**DB & DW**

|  |  |
| --- | --- |
| **审核人** |  |
| **重要性** | 中 |
| **紧迫性** | 中 |
| **拟制人** | 张包峰 |
| **提交日期** |  |

**作者：张包峰**

(版权所有,翻版必究)

**修改记录**

|  |  |  |
| --- | --- | --- |
| **更新时间** | **变更内容** | **变更理由** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

[**作者：张包峰** 1](#_Toc394787100)

[1.1 Parallel Database 4](#_Toc394787101)

[1.2 GreenPlum 4](#_Toc394787102)

[1.3 Vertica 4](#_Toc394787103)

[1.4 AsterData 4](#_Toc394787104)

[1.5 Paraccel 4](#_Toc394787105)

[1.6 TeraData 4](#_Toc394787106)

[1.7 Netezza 4](#_Toc394787107)

[1.8 列数据库 5](#_Toc394787108)

[1.8.1 InfoBright 5](#_Toc394787109)

[1.8.2 Sybase IQ 7](#_Toc394787110)

## Parallel Database

## Google Mesa (再读)

### 背景

Geo-Replicated, Near Real-Time, Scalable Data Warehousing

### 存储方面的设计

数据模型，

更新和查询，

数据版本管理，

多版本管理中，delta version的预处理是主要设计。delta compaction策略。

物理存储和索引，

物理存储形式比较简单，主要是要能节约存储空间。每张table有一个或多个索引，在选择压缩算法上，更看重压缩率和读的时候解压缩的速度，胜过于写的时候压缩的速度。

基本的索引查找是二分的方式。

### 系统架构

Mesa建立在BigTable和Colossus之上，是跨数据中心建立的。每个数据中心上一个实例。

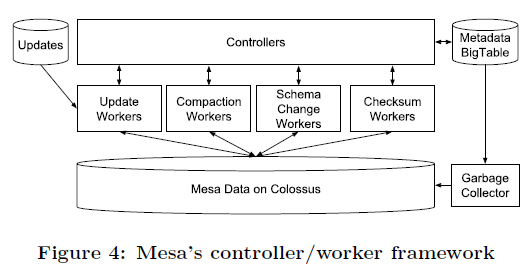
Mesa单个数据中心的实例由两个子系统组成：负责更新/维护的子系统和负责查询的子系统。

两套子系统是解耦的，方便各自的扩容。

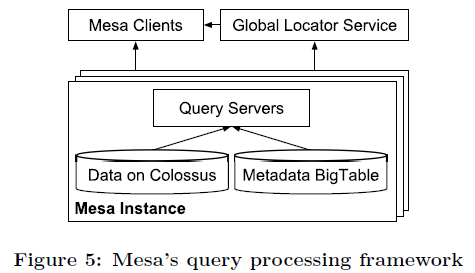
更新子系统

负责更新的子系统需要确保数据在该Mesa实例上的正确性，时效性以及为查询做好优化。后台需要处理很多不同的操作，包括载入更新数据，执行table合并，更改schema信息，给table做checksum。更新子系统主要由controller/workers框架实现。

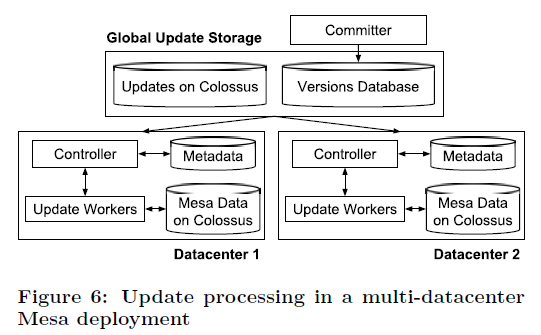
controller类似master节点，负责让worker执行工作，维护所有表的元数据。既充当了一个可扩展的table metadata cache的角色，又充电了work调度和work队列管理者的角色。



