国台数据中心海量数据库设计

肖健、于策、王润涛， 天津大学
赵青， 天津科技大学

内容简介

- 基于现有的软硬件环境，现有的天文观测数据，现有的数据库优化技术
- 以天文科学家普遍应用的锥形检索和ID检索等数据访问协议为应用案例
- 分析国家天文台数据中心数据存储和数据服务的特点，综合考虑多方面因素
- 提出并比较了PostgreSQL和SciDB为基础的海量数据库设计与实现方案

提纲

- 工作背景介绍
- 基于传统数据库PostgreSQL的设计与实现
- 基于非传统数据库SciDB的设计与实现
- 小结

工作背景介绍

- 国家天文台中国天文数据中心（CASDC）
  - 天文数据 （CSTAR…）
  - 天文数据服务(锥形检索…)
  - 实际需求
- 国家天文台-天津大学天文信息技术联合实验室：
  - 天文计算、天文图像处理、嵌入式系统
    - 阶段成果：AST3实时相减测光、数据归档系统
  - 虚拟天文台、海量数据处理
    - 阶段成果：基于MapReduce的交叉证认
    - 在研课题：中国天文数据中心 数据服务

AST3 数据处理

- 阶段成果
  - AST3实时相减测光处理系统 需要在无人值守的条件下长时间持续运行，并且每2.4分钟需要完成200MB数据的处理
  - AST3实时星表数据归档系统 需要在无人值守的条件下长时间持续运行，并且每2分钟需要完成2.3万条星表记录的证认与归档处理
- 后续工作
  - 大规模离线数据处理
    - 多CPU+GPU并行解决方案

天文数据中心 数据服务

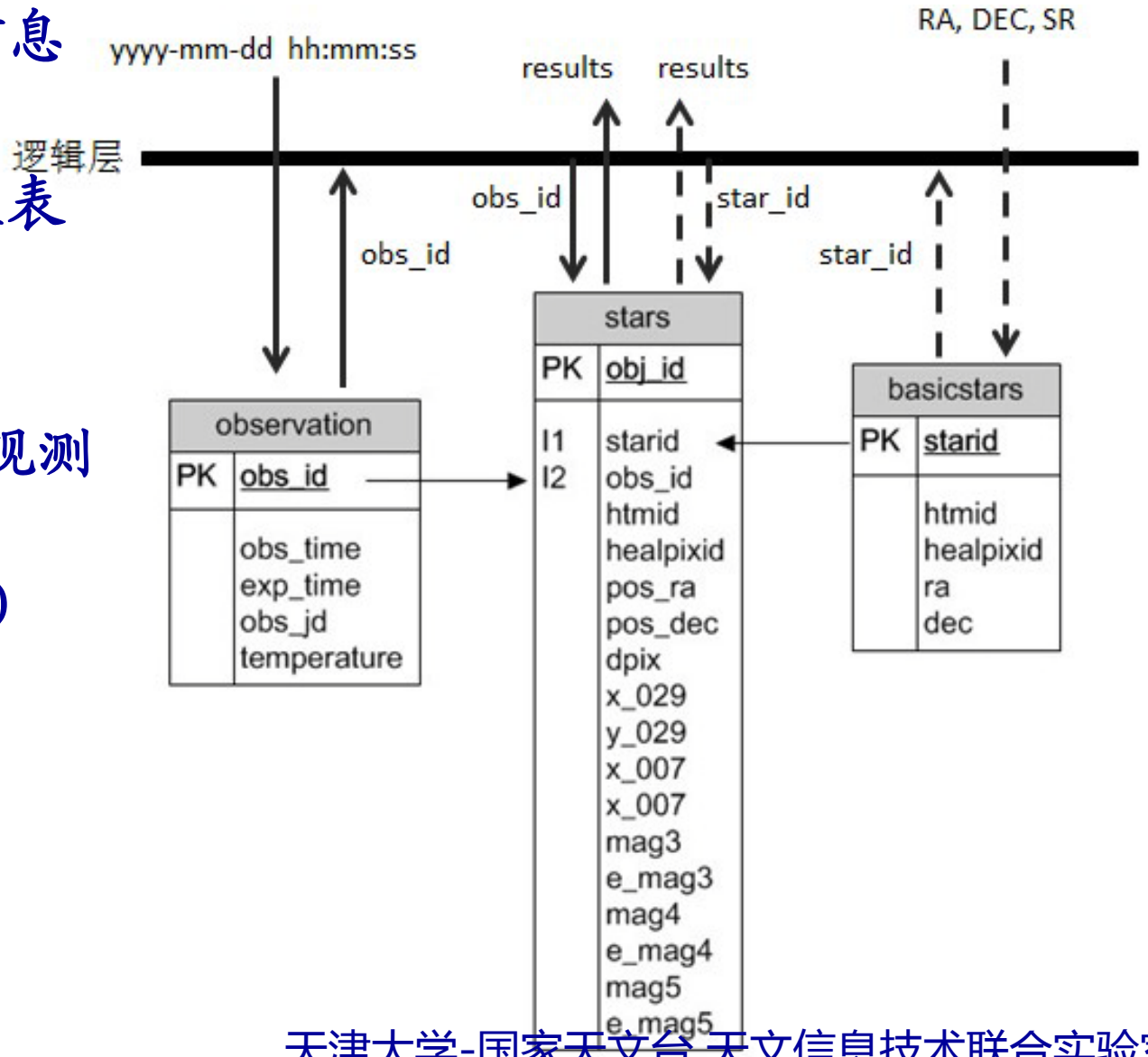
- 当前任务: CSTAR数据服务(22亿条)
- 功能
  - 按时间戳的检索
  - 锥形检索
  - ID检索
- 性能
  - ID检索<20秒
  - 锥形检索<40秒
  - 不间断运行...

