

德塔 socket 数据库 例子

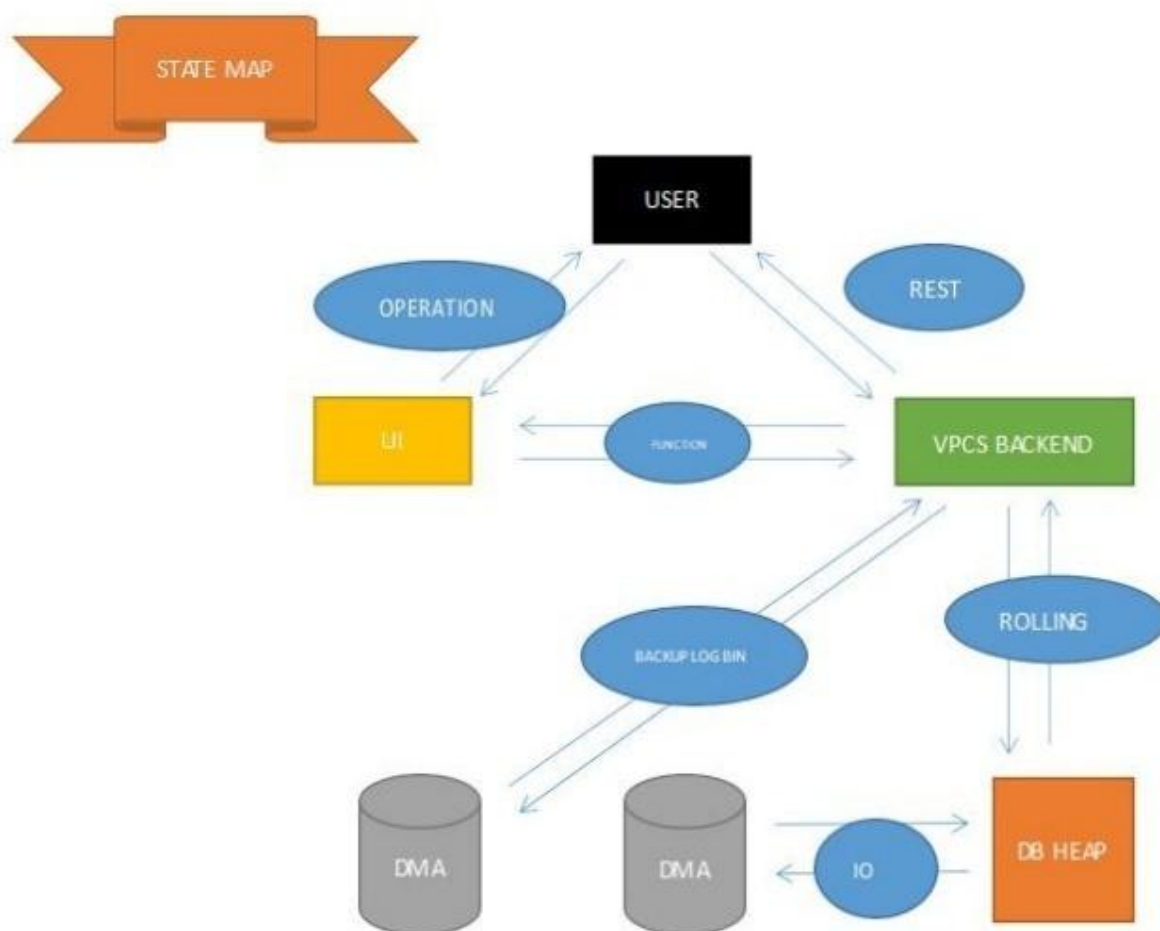


Alkaid 罗瑶光的视频

- 6 播放

Socket rest TCP 握手协议，

- 1 德塔数据库的 admin 界面采用 web 页进行配置操作。refer page 376
- 2 web 页配置操作采用 TCP 握手访问模式，基于 socket 的 http 请求握手。refer page 464~
- 3 德塔数据库将 socket 握手进行线程封装，然后多线程组织页面。refer page 392，
- 4 封装和组织页面设计过程逐步进行优化形成 VPCS 后端管理体系。refer page 383, 476

