Infinidb在大数据的实战应用

DTCC

2015中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2015

大数据技术探索和价值发现

IT168.com ChinaUnix ITPUB

赖亿 2015/4/16

目录

- 背景
- InfiniDB的特点
- Infinidb的实战

问题

一个真实的血案：

- 需求：我们在数据库mysql要做基于pv的分析。日均裸数据增量>10g
- 初始方案： 使用innodb

  问题：数据量增加太快,磁盘空间增加太快(40g)

  数据加载太慢了

  最最重要统计类查询太慢了，需要建太多的索引/汇总表

- 改进方案：换成tokudb

  解决问题：数据压缩4倍，空间增加勉强可以接受(10g)

  数据加载快些了4倍左右，勉强可以接受

  未解决：

  最最重要查询太慢了，一个查询5分钟甚至更长，

  优化太痛苦，需要建太多的索引/汇总表

问题

一个真实的血案：

- 需求：我们在数据库mysql要做基于pv的分析。日均裸数据增量>10g
- 初始方案： 使用innodb

  问题：数据量增加太快,磁盘空间增加太快(40g)

  数据加载太慢了

  最最重要统计类查询太慢了，需要建太多的索引/汇总表

- 改进方案：换成tokudb

  解决问题：数据压缩4倍，空间增加勉强可以接受(10g)

