计算请求处理造成的，而数据处理的负载主要体现在VaR计算线程对CPU和IO带宽的占用上。缺省情况下，每个CPU分配1个VaR计算线程。数据的加载和VaR计算的查询请求都会分散在集群中的多个服务器上进行处理，以达到负载均衡的目的。

**用户查询等待时间分析**

VaR查询的等待过程主要分为下面几个阶段：

1. 一个VaR的查询请求会在任务控制分发模块被分解到不同的线程和集群节点上去执行
2. 查询任务会被划分到多个子任务并行进行查询，这个步骤是最耗费时间的，VaR的计算不会按时间汇总，其查询耗费时间是和查询需要查询的时间区间成正比的

最后查询的结果会在结果合并模块进行合并并返回

因此对于服务器的配置估算主要依据，应从以下几个方面出发：

1. 用户访问的并发量
2. VaR查询的复杂度，复杂度的依据主要体现在查询的时间区间和模拟损益的数据量
3. 占用资源的使用

**估算基础**

1. 数据量，目前估算的生产环境的每日数据量如下表所示：

|  |  |
| --- | --- |
| **项目** | **数量** |
| 1天的Murex给的模拟损益数据行数 | 约647741358 |
| 1天的Murex给的原始数据量 | 45G |
| 1年的原始数据量 | 16.04T |
| 1天的加工后的原始数据量 | 约6.2G |
| 1年的加工后的原始数据量 | 2.2T |
| 3年的加工后的原始数据量 | 6.6T |

1. 并发访问数量

|  |  |
| --- | --- |
| **项目** | **数量** |
| 报表的并发计算数目 | 15 |
| 报表中需要访问VaR的比率 | 30% |
| VaR即席查询的并发数 | 4 |
| 实际VaR计算的并发数 | 8 |
| 多并发下的IO等待系数 | 1.3 |

1. 查询复杂度

|  |  |  |
| --- | --- | --- |
|  | **一阶段** | **说明** |
| 简单查询 | 45% | 查询某一天的某维度组合的VaR |
| 一般查询 | 40% | 简单的查询条件 查询连续一周以内的某维度组合的VaR 所使用到的维度个数小于等于4  每天的模拟损益数字小于2000\*10000 |
| 复杂查询 | 15% | 查询条件较为复杂，用到5个维度以上  查询连续一周以上(平均200天)的某维度组合的VaR 所使用到的维度个数大于5  每天的模拟损益数字大于2000\*10000 |

1. 客户要求查询响应时间

用户对IMA系统的VaR查询平均响应时间不超过\_\_9\_\_\_秒。

**估算步骤**

1. 评估计算一天的VaR耗时

以能力评估时测试数据为基础，推算生产环境上计算1天的某维度组合的VaR的时间：

|  |  |
| --- | --- |
| 场景 | 计算时间 |
| 能力评估时加载一天的（2G）数据 | 90秒 |
| 推算加载生产环境1天（每天45G）数据 | 33分钟 |
| 能力评估时计算1天的某维度组合VaR（8000蒙卡模拟，平均） | 6秒 |
| 推算生产环境上计算1天的某维度组合VaR（2000蒙卡模拟，平均） | 1.5秒 |

说明：

* VaR的数据加载处理只是每天处理每天增量的数据，这个压力不大
* 最耗时的是查询一段时间内的某维度组合的VaR值

1. 按一般时间段15个并发用户，峰值时间20个并发用户进行估算

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **占比** | **一般时段** | **峰值时段** | **前端用户等待时间（需要补充）**  **单位（秒）** |
| 简单查询 | 45% | 3.6 | 5.4 | 1.5 |
| 一般查询 | 40% | 3.2 | 4.8 | 11 |
| 复杂查询 | 15% | 1.2 | 1.8 | 300 |
| 总人数(并发) |  | **8** | **12** |  |