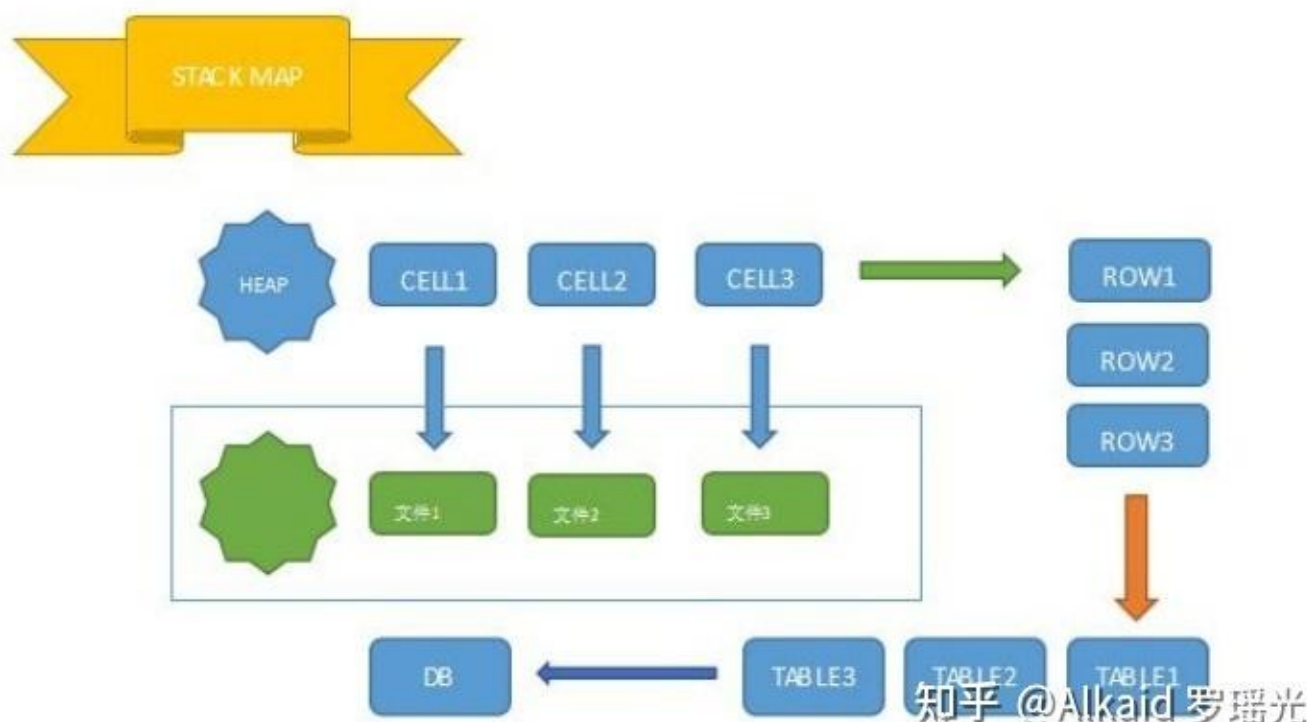


知乎 @Alkaid 罗瑶光

文件数据库，

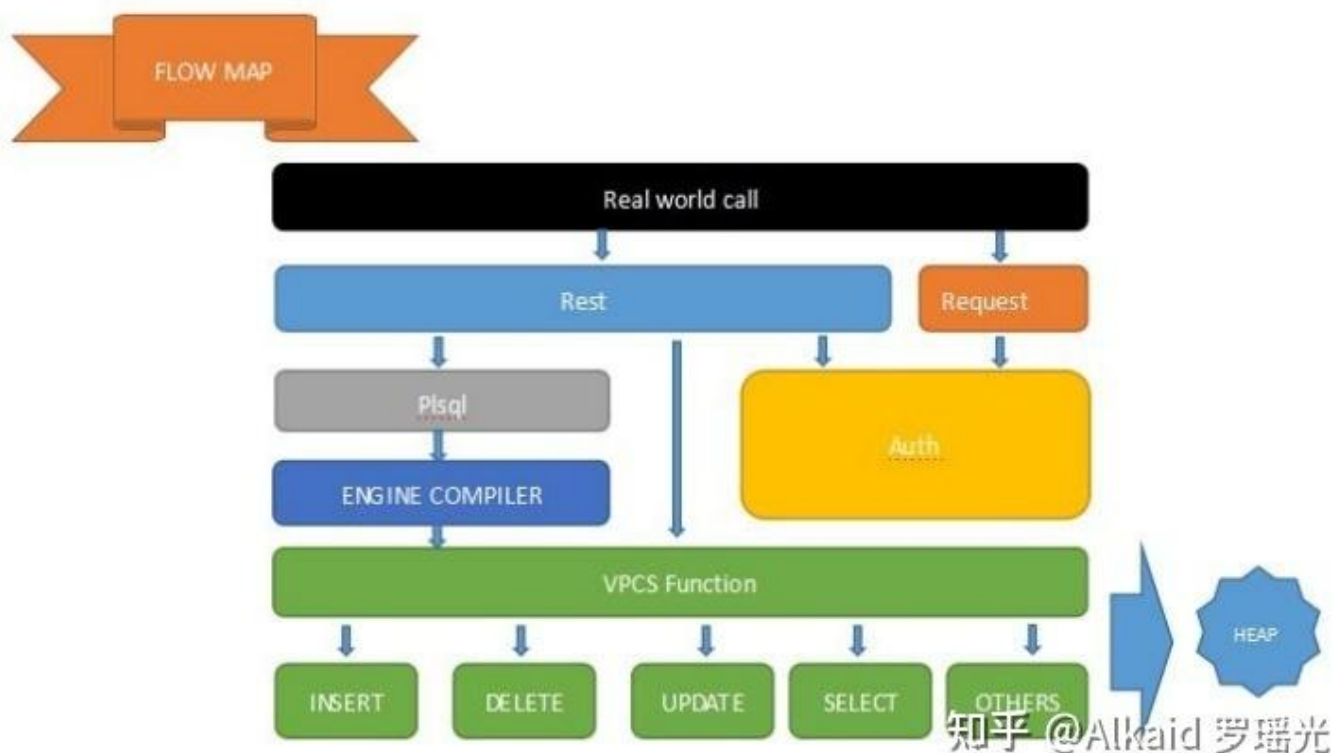
1 德塔数据库的数据存储是一种文件存储模式。refer page 408, 409, 469, 473