  数据加载快些了4倍左右，勉强可以接受

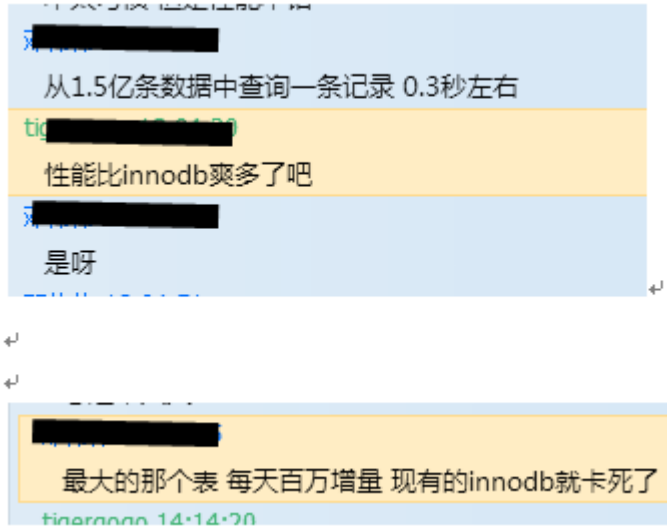
  未解决：

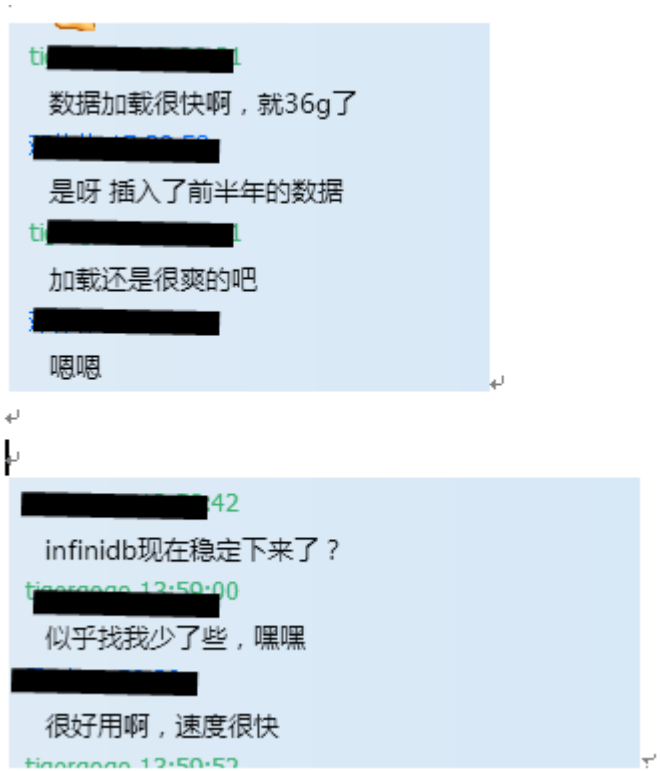
  最最重要查询太慢了，一个查询5分钟甚至更长，

  优化太痛苦，需要建太多的索引/汇总表

解决：--换成infinidb

- 最终方案： 使用infinidb(和最初方案innodb比较)
  - 空间增量2g (原来增量40g)
  - 加载数据 20万/每秒 （原来 <1万/每秒）
  - 查询一般小于1分钟（原来5分钟，甚至20分钟）
  - 免优化（再也不要建index了哦）
- 业务线的反馈