1. 计算公式

(T1\*Q1+T2\*Q2+T3\*Q3)\*1.3/N≤TW

其中：

T1/Q1：简单查询前端用户等待时间/用户数

T2/Q2：一般查询前端用户等待时间/用户数

T3/Q3：复杂查询前端用户等待时间/用户数

N: CPU数量

TW：前端用户平均等待时间

对于一般时段，代入参数后，计算公式如下

（3.6\*1.5+3.2\*11+1.2\*300）\*1.3/ N = TW

对于高峰时段，带入参数后，计算公式为

（5.4\*1.5+4.8\*11+1.8\*300）\*1.3/ N = TW

综上，CPU数量推算如下表所示：

|  |  |  |
| --- | --- | --- |
| **CPU数量（核数）** | **前端用户平均等待时间(一般时段)单位（秒）** | **前端用户平均等待时间(高峰时段)单位（秒）** |
| 32 | 12.5 | 24.4 |
| 64 | 6.3 | 12.2 |
| 128 | 3.13 | 6.1 |

按照上面推算，如果满足用户对平均响应时间为9秒的要求，生产环境服务器的CPU个数总数应为128个。

对于生产环境将采用4台PC-Server集群的方式，每台服务器的配置如下

|  |  |
| --- | --- |
| 项目 | 配置 |
| CPU | 4C8核 |
| 主频 | 3.3GHz |
| 内存（GB） | 64G |
| 内置硬盘 | 8块高速硬盘 |

## 数据库服务器生产环境配置估算

能力评估环境下实现所有ETL加工时间为(包括数据加载和库内的ETL加工过程)：；平均运行时间为177秒，约2.95分钟

以能力评估场景的情况作为基础，考虑银行的IMA系统需要处理的交易，头寸和每日的接口数据量对CPU、内存及I/O的要求水平进行估算。数据库服务器的配置主要取决于加工报表的数量，计算引擎数量和加工流程的复杂度。

**步骤1**：生产环境的每日交易数估算、完成的报表的数量。如下表所示。

|  |  |
| --- | --- |
| 项目 | 数量 |
| 每日交易 | 100,000 |
| 交易年增长率 | 15% |
| 3年内单日交易量 | 152,000 |
| 实现的报表总数 | 300 |
| 复杂报表所占比例 | 10% |
| 复杂报表数量 | 30 |

说明：

复杂报表中所要实现的加工逻辑较为复杂，并且一张复杂报表所包含的计算量从能力评估的经验来看，是普通一张报表的5倍。因此，要实现的报表总数估算为：30\*5+270=420

**步骤2**：以能力评估时的CPU配置（1颗4核 2.13GHz）为基础，推算3年内数据库ETL加工的计算时间

|  |  |
| --- | --- |
| 场景 | 计算时间 |
| 能力评估10个报表(复杂报表折算) | 3:27 |
| 生产系统420个报表 | 27:14 |

说明：

每日加工数据量比例为：152000/10000=15.2

报表实现数量比例为：420/10=42 (复杂报表折算后比例)

在能力评估的机器配置下，完成项目要求的报表的所有ETL过程所需时间为:

177\*15.2\*42=132418秒

约为36小时47分。

**步骤3**： 考虑到生产环境需要实现T+1的数据加工逻辑，并且需要留出足够充分的时间窗口给数据备份操作，同时考虑到用户要求ETL整体加工时间理论极限不超过4个小时。

另外，考虑到IBM AIX power750的CPU主频平均值为3.3GHz，与能力评估场景所使用的设备的比率约为1.54，在进行性能估算的时候，按照保守1.2倍的处理速度进行估算

因此，CPU数量推算如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 数据库服务器 | CPU处理速度比率 | CPU数量 | 计算时间(秒) | 计算时间 |
| 能力评估计算 | 1 |  | 4 | 112996 | 31:46:58 |
| IBM p750 | 1 | 1.2 | 8 | 47082 | 13:4:42 |
| 1 | 1.2 | 16 | 23541 | 6:32:21 |
| 1 | 1.2 | 24 | 18832 | 5:13:52 |
| 1 | 1.2 | 32 | 11770 | 3:16:10 |