2 文件的读写进行子集，行，表，映射，表头，按 1 范式分类。refer page 375, 434,



3 数据库的数据读写支持加密。refer page 见元基加密

4 每一个文件不但有物理空间，还有相应的内存空间。refer page 375

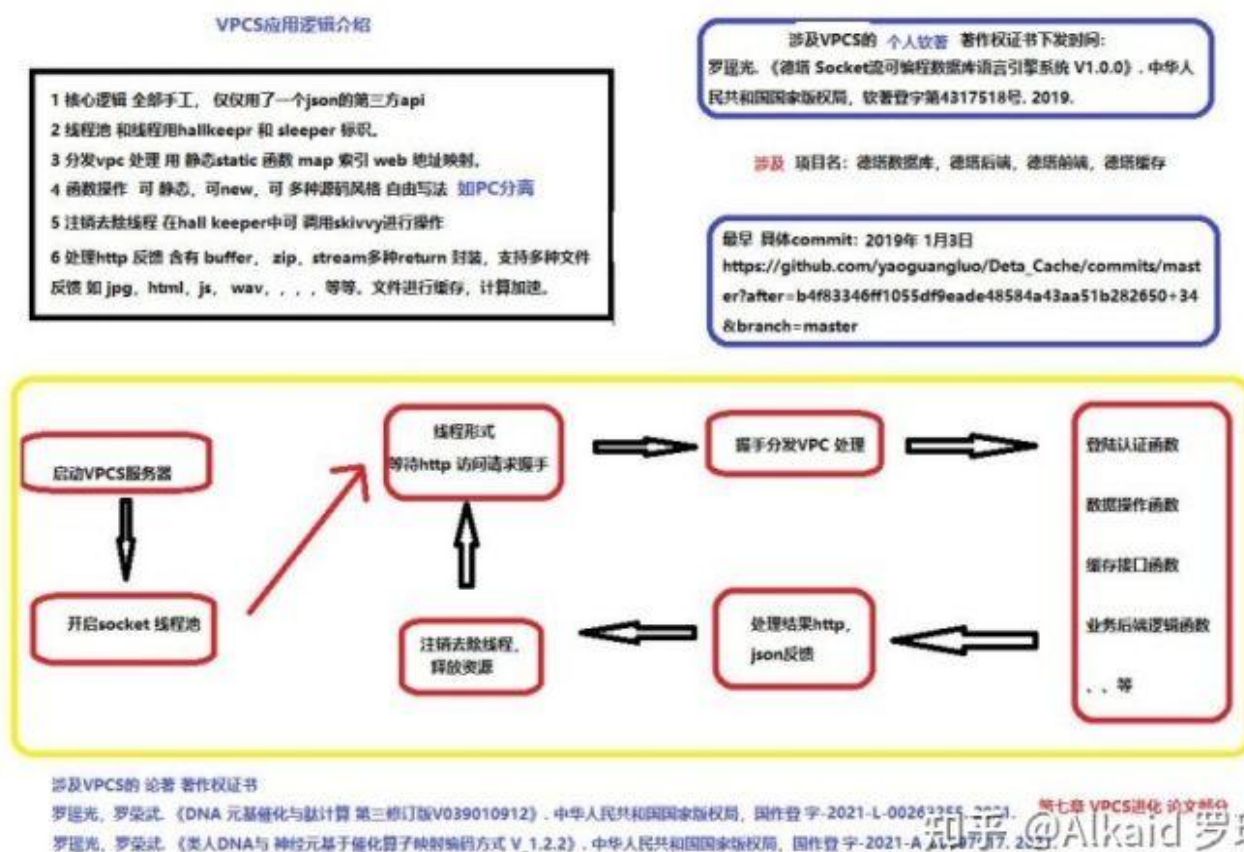


VPCS 服务器，

- 1 VPCS 服务器支持每秒 400 万 QPS 的 web 请求。refer page 389,
- 2 VPCS 服务器采用 TCP rest request 模式，标准化 http response。refer page 388, 395
- 3 VPCS 服务器可自由设计前端和后端集成。refer page 见德塔官网 和 养料经 admin 两个实例
- 4 VPCS 服务器完全支持 post 个 get 2 种请求模式，可扩展。refer page 481, 488

DNA 元基催化与肽计算，第四次修订版本

376



VPCS 调度架构，

- 1 VPCS 服务器包含 视觉模块，处理模块，控制模块，资源模块。refer page 396, 394, 392, 383
- 2 每一种模块有各自的名称标识 和 内存标识，方便精确查找。refer page 492, 493,
- 3 VPCS 服务器包含执行者-生产者-造梦者-sleeper，管理者-分配者-登记者-HallKeeper，运维者-服务员-清洁员-skivvy 3 个模式。refer page 392, 394,
- 4 支持控制与执行分离，线程与资源分离。refer page 385~389, 486, 490, 492