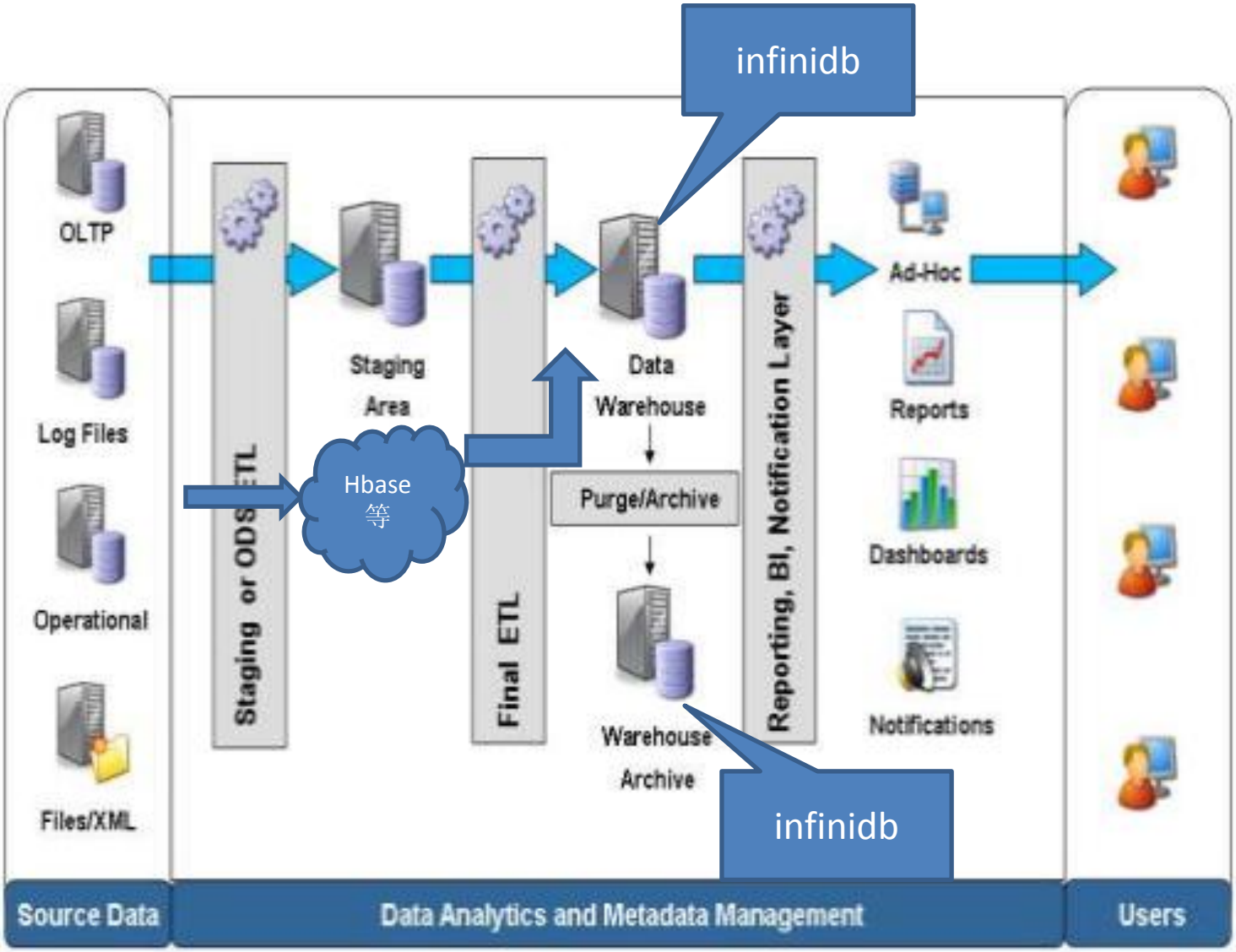




目录

- 背景
- **InfiniDB的特点**
- **Infinidb的实战**

Infinidb的定位



infinidb产品介绍

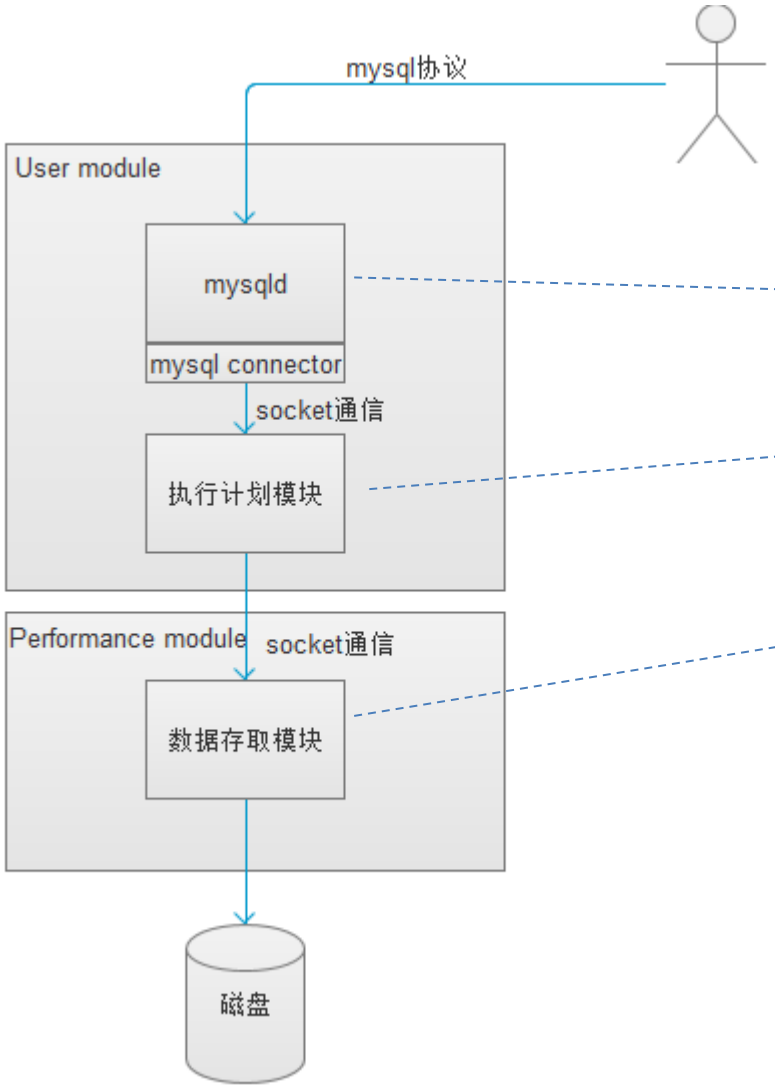
产品特点：

- Mysql协议兼容
- 全功能，支持dml

- 统计类查询10倍
- Load数据快（每秒>10万）
- 压缩率5倍（和裸数据比）

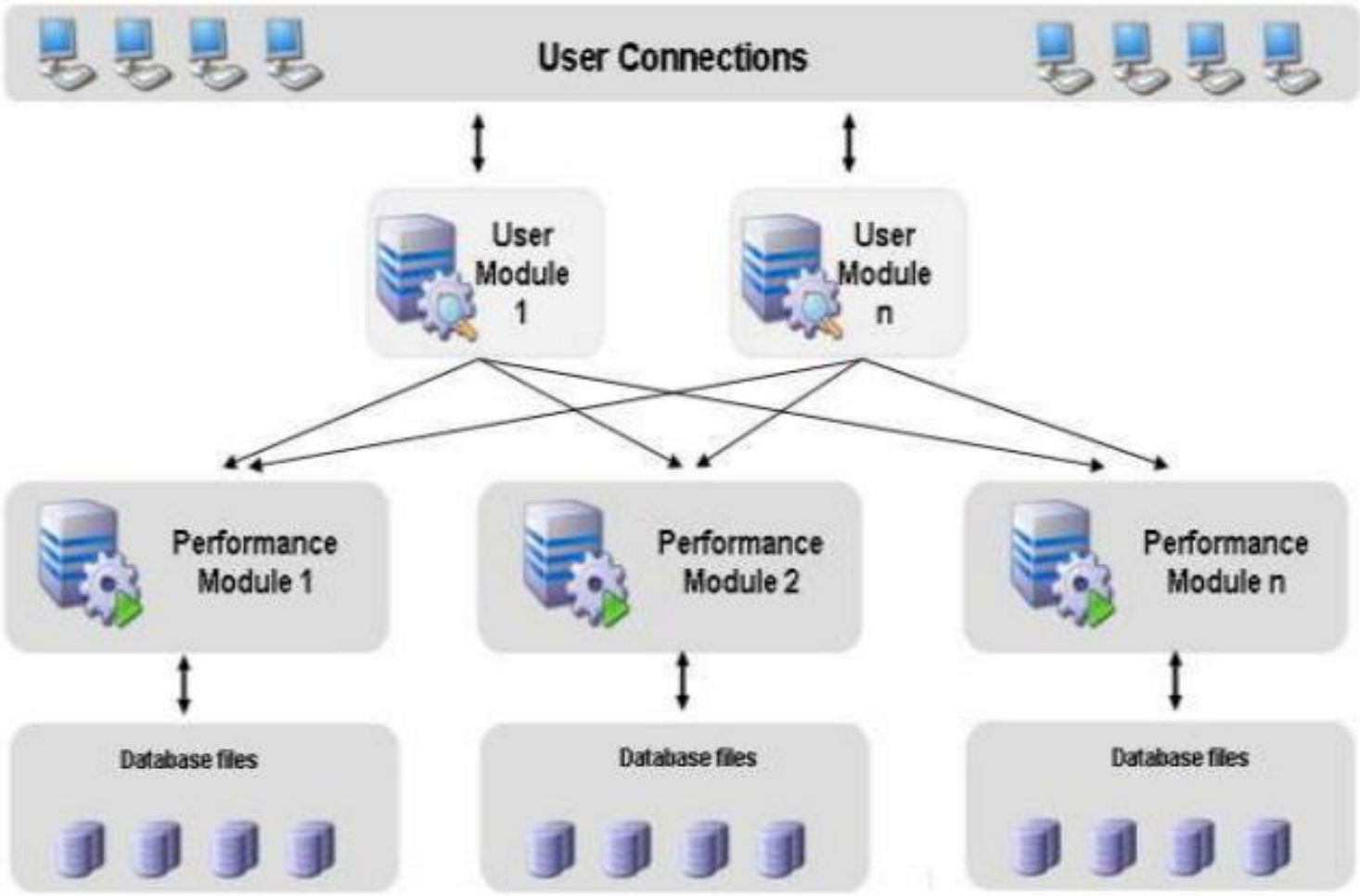
- 免优化

Infinidb的单机构架




```
nit─┬─crond
    ├─exim
    ├─jagentwatchdog.───sleep
    ├─java───35*[{java}]
    ├─mfsmount───60*[{mfsmount}]
    ├─5*[mingetty]
    ├─mysqld_safe───mysqld───3*[{mysqld}]
    ├─perl───java───75*[{java}]
    ├─rsyslogd───3*[{rsyslogd}]
    ├─run.sh───ProcMon─┬─DDLProc───{DDLProc}
    │                  ├─DMLProc───3*[{DMLProc}]
    │                  ├─DecomSvr
    │                  ├─ExeMgr───9*[{ExeMgr}]
    │                  ├─PrimProc───116*[{PrimProc}]