根据上表推算结果，如果要满足ETL加工时间窗口的要求，需要单台小型机的CPU数量为32C。考虑到数据库服务器还需要作为调度软件，报表软件的知识库存储以及运行环境，建议生产环境的服务器总的配置不低于32C。生产环境上如果采用两台小型机搭建Oracle RAC架构进行集群运算服务的话，每台数据库服务器的CPU数量不低于16C

**结论：**

综合考虑服务器的计算压力、报表的多用户访问机制，以及计算引擎等因素的增长和冗余的情况，可以得出每台数据库服务器的配置如下表所示：

|  |  |
| --- | --- |
| 项目 | 配置 |
| CPU | 16C |
| 主频 | 3.3GHz |
| 内存（GB） | 64G |

## WEB/报表服务器生产环境配置估算

在WEB/报表服务器上，部署的软件包括WAS和Cognos。服务器上负载主要是由于数据处理造成的，而数据处理的负载体现在报表进程对CPU的占用上。缺省情况下，每个CPU分配1个报表进程。当集群服务器中某台服务器的所有CPU均被报表进程占用的情况下，新的报表处理请求将被转给报表进程未被占满的服务器进行处理，达到负载均衡的目的。

另外，还有一部分用户是通过WEB服务器进行VaR的灵活查询。在进行查询的过程中，也是要占用CPU资源进行查询请求处理。

**用户查询等待时间分析**

用户通过WEB/报表服务器进行报表/VaR查询的等待过程主要分为三个阶段：

1. Cognos解析用户的输入，生成报表查询SQL语句；对于VaR查询则是生成一个包含查询信息的xml片段。
2. 提交SQL到数据库端，等待数据库查询结果返回；对于VaR是提交查询请求到VaR计算引擎服务器，等待查询结果返回。通常影响前端客户等待时间的主要是这一阶段的工作。对于简单查询来说，能够迅速返回结果。而对于复杂查询或者灵活查询，会由于查询SQL的复杂度高，关联的表多，或者汇总过滤的条件较为复杂等原因，处理时间较长；而VaR查询基本上都是条件不固定的，在处理模式上可等同于数据库的灵活查询。
3. 将查询结果进行必要的后续处理后，将最终报表展现给用户。这一阶段的工作一般情况下耗时不多。但当用户查询的结果集合较大，前端报表需要进行分页处理等页面展现的加工处理过程。或者有用户进行EXCEL数据下载或手工数据导入操作时候。这种情况下都会因为该用户的操作导致对CPU占用时间过长而降低系统并发度。

因此对于服务器的配置估算主要依据，应从以下几个方面出发：

* 用户访问的并发量
* 报表/VaR查询的复杂度。
* 占用资源的使用

**估算基础**

1. 并发用户的数量估算

对于IMA系统来说，用户群的总人数约为1000人，参照其他系统的实际经验，同时在线的人数为15%，即150人在线。在线的并发查询率为10%，所以IMA系统需要支持的并发人数为15，考虑到系统的服务对象，在工作日的上午时段，尤其是月初的时候，可能出现系统的访问高峰，峰值系数按照1.3计算，则在高峰时间，需要支持的并发访问率为20人

1. 交易数量估算

|  |  |
| --- | --- |
| **项目** | **数量** |
| 每日交易 | 100,000 |
| 交易年增长率 | 15% |
| 3年内单日交易量 | 152,000 |
| VaR查询每日接收的模拟损益数据量(记录数) | 约647741358 |
| 实现的报表总数 | 300 |

