Atitit 效率提升法细则 v3 t028.docx

Atitit 提升效率细则

目录

[1. 目标 2](#_Toc18856)

[1.1. 配置化增加扩展性 尽可能消除编译 方便增加 调整业务逻辑 2](#_Toc9154)

[1.2. 统一接口（参数传递完整化，同时附加传递xml selectid） 2](#_Toc16509)

[1.3. QL动态通用化，消除mybatis内大量的查询与简单操作类似sql 2](#_Toc22435)

[1.4. 尽可能消除部署 2](#_Toc16098)

[1.5. Scheme free 3](#_Toc25861)

[2. 提升语言级别4gl 4.5gl 5G 3](#_Toc31016)

[2.1. 语言开发效率与可读性级别表 3](#_Toc6791)

[2.2. Sql，存储地方可选择放在mybatis xml 或者数据库sp（可以免部署），都可以实现免编译目标，配置化 3](#_Toc2243)

[2.3. 业务逻辑尽可能使用4gl(比如sql 工作流 规则引擎等类似东东) 3](#_Toc17392)

[2.4. 业务逻辑适当的使用脚本3.5g等实现 3](#_Toc18937)

[3. 动态化 统一化 适当抽象化 4](#_Toc30115)

[3.1. 动态语言 适当的使用脚本（js php xml sql等） 4](#_Toc19462)

[3.2. 动态接口 ，统一接口 4](#_Toc17577)

[3.3. ，动态对象 4](#_Toc3979)

[3.4. servless模式 4](#_Toc22516)

[4. 利用现有机制配置化 5](#_Toc27964)

[4.1. 利用Mybatis实现业务逻辑配置化（脚本化工作流模式） 5](#_Toc25801)

[4.2. 多利用数据库 view sp 触发器等机制实现配置化 5](#_Toc3877)

[4.3. 多利用数据库user与权限配置机制实现权限配置 5](#_Toc30582)

[4.4. 多使用数据库的各种约束校验机制 unique索引，外键，触发器 约束等机制 5](#_Toc30982)

[4.5. 多使用数据库定时器 5](#_Toc12990)

[4.6. 跨库调用，可以把次模式看成看成一个调用socket非文本模式接口即可（ 通过数据库驱动） 5](#_Toc16527)

[4.7. 使用脚本实现配置化 5](#_Toc17314)

[4.8. 多利用数据库报表功能 5](#_Toc23687)

[5. 前后端数据查询操作语言QL 5](#_Toc29806)

[5.1. 自定义QL 4.5g 5](#_Toc27172)

[5.2. Jpql不错 替代rest的一个选择 6](#_Toc24884)

[5.3. 受限sql 6](#_Toc537)

[5.4. Apijson等 6](#_Toc8476)

[6. 设计简化 6](#_Toc17193)

[6.1. 简化流程与层次 6](#_Toc4913)

[6.2. 客户端直连数据库，瘦服务端 6](#_Toc9791)

[7. 数据传输与存储层面的优化 6](#_Toc12944)

[7.1. Scheme free模式 多使用nosql json 半结构化数据 6](#_Toc4189)

[7.2. Mysql5.7以上可多多 使用json数据 6](#_Toc13609)

[7.3. 子母表设计可使用json集合模式等 6](#_Toc30940)

[7.4. 适当的反范式设计 6](#_Toc4466)

[8. 优先选择最简单方案（高中低三挡方案定制） 7](#_Toc28369)

[9. 使用java js来扩展sql 与mybatis 7](#_Toc10937)

[9.1. 使用mybatis提供的ongl调用java 7](#_Toc24226)

[9.2. 使用自定义event模式来进行 7](#_Toc8440)

[10. 问题 7](#_Toc16634)

[10.1. 复杂业务与多语句的问题 7](#_Toc26709)