    │                  ├─ProcMgr───2*[{ProcMgr}]
    │                  ├─ServerMonitor───3*[{ServerMonitor}]
    │                  ├─WriteEngineServ
    │                  ├─controllernode───7*[{controllernode}]
    │                  ├─workernode───4*[{workernode}]
    │                  └─4*[{ProcMon}]
```

InfiniDB 分布式框架



集群文件系统（hdfs/gfs）

2013.10.15 支持hdfs –不太建议生产环境用

真实业务性能测试—查询性能

1. 测试条件
   load 29亿的业务数据

查询

```
select vid,sum(vv),sum(fuv),sum(realvv),sum(realfuv),sum(playtime)
from dm_vid_day where date>=20140807 and date<=20140813  and domain=10
and catecode!='-1'    group by vid
infini: 3.130 秒
```

分析类存储引擎InfiniDB – 查询性能对比测试

TPCH测试（以下以1G数据量，150000行用户数据测试）

测试环境:
服务器配置: 虚拟机 4核4G内存 SAS RAID5分区
数据库配置: 2G内存，InfiniDB 4.5.0-1，MariaDB-10.0.11

分析类SQL	MariaDB 执行时间(s)	InfiniDB 执行时间(s)	速度提升%
1	20.18	4.82	318.67
2	16.08	0.69	2230.43
3	14.4	0.95	1415.79
4	13.03	2.37	449.79
5	200.3	1.28	15548.44
6	11.65	0.41	2741.46
7	15.12	4.5	236.00
8	38.31	1	3731.00
9	13.5	3.59	276.04
10	17.22	3.82	350.79
11	1.63	0.3	443.33
12	27.8	2	1290.00
13	22.85	1.72	1228.49
14	11.42	0.89	1183.15
15	44.29	0.91	4767.03
16	1.8	0.67	168.66
17	1.61	4.19	-61.58
18	36.39	3.09	1077.67
19	2.68	9.69	-72.34
20	10.77	3.26	230.37
21	28.67	8.65	231.45
22	1.78	2.08	-14.42

InfiniDB存储 – 为啥查询这样快

数据存储方面，“拆拆拆”：

- 按列拆
- 按行（范围）拆：

核心算法：hash join

InfiniDB存储 – 按列拆

Employee_ID	*Job*	*Dept*	*City*
1	Shipping	Operations	Toronto
2	Receiving	Operations	Toronto
3	Accounting	Finance	Boston