作者最早设计 VPCS 服务器的动机，是为了弥补 VPC 的计算过程观测困难的问题。因为作者设计的 VPC 是采用 springboot + mybatis 的结构，底层全是是开源插件的封装，很多核心源码又不能调试仅仅通过几个 log 和 try catch 给作者带来了无形的压力（作者的思维很简单，就是自己写个服务器，能够调试断点从头断到尾），于是有计划从无到有进行设计一个 TCP/IP 的 SOCKET 协议做服务器 HTTP 请求。作者当时没有想到，一个这样的小动机给带来了丰厚的回报，VPCS 目前成为了 DNA 元基映射编码算子 的核心组成部分。

PLSQL 语言,

- 1 德塔 PLSQL 语言是一种从上到下的脚本执行语言。refer page 377,
- 2 德塔 PLSQL 语言包含常用增删改查命令。refer page 406~409, 471, 1035
- 3 德塔 PLSQL 语言支持 join 和 aggregation 高级操作。refer page 419, 431, 435, 438, 447
- 4 德塔 PLSQL 语言行 可批处理, 可拆分。refer page 1035~1041 将例子写入 main, class 编译, 然后 bash boot class 即可。 还可以 bash 定时批处理。

RealWorldSamplesByUsingDETAPLSQLDatabase

```
setRoot:C:/DetaDB;  
baseName:backend;  
tableName:usr:select;  
condition:or:u_id|<=|3:u_id|>|7;  
condition:and:u_email|!equal|321:u_name|!equal|123;  
getCulums:u_id|as|detaId:u_email|as|detaEmail;  
join:backend:usrToken;  
condition:and:u_level|equal|low;  
getCulums:u_id|as|sId:u_level:u_password|as|SSID;  
relation:and:detaId|==|sId;  
aggregation:limit:0|~|1;
```