查询子系统



### 跨数据中心部署



一致性更新策略

Committer、版本数据库，如何做到查询和更新的搞吞吐量的。

新的Mesa实例

### 其他增强设计

### 经验教训

引用里有丰富的资源可以扩展阅读。

## GreenPlum

## Vertica

## AsterData

## Paraccel

## TeraData

## SAP

## SAS

## Netezza

## Column-Based DB/DW

### InfoBright

论文

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.5467&rep=rep1&type=pdf>

<http://bookshadow.com/weblog/2014/04/22/infobright-datawarehouse/>

Infobright是开源的MySQL数据仓库解决方案，引入了列存储方案，高强度的数据压缩，优化的统计计算(类似sum/avg/group by之类)，

infobright 是基于mysql的，但不装mysql亦可，因为它本身就自带了一个。mysql可以粗分为逻辑层和物理存储引擎，infobright主要实现的就是一个存储引擎，但因为它自身存储逻辑跟关系型数据库根本不同，所以，它不能像InnoDB那样直接作为插件挂接到mysql，它的逻辑层是mysql的逻辑 层加上它自身的优化器。

几大优点：

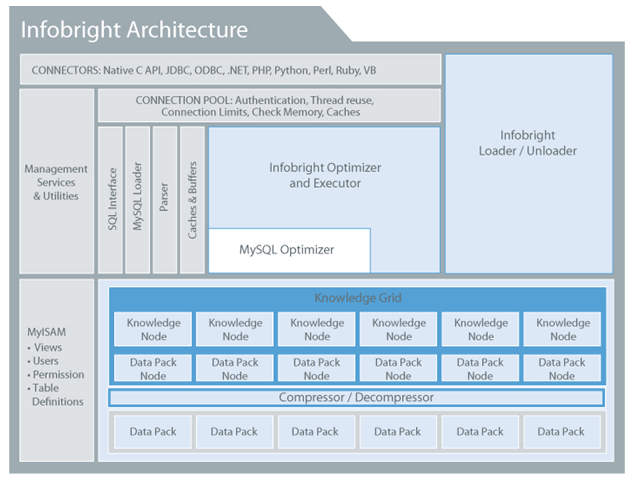
1、高压缩比率，平均压缩比可达10：1，甚至可以达到40:1，我用infobright把3.1G的数据存成不足300M。

2、列存储，即使数据量十分巨大，查询速度也很快。用于数据仓库，处理海量数据没一套可不行。

3、不需要建索引，就避免了维护索引及索引随着数据膨胀的问题。把每列数据分块压缩存放，每块有知识网格节点记录块内的统计信息，代替索引，加速搜索。

4、单一台服务器可以高效地读写30T数据。具有可扩展性，这里是指对于同样的查询，当数据量是10T时，它耗费的时间不应该比1T数据量时慢太多，基本是一个数量级内。

下面是Infobright的架构图：



灰色部分是mysql原有的模块，白色与蓝色部分则是 infobright自身的。

系统结构分析：

跟mysql一样的两层结构，上面的逻辑层处理查询逻辑，下面的是存储引擎。

逻辑层右端的loader与unloader是infobright的数据导入导出模块，也即处理SQL语句里LOAD DATA INFILE … 与SELECT … INTO FILE任务，由于infobright面向的是海量数据环境，所以这个数据导入导出模块是一个独立的服务，并非直接使用mysql的模块。

逻辑层的infobright优化器包在mysql查询优化器的外面，如下面将会提到的，因为它的存储层有一些特殊结构，所以查询优化方式也跟 mysql有很大差异。

存储层最底层是一个个的Data Pack（数据块）。每一个Pack装着某一列的64K个元素，所有数据按照这样的形式打包存储，每一个数据块进行类型相关的压缩（即根据不同数据类型采 用不同的压缩算法），压缩比很高。它上层的压缩器与解压缩器就做了这个事情。

压缩层再向上就是infobright最重要的概念：Knowledge Grid（知识网格），这也是infobright放弃索引却能应用于大量数据查询的基础。它包含两类结点：每个Data Pack Node（知识节点）对应于一个Data Pack，存储该Data Pack的一些统计信息，如min, max, avg, null的个数，甚至不同值的量等等；Knowledge Node则存储了一些更高级的统计信息，以及与其它表的连接信息，这里面的信息有些是数据载入时已经算好的，有些是随着查询进行而计算的，所以说是具备一定的“智能”的。

### Sybase IQ

Sybase可以说是列式数据库的先驱，Sysbase IQ 15 就是Sybase 目前最新的列式数据库。它具有强大的功能，包括数据的快速加载、超高速的分析

性能、强大的业务智能分析、领先的数据建模能力等等。