数据按行进行存储

1	Shipping	Operations	Toronto
2	Receiving	Operations	Toronto
3	Accounting	Finance	Boston

数据按列进行存储

1	Shipping	Operations	Toronto
2	Receiving	Operations	Toronto
3	Accounting	Finance	Boston

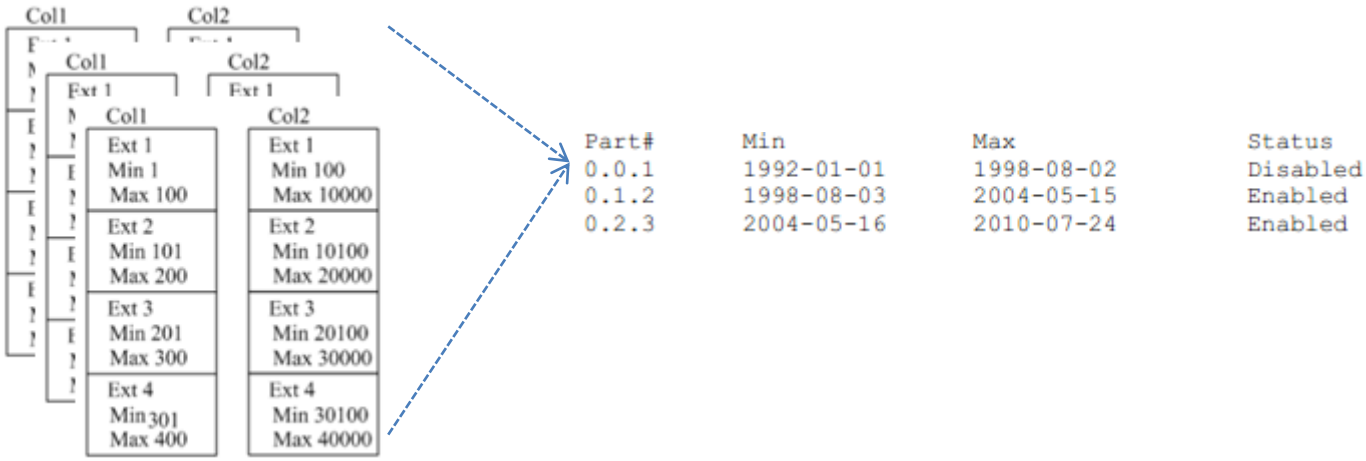
列存储适用于 ...

- 只有个别列需要涉及的海量应用
- 统计报告（sum、count、average etc）

益处：

- 更高的压缩比率
- 更快的数据分析效率
- 比行存储更小的CPU资源利用

InfiniDB存储 – 按行（范围）拆



每个范围（术语：Extent Map）都有最大/最小值，方便过滤

Extent Map 的向上扩展更大的范围（术语：Partition）也有最大值/最小值

每个Extent Map 可以并行计算

InfiniDB存储 – hash join

核心算法：hash join

- 每行都有一个rowid
- 查询2列以上：通过行rowid关联，使用hash join
- 不太怕表的关联
- 很怕Select *

Employee_id	job	Dept	City
1	shipping	Operations	Toronio
2	Receiving	Operations	Toronio
3	Accounting	Finance	Boston

rowid	Employee_id
a0001	1
a0002	2
a0003	3

rowid	job
a0001	shipping
a0002	Receiving
a0003	Accounting

rowid	Dept
a0001	Operations
a0002	Operations
a0003	Finance

rowid	city
a0001	Toronio
a0002	Toronio
a0003	Boston

InfiniDB存储 – 为啥查询这样快（总结）

数据存储方面，“拆拆拆”：

- 按列拆
- 按行（范围）拆：
- 通过核心算法：hash join实现关联

装载和更新-真实业务性能测试

load 29亿数据

1.测试条件
  load 29亿的业务数据
1.结论：
 load file的速度：18.8万/每秒，中间没有明显的性能毛刺，
 占用空间：infinidb大约是inforbright1.8倍 。压缩比是没有压缩的5倍。
          infinidb占用空间38g,ib数据量21g.
 其他情况：
    1个load loal file会被自动转换成3个cpimort命令。所以性能比较高
    从os看，远远没有达到io的瓶颈 ：每秒35m,tps 188. %util 1.7
    内存占用PrimProc(它缓存io数据块的部件）占用了13g,其他部件内存占用<100m

更新部分数据

30亿中更新1000w,　　更新速度148w/每秒
```
[SQL] update dm_vid_day set date=20140811 where date=20140810
Affected rows: 10943823
Time: 7.384
10943823/7.384
update:148w/每秒
```

30亿中删除1000w，删除速度6w/每秒
```
DELETE from dm_vid_day where date=20140811;
Affected rows: 10853236
Time: 169.408s
delete :6w/每秒
```