Compare Tranditioanl SQL:

```
SELECT u.u_id as detaId, u.u_email as detaEmail, t.u_id as sId, t.u_level, t.u_password as SSID  
FROM usr as U  
    INNER JOIN (SELECT t.u_id as sId, t.u_level, t.u_password as SSID  
                FROM usrToken as t  
                WHERE t.u_level equal "low") AS B on U.detaId == B.sId;  
WHERE (u.u_id <=3 || u.u_id>7) && (u.u_email !equal '321' && u.u_name !equal '123');  
LIMIT 0,1;
```

PLSQL 编译器,

- 1 德塔 PLSQL 编译器 用于理解和执行 德塔 PLSQL 语言。refer page 413, 414
- 2 德塔 PLSQL 编译器 包含常见脚本命令计算算子如 条件算子, 比较算子, 包含算子, 离散算子。refer page 419
- 3 德塔 PLSQL 编译器 采用 map 进行的内部中间数据缓存。refer page 431, 432~

PLORM 语言,

- 1 德塔 PLORM 语言 用于 德塔 PLSQL 语言进行函数封装。refer page 1003~

2 德塔 PLORM 语言 有先后顺序，需要遵循 德塔 PLSQL 语言语法。refer page 1019~

```
Map<String, Object> map= null;
try {
    String plsqli= "setRoot:C:/DataDB1;" +
        "baseName:ZYY;" +
        "tableName:"+ tabKey +":select;" +
        "condition:or.ID|<={3000}";
    map= ME.SM.OP.SM.AOP.MEC.SIQ.E.E_PLSQL.E.E_PLSQL(plsqli, true);
} catch (Exception e1) {
    //准备写回滚
    e1.printStackTrace();
}
return map;
```

德塔 PLSQL

```
Map<String, Object> map= null;
try {
    PLORM_C orm= new PLORM_E();
    map= orm.startAtRootDir(rootPath).withBaseName(baseName)
        .withTableSelect(tabKey).withCondition("or")
        .let(conditionSubject).lessThanAndEqualTo(conditionObject)
        .checkAndFixPlsqliGrammarErrors()
        .checkAndFixSystemEnvironmentErrors()
        .finalE(unTest).returnAsMap();
    //map= org.plsqli.db.plsqli.imp.E_PLSQLImp.E_PLORM(orm, true);
} catch (Exception e1) {
    //准备写回滚
    e1.printStackTrace();
}
return map;
```

德塔 PLORM

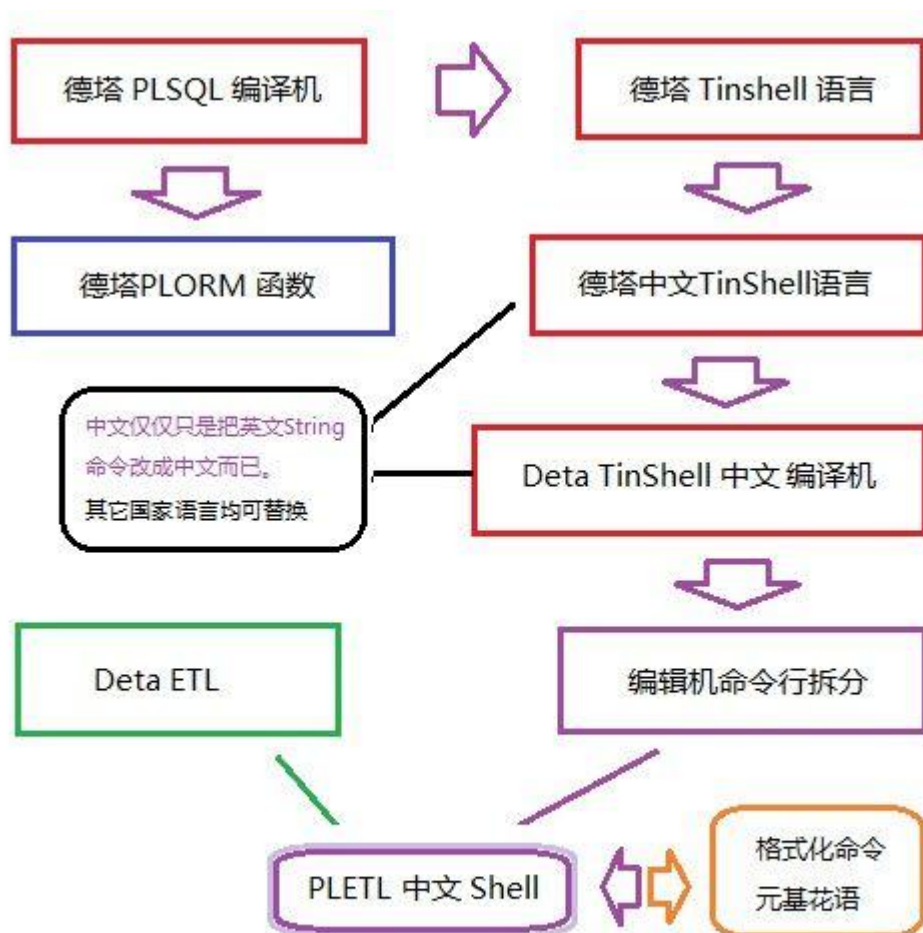
```
select *
from tabKey
where ID<=3000
```