1. 报表复杂度分析

用户总需求的报表数量为300张左右，按照一阶段确定的报表数量进行推算如下表所示

* 按照报表的周期分析

|  |  |  |  |
| --- | --- | --- | --- |
|  | **一阶段** | **全部** | **说明** |
| 日报 | 109 | 225 |  |
| 月报 | 33 | 69 | 月报数量占日报数量30%，因此对月初访问的峰值系数计算为1.3 |
| 年报 | 3 | 6 |  |
| 小计 | 145 | 300 |  |

* 按照报表样式来进行场景分析

|  |  |  |  |
| --- | --- | --- | --- |
|  | **一阶段** | **全部** | **说明** |
| 固定报表 | 103 | 213 | 表样固定，前端只需要从后台数据库中直接获取加工结果  或者是进行简单的固定的条件过滤（如报表期数，机构信息等） |
| 动态查询 | 42 | 87 | 表样不固定，根据用户即席输入或选择的查询条件动态生成查询SQL  或虽然表样固定，但也需要根据用户即席选择确定多个维度的条件，来生成查询结果的场景 |

* 按照报表查询过程涉及到的报表层数据的存量数据分析

对于报表查询的过程来说，报表层存放的存量数据量的大小也是影响到查询效率的因素之一。对于条件复杂的查询过程，返回数据量不大的情况下，其响应过程也较为迅速。而即便是条件简单的查询，如果涉及到存量数据巨大，需要进行检索的过程也会增加整个过程的耗时。

|  |  |  |  |
| --- | --- | --- | --- |
| **应用层** | **一阶段** | **全部** | **说明** |
| 明细数据类查询 | 35 | 72 | 1. 涉及到交易明细的查询。 2. 应用层的存量数据每日存储的交易数量级别在10万 |
| portfolio类查询 | 80 | 165 | 1. 在交易明细级别上的portfolio级别汇总。 2. 汇总粒度在1:4~1:6之间 3. 汇总表每日的数据量在1~6万条之间 |
| 高度汇总查询 | 30 | 62 | 1. 根据用户定义的表样或者指定的维度需求，预先进行了高度汇总加工处理，汇总表每日数据量不大，可能在几百条到数千条记录不等 |
| 小计 | 145 | 300 |  |

* 按照报表的访问复杂度进行场景分析

考虑到以上多种因素，进一步将报表查询的场景划分成为四种，每种情况都考虑各种方面的性能因素影响。对于每一种场景的查询都设定其基准的响应时间。最后按照每种场景的并发度进行加权求和后除以CPU个数就能够得到用户的平均等待时间，这个等待时间要能够满足用户对于报表查询的响应时间指标。

|  |  |  |  |
| --- | --- | --- | --- |
|  | **一阶段** | **全部** | **说明** |
| 简单查询 | 29 | 60 | 1. 查询条件简单 2. 无进一步分组/汇总等操作 3. 无表关联或简单的码表关联 |
| 一般查询 | 73 | 150 | 1. 查询条件不复杂(Where条件不超过5个) 2. SQL查询需要进一步分组/汇总 3. 关联表的个数不超过5个，没有超过10万条记录的大表关联操作 4. 对于时间跨度范围较大(一个月或多个月)的查询不会到明细级别或者portfolio级别 |
| 复杂查询 | 29 | 60 | 1. 查询条件较为复杂 2. SQL查询需要进一步分组/汇总 3. 关联表的个数大于等于5个，或出现超过10万条记录的大表之间的关联操作 4. 在明细及portfolio级别的报表层数据中进行大跨度的时间范围查询 |
| 灵活查询 | 14 | 30 | 1. 用户提交的查询起止时间不确定 2. 或者在明细级别的表中进行大范围的数据查询 3. 提交之前查询条件不固定，在查询过程中根据用户设定的条件临时生成各种查询条件 4. 查询范围较广 5. 灵活查询中，用户指定的多个维度，一般在6~7个左右 |
| 小计 | 145 | 300 |  |