基于PostgreSQL的设计与实现

observation观测信息

Basicstars 基准星表
（10000多颗）

Stars 所有22亿条观测记录
（交叉证认处理后）



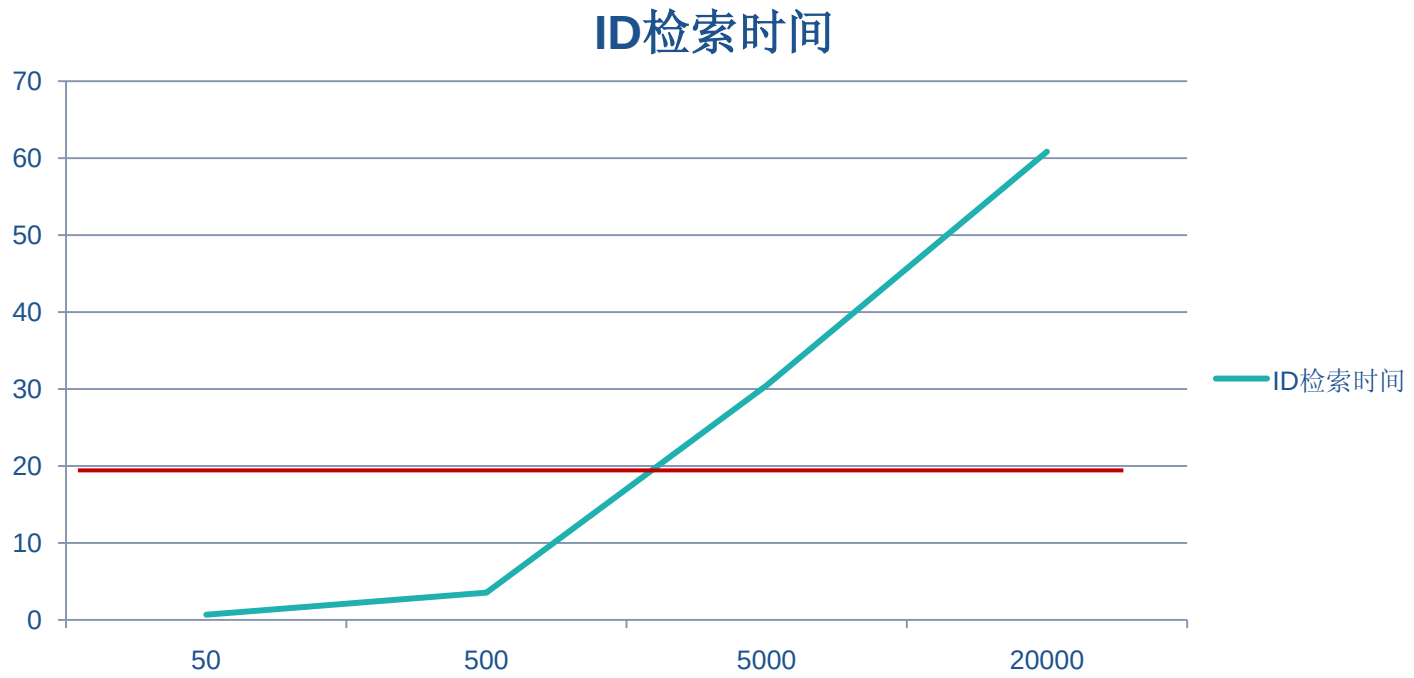
基于PostgreSQL的设计与实现

- 性能测试——ID检索
  - 22亿条数据中检索不同条数的时间

检索数据量	时间					
	Time1	Time2	Time3	Time4	Time5	Average Time
50	0.745710135	0.641433001	0.733093977	0.663695097	0.632187843	0.68322401