普通带变量 SQL

知乎 @Alkaid 罗瑶光

3 德塔 PLORM 语言 对比 德塔 PLSQL 语言 用于一些不需要配置的 nosql 的场景，类似 hibernate 对比 ibatis。refer page 1019~

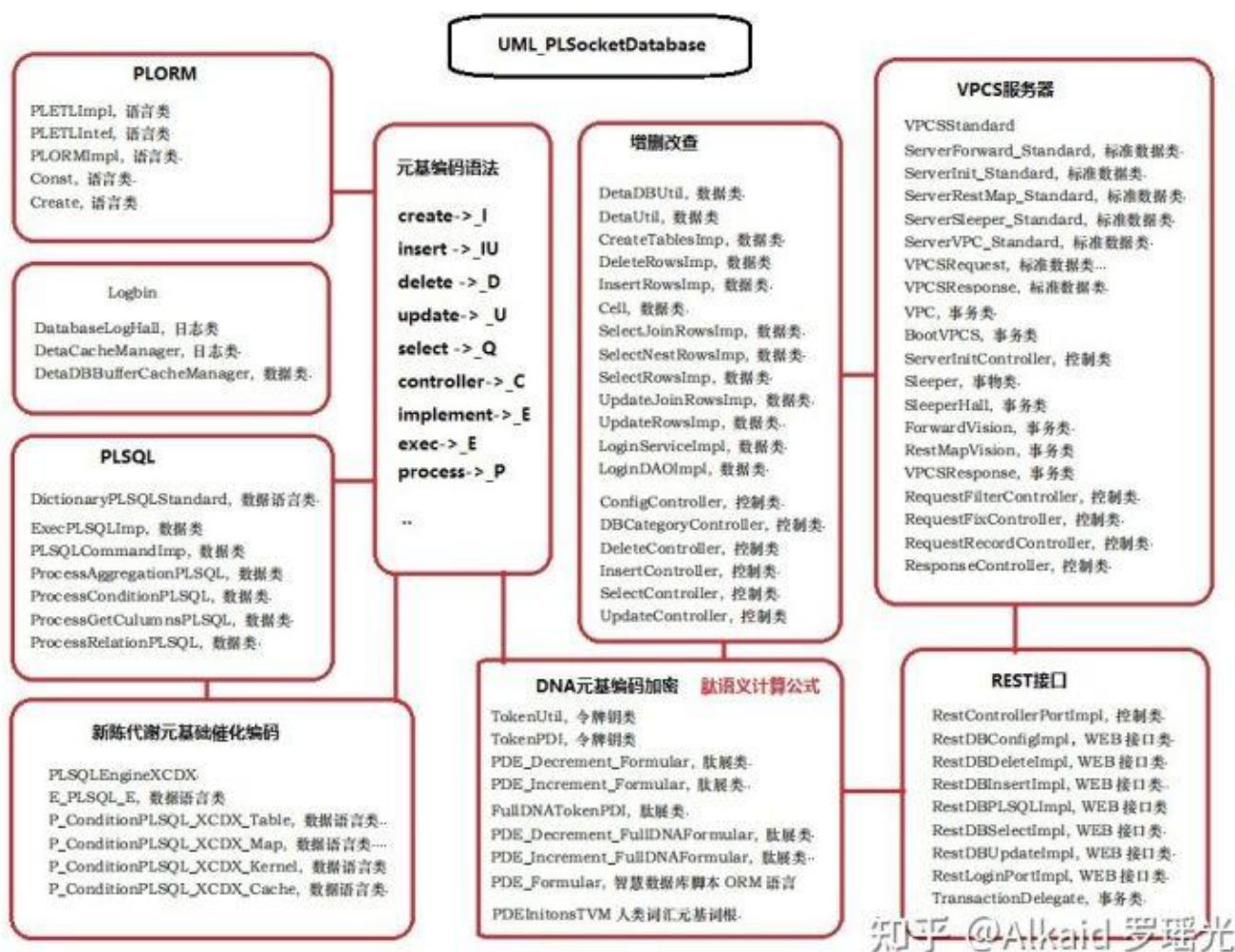
4 VS hibernate 对比 ibatis 的不同，德塔 PLORM 语言 另外也是 德塔 PLSQL 的上层语言。refer page 1019~