* 客户要求查询响应时间

用户对IMA系统的报表查询平均响应时间不超过\_\_8\_\_\_秒。

**估算步骤**

1. 按一般时间段15个并发用户，峰值时间20个并发用户进行估算

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **占比** | **一般时段** | **峰值时段** | **前端用户等待时间（需要补充）**  **单位（秒）** |
| 简单查询 | 20% | 3 | 4 | 2 |
| 一般查询 | 40% | 6 | 8 | 4 |
| 复杂查询 | 27% | 3 | 5 | 12 |
| 灵活查询 | 13% | 3 | 3 | 9 |
| 总人数(并发) |  | **15** | **20** |  |

1. 计算公式

(T1\*Q1+T2\*Q2+T3\*Q3+T4\*Q4)/N≤TW

其中：

T1/Q1：简单查询前端用户等待时间/用户数

T2/Q2：一般查询前端用户等待时间/用户数

T3/Q3：复杂查询前端用户等待时间/用户数

T4/Q4：灵活查询前端用户等待时间/用户数

N: CPU数量

TW：前端用户平均等待时间

对于一般时段，代入参数后，计算公式如下

（3\*2+6\*4+3\*12+3\*9）/ N = TW

对于高峰时段，带入参数后，计算公式为

（4\*2+8\*4+5\*12+3\*9）/ N = TW

综上，CPU数量推算如下表所示：

|  |  |  |
| --- | --- | --- |
| **CPU数量** | **前端用户平均等待时间(一般时段)单位（秒）** | **前端用户平均等待时间(高峰时段)单位（秒）** |
| 4 | 23.25 | 31.75 |
| 8 | 11.63 | 15.88 |
| 12 | 7.75 | 10.58 |
| 16 | 5.81 | 7.93 |
| 20 | 4.65 | 6.35 |

按照上面推算，如果满足用户对平均响应时间为8秒的要求（包括高峰时段），生产环境服务器的CPU个数总数应为16个。

对于生产环境将采取双机集群的方式，每台WEB报表服务器的配置如下

|  |  |
| --- | --- |
| 项目 | 配置 |
| CPU | 8C |
| 主频 | 3.3GHz |
| 内存（GB） | 32G |

## AIX+PC Server(VaR+报表服务器)资源配置建议方案

根据科技部的建议，实际生产环境使用power780的服务器配置，因此，适当降低单机的CPU个数，提高服务器的CPU和内存的比例关系。实际的资源配置方案如下：

1. 生产环境：

作为系统生产用环境。采用集群的方式来保证系统的安全性和高可用性。

|  |  |  |  |
| --- | --- | --- | --- |
| **设备**  **类型** | **硬件配置** | **软件环境** | **数量** |
| VaR计算引擎服务器 | 服务器配置建议：PC Server   1. CPU：4C8核 2. 内存: 64GB 3. 内置磁盘： 4. 8 \* 300GB 1.5w转速高速硬盘(RAID 1) | 1. OS: Red Hat Linux企业版6.1 2. VAR查询引擎及其他分析程序 | 4 |
| 数据库/ETL服务器 | 小型机配置建议：IBM Power 750   1. CPU：10C； 2. 内存：60GB 3. 内置磁盘：4\* 600GB 1.5w转速高速硬盘(RAID 1)； | 1. OS：AIX 6.1 2. DB：Oracle 11.2.0.3 RAC 3. JobControl 1.10 | 2 |
| Web/报表服务器 | 小型机配置建议：IBM Power 750   1. CPU：8C核 2. 内存：32GB 3. 内置磁盘：4 \* 600GB 1.5w转速高速硬盘(RAID 1) | 1. OS: AIX 6.1 2. WebSphere 8.0.0.2 3. Cognos 10.0 4. 报表前端集成工具 | 2 |
| 负载均衡服务器 | 1. F5 BIG-LTM-1600-4G-R |  | 1 |
| 磁盘阵列1 | 磁盘阵列配置建议：；   1. 10TB磁盘裸容量(SAN)； 2. RAID 0+1 | 为数据库和报表服务器共享使用 | 1 |
| 磁盘阵列1 | 磁盘阵列配置建议：；   1. 20TB磁盘裸容量(NAS)； 2. RAID 0+1 | 为VaR服务器单独使用 | 1 |
| 磁带库 | 磁带库配置建议：   1. 由中行根据实际情况进行配置 |  | 1 |
| 光纤交换机 | 16口光纤交换机（16口激活） |  | 2 |
|  |  |  |  |