Insert速度（不建议使用）

模式	操作类型	执行记录数	总执行时间	每条记录执行时间	速率(记录数/s)
单行	Insert	1w	1h3m	0.38s	2.6/s
多行	Insert	1w	5.68s		2165.6/s
多行(hadoop)	Insert	1w	4.73s		2114.4/s

InfiniDB存储 – 数据装载

语法load data local infile …

- 速度超快 (>10万/每秒)
- 一个表只能对应一个load语句，不可并行

内部过程：

- 内部实现转换成cpimport的方式
- 内部实现 并行加载（不可以调并行度，代码写死了）
- Cpimport的实现是append 文件的方式

InfiniDB存储 –-锁、事务和mvcc

对于DML:页级别的锁

Version Buffer (SCN) :

1. 保存被修改的数据块，用于管理回滚、MVCC支持和snapshot
2. Initial 4M 内存hash表，默认文件1G，VersionBufferFileSize控制大小
3. 在HDFS上，MVCC是disabled的，回滚只支持在语句级

对应load数据：append数据到文件末尾，需要回滚时直接抛弃数据

InfiniDB – 压缩

每一列的重复值多，所以压缩率5倍

set infinidb_compression_type = n

可以在实例级或session启用关闭压缩。

0) 关闭压缩

1 or 2) 启用压缩，默认为2（quicklz算法）

InfiniDB – 免优化

- 无index
- 自动分区

- dba唯一可以做的：sql优化只能调整表的连接次序

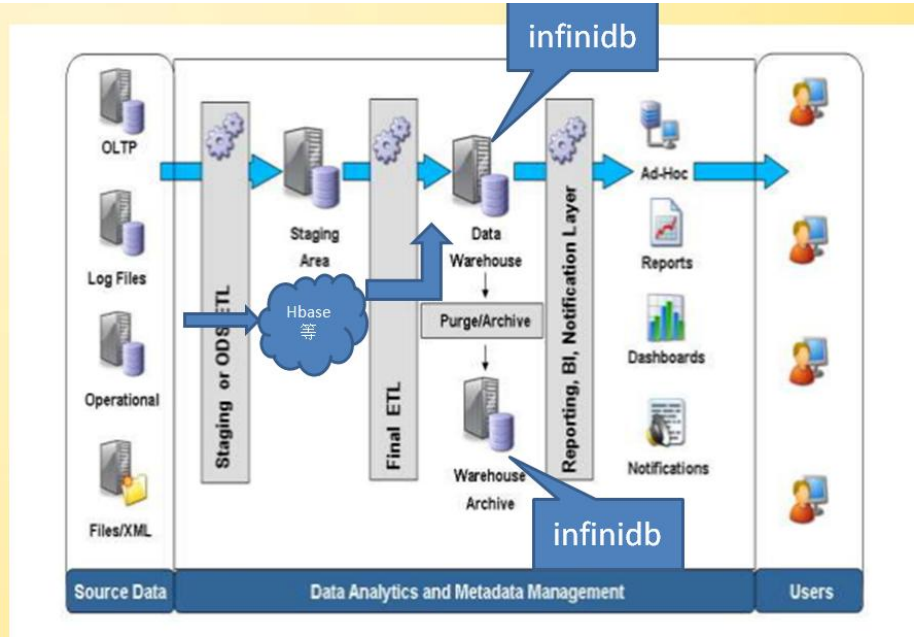
InfiniDB – 和其他产品的对比

Inforbright社区版

- 功能：不支持DML
- 限制功能的开源

Hbase

- Hbase本质上是个key –多value的构架
- 复杂
- 扩展性好
- 和infinidb是互补的结构



infinidb产品特点（总结）

产品特点：

- Mysql协议兼容
- 全功能，支持dml

- 统计类查询10倍
- Load数据快（每秒>10万）
- 压缩率5倍（和裸数据比）

- 免优化

目录

- 背景
- **InfiniDB的特点**
- **Infinidb的实战**

InfiniDB – 社区支持问题

现在支持比较差，未来前景比较好

- （现状）文档和问题资料比较少
- 2000 年公司，发布产品
- 2013.10月，支持hadoop文件系统
- 2014.10月公司倒闭
- 2014.10月 mariadb接手
- 2015.Q1会发布新的版本