德塔shell语言演化路线图

知乎 @Alkaid 罗瑶光

德塔的 PLORM 和 PLSQL 的引擎出现，作者开始有信心将其优化成 节点执行的命令行脚本模式，于是之后的 TinShell 和 PLTinShell，PLETL Shell 诞生了。这个 PLETL 体系弥补了 当前世界按语言理解方式来模拟神经网络计算的映射空白。



灾后重建，

- 1 德塔数据库包含 logbin 系统。refer page 398,
- 2 德塔数据库包含 logbin 系统基于单个写操作进行 log 保存 并行加密成文件。refer page 399
- 3 单个写操作作用时间戳作和写增量序列进行对应标识，避免混乱。refer page 399
- 4 德塔数据库包含 logbin 系统 并支持热备和错误写 实时 rollback 检测。 refer page 398

德塔的 logbin 系统，一开始是设计在 try catch 中，因为德塔数据库融合了 cache 和 DMA 两种存储系统，于是，作者将 logbin 的 rollback 进行先内存模拟执行写操作，成功后再执行物理写操作，并记录操作日志。如物理写操作还失败，就 rollback 到上次写请求。这种 3 步 logbin 机制，作者认为 高安全性。

章节的著作权文件列表：

1. 罗瑶光。《德塔 Socket 流可编程数据库语言引擎系统 V1.0.0》。中华人民共和国国家版权局，软著登字第 4317518 号。2019.

2. 罗瑶光, 罗荣武. 《类人 DNA 与 神经元基于催化算子映射编码方式 V_1. 2. 2》. 中华人民共和国国家版权局, 国作登字-2021-A-00097017. 2021.
3. 罗瑶光, 罗荣武. 《DNA 元基催化与肽计算第二卷养疗经应用研究 20210305》. 中华人民共和国国家版权局, 国作登字-2021-L-00103660. 2021.
4. 罗瑶光, 罗荣武. 《DNA 元基催化与肽计算 第三修订版 V039010912》. 中华人民共和国国家版权局, 国作登字-2021-L-00268255. 2021.
5. 类人数据生命的 DNA 计算思想 Github [引用日期 2020-03-05] [GitHub - yaoguanguo/Deta_Resource: DNA 微分催化与肽计算, 元基花计算, 进化计算, 遗传计算, 智慧计算, 索引计算, 元基编码, 肽展公式, 大数据计算分析](#)
6. 罗瑶光, 罗荣武. 《DNA 元基催化与肽计算 第四修订版 V00919》. 中华人民共和国国家版权局, SD-2022Z11L0025809. 2022.

文件资源

1 jar: https://github.com/yaoguanguo/ChromosomeDNA/blob/main/BloomChromosome_V19001_20220108.jar

2 book 《DNA 元基催化与肽计算 第四修订版 V00919》上下册

<https://github.com/yaoguanguo/ChromosomeDNA/tree/main/元基催化与肽计算第四修订版本整理>

3 函数在 git 的存储地址: demos

Github: <https://github.com/yaoguanguo/ChromosomeDNA/>

Coding: [公开仓库](#)

Bitbucket: [Bitbucket](#)

Gitee: [浏阳德塔软件开发有限公司 GPL2.0 开源大数据项目 \(DetaChina\) - Gitee.com](#)