## 性能方面的考虑

### 最小化数据量

通过计算减少数据量比用技术调整来提高性能有效得多。例如，在作业的早期使用一个过滤工具来减少75%的数据量与购买更多/更快的服务器相比来达到同样的目的，前者能更有效地提高性能。

尽可能早地使用复本/转换工具来删除字段和缩窄字段。

尽可能早地使用过滤工具来删除不必要的记录。

可以将过滤工具和转换操作结合为一个步骤。如果数据流处理逻辑发现分开的过滤工具和重新格式化，具有最大的减小因素的那个应该先发生。

### 延迟数据膨胀

当反范式化或增加一个作业内的数据量时，尽可能迟地进行，这不仅仅是指在作业内，也包括多个作业间的情况。要尽可能地延迟一个ETL周期内的数据膨胀操作。

### DB装载性能

在DB装载中，性能可能是由许多变量决定的。估计每个变量的相对重要性是不可能的。有必要评估一个操作的总成本。我们可以看出，总成本是由整个操作的变量和固定成本决定的。在有些情况中，要想在付出相对高的变量成本的同时避免一个固定的成本，从而使变量的高成本变得不很严重是可能发生的。同样地，固定部分的相对成本也可以很小，使人们能够在执行每一操作时都能够购买一个。

在短期内，在DB中重新建立索引的成本是不固定的，但是相对比较低。当一个表中的列数变多，建立索引增量成本是低的，但是，总成本确很高。由于这个原因，人们可以将索引的建立当作一个相对固定的成本，形成一个单一的装载周期。任何单一的装载周期不会很明显地增加数据库的总大小。

另一个与建模DB装载性能相关的变数是是否采用数据库的强制约束条件。此外，总成本的估计应该在基于固定成本删除索引然后使用RI限制条件和装载的增量成本采用强制约束条件进行比较。

### 分区

大部分作业在一系列输入文件后的第一个步骤是进行分区。

作业的并行程度应该与数据库分区去耦合。在非常高性能环境中，也有例外情况，但是如果让数据库分区来驱动并行将降低可扩展性和可操作性。实际上，数据库分区往往是不稳定的，当数据库分区级别改变时，进行数据库分区的作业将失败。最好的方法是使用配置文件管理并行作业。

应该对大部分分区加键值。

加了键值的分区很少会没有帮助作用，应使用循环，这是由于循环常产生较少的倾斜。

应少使用范围、随机、或模块分区且只有在非常特殊的情况下才使用这种分区。

范围的分区不能用来平等地跨不同分区分布数据来最小化倾斜；它会限制可扩展性，业务的外部变化会产生倾斜。

范围的分区与一个序列结合能用于整体的排序，但是很少需要整体排序且可以由一个加键值的分区、一个本地排序和一个排序漏斗完成。

考虑下游的处理及其加键的操作并最小化再分区。

### 并行处理

根据数据量、可用资源和作业的重要性程度，人们能够确定DataStage作业的并行程度。在这一阶段，每一作业都有一个预先定义的APT配置文件来控制并行注意。配置文件的名称被参数化，且在运行时间可以改变名称。我们的目标是利用一个工具来检查可用的资源并动态地生成配置文件和分派给服务器。

Important information that does not fit into one of the preceding sections…