InfiniDB – 高可用问题

集群版本没有高可用文档，但是也许不太稳定

单机版本本身不提供高可用

- 备份/恢复方案+手工补缺少数据
  使用lvm
- Drbd的方案
- Mysql主从方案不适应（ 因为有binlog问题）

InfiniDB – 高可用drbd

方案1：（infinidb高可用方案）

InfiniDB – 规范（合适的场景）

合适的场景

[编辑

1. 数据量大，通常单表都是>50亿的数据
1. 如果数据量超级大，比如200亿，请拆表
2. 查询里有统计分析的查询，比如包括sum,group by,order by和相应的多表关联
3. 数据通过批量加载导入（而不是单行导入），比如mysql的loadfile来加载数据就很好
    1. loadfile大约每秒能大于10万
4. 查询的响应时间在0.5秒到几分钟之间
    1. 即使你的查询很复杂，它的查询比较稳定（一般在1-100秒之间），不可能很快，但是也不会太慢。如果想很快，可以建汇总表，或者更近一步，把汇总表挪到innodb中去
5. 并发不能太高，复杂查询并行至少不能超过cpu的核数2倍，否则会把cpu/io耗尽
6. 数据中数值比较多，字符类型比较少，varchar不能超过8000
7. 支持事务
8. 支持dml语句，可以update/delete/insert等
9. 支持常用的统计函数和窗口函数（比如同比/环比）
10. 不太建议统计分析类的和在线类应用放在同一个实例中，虽然infinidb支持混用存储引擎
    1. 因为统计分析类的和在线类应用对性能和数据量本质要求是不同
    2. 另外因为运维上差异也比较大（比如对可用性，备份等的要求）
11. 和源数据相比，可以压缩大约10倍

InfiniDB –规范（不合适的场景）

不合适的场景

1. 查询包括大量单行查询，它的单行查询效率其实和统计类查询效率差不多
2. 大量存在select * 这种列出所有列的查询
3. 通过在线的单行insert录入数据，它的单行插入每秒只能是几十条
4. 需要特别快的查询响应时间，比如50ms内
    1. 这种一般通过把汇总表放到innodb解决
5. 有大的并发，比如200个并发
6. 大量的varchar甚至text的字段

InfiniDB-兼容 性

Operating Mode

有一个session级的变量，影响对SQL的支持：

set infinidb_vtable_mode = n

n的值：

- 0，与row-by-row处理模式高度兼容，有一些WHERE子句组件可以由infinidb处理，但join由mysqld处理，使用nested-loop机制
- 1，infinidb模式，性能更好，但infinidb不支持的语句会无法执行
- 2，自动切换（auto-switch）模式，会内部先尝试用infinidb模式执行，如果无法执行，则采用row-by-row模式执行

不支持最新版本mysql 官方的java驱动

支持marridb的mysql驱动

InfiniDB –应用类问题

- Infinidb数据会混乱
  - 使用最简单的语法
- Infinidb的数据表损坏
  - 重新建立表，然后把数据导回去
- 大量的delete/load并行容易死锁
  - 比如一天 84次delete,每次600万
- Infinidb数据量排序大报错
  - max_length_for_sort_data
- infindb server本地mysql客户端不能load file
  - 远程做，不要本地做

总结

- 目前：在可用性要求不高情况下，sql统计分析的利器

  统计类查询10倍

  Load数据快（每秒>10万）

  压缩率5倍（和裸数据比）

  免优化

- 将来：在数据仓库中环境中，很有前景的产品
  - 构架先进
  - 有mariadb组织的支持


ChinaUnix
THANKS