# 目标

## 配置化增加扩展性 尽可能消除编译 方便增加 调整业务逻辑

查询与简单数据操作直接QL解决

复杂逻辑可以通过脚本 sql 或类似工作流 逻辑引擎的东东来消除

## 统一接口（参数传递完整化，同时附加传递xml selectid）

避免散落在多处，实现业务层接口统一化

## QL动态通用化，消除mybatis内大量的查询与简单操作类似sql

实现数据层接口统一化

综合流程

综合数据类业务查询操作实践

JPQL》>jpql AST》 转换为sql ast 》转换为sql ，执行，返回数据

## 尽可能消除部署

Webdav远程编辑器模式 + sp

## Scheme free

# 提升语言级别4gl 4.5gl 5G

## 语言开发效率与可读性级别表

|  |  |  |
| --- | --- | --- |
| 语言级别 | 具体实现 | 范例（分组查询操作日志） |
| 5g | 完全自然语言实现 | 查询操作日志表，按照人员id分组，然后统计每个人的操作次数 |
| 4.9g | 基本自然语言，可用注解标识出精确化归一化，方便解析识别构造AST | @q查询@t操作日志表，按照@grp(人员id)分组，然后统计每个人的@sum操作次数 |
| 4.5g | 受限自然语言（ql语法）  比较严格的NL自然语言语法 | 查询(操作日志表).按照分组(人员id).获取(人员id,记录条数) |
| 4gl | (mybatis xml ,sql 工作流 规则引擎等 ) | Select 人员id,count(id) From 操作日志表groupby 人员id |
| 3.5 | 各种script （js python Php等） |  |
| 3gl | Java net c# golang等编译型语言 |  |
| 2gl | C c++类 |  |
| 1gl | Asm汇编 |  |

## Sql，存储地方可选择放在mybatis xml 或者数据库sp（可以免部署），都可以实现免编译目标，配置化

## 业务逻辑尽可能使用4gl(比如sql 工作流 规则引擎等类似东东)

## 业务逻辑适当的使用脚本3.5g等实现

# 动态化 统一化 适当抽象化

## 动态语言 适当的使用脚本（js php xml sql等）

## 动态接口 ，统一接口

## ，动态对象

## servless模式

# 利用现有机制配置化

## 利用Mybatis实现业务逻辑配置化（脚本化工作流模式）

## 多利用数据库 view sp 触发器等机制实现配置化

## 多利用数据库user与权限配置机制实现权限配置

## 多使用数据库的各种约束校验机制 unique索引，外键，触发器 约束等机制

## 多使用数据库定时器

## 跨库调用，可以把次模式看成看成一个调用socket非文本模式接口即可（ 通过数据库驱动）

## 使用脚本实现配置化

sql 或类似工作流 逻辑引擎的东东来消除

## 多利用数据库报表功能

# 集成化

## Wolrd excel等

## 少量数据情况下直接集成化excel，免得开发ui与crud概念，只需要读取就可以了

## 直接json配置化通用配置化等

## 多使用proper配置化文件 与ui集成

# 前后端数据查询操作语言QL

## 自定义QL 4.5g

1.2.效果概览

查询表格(操作日志表),条件( 操作人=小新),and(时间=2018),排序(时间),翻页(页数=7,每页=50)";

经过全面的解析，转换为sql

SELECT \* FROM 操作日志表 WHERE 操作人 = '小新' AND 时间 = '2018' ORDER BY 时间 LIMIT 50 OFFSET 300

## Jpql不错 替代rest的一个选择

## 受限sql

## Apijson等

# 设计简化

## 简化流程与层次

## 客户端直连数据库，瘦服务端

特别是管理类系统（包括web系统后台管理系统），可以不用中间层中转。。直接客户端收发sql， 业务逻辑也可放在客户端处理。瘦服务端

# 数据传输与存储层面的优化

## Scheme free模式 多使用nosql json 半结构化数据

## Mysql5.7以上可多多 使用json数据

数据库 nosql mongodb 直接json 对用redis

## 子母表设计可使用json集合模式等

## 适当的反范式设计

范式设计主要是为了减少数据冗余，对存储空间简约有帮助，时过境迁，今天存储已经很便宜。。适当的反范式设计同时也可提升性能与开发效率

# 优先选择最简单方案（高中低三挡方案定制）

根据团队规模，公司规模，项目需求来定制方案完善

语言选择java net js python php xml sql等

比如微服务的nginx模式，dobbo模式，springcloud模式多种实现方案

数据分片的 msql oracle实现模式，读写集群，分区机制模式，分库，数据库中间件模式

Share jdbc mycat等

大公司的解决方案慎用，绝大部分不适合中小公司，过重

# 使用java js来扩展sql 与mybatis

## 使用mybatis提供的ongl调用java

## 使用自定义event模式来进行

可支持sp里面调用

# 问题

## 复杂业务与多语句的问题

处于安全考虑，复杂业务多语句放入后端，第一条语句作为触发机制，然后可以使用触发器级联下级sp业务。。这样就解决了jpql处理复杂业务的问题

或者使用mybatis调用 selectid 解决得了。。

也可以窜入sp id，直接调用，免部署