500	3.526250124	3.538226128	3.580218077	3.545917988	3.585434914	3.555209446
5000	30.13168693	29.52283216	31.38443685	30.73243213	30.67027092	30.48833179
50000	59.36318517	61.31051243	60.79106516	61.08328496	61.73404057	60.85641766

基于PostgreSQL的设计与实现

- 性能测试——ID检索
  - 22亿条数据中检索不同条数的时间



结论：检索性能基本呈线性增长，不满足需求

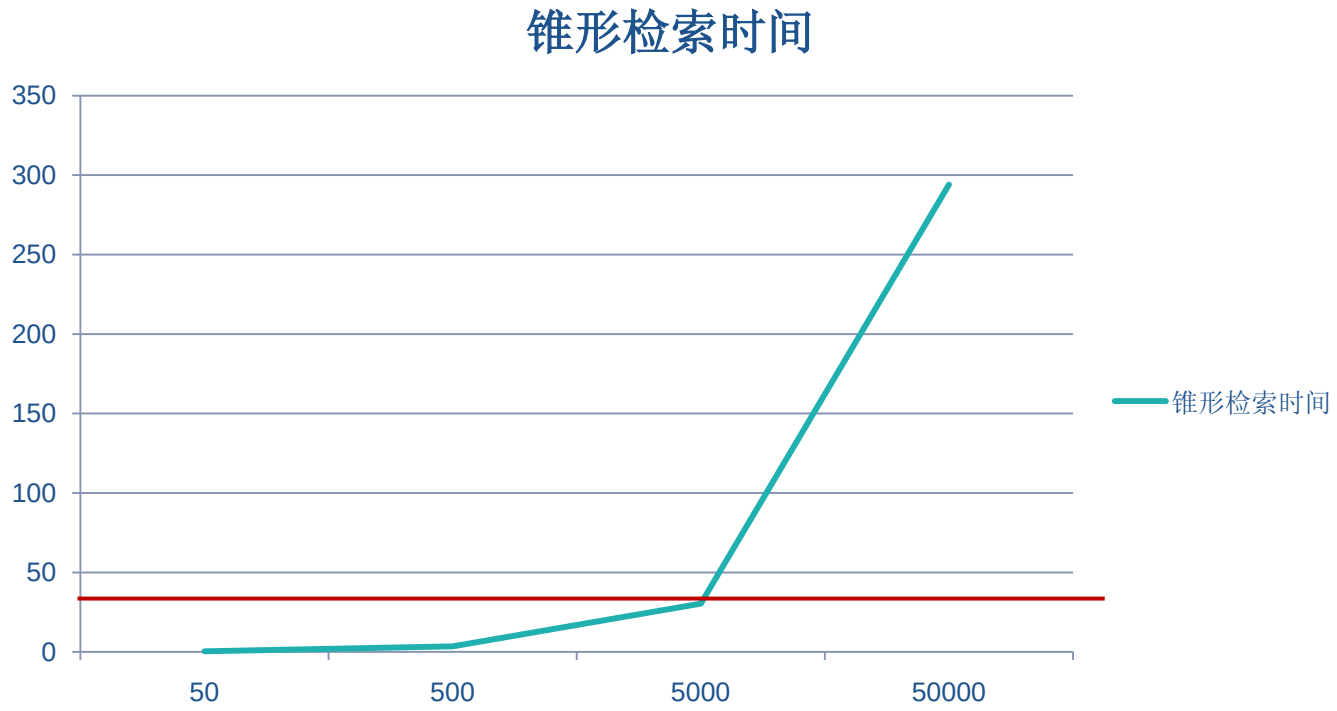
基于PostgreSQL的设计与实现

- 性能测试——锥形检索
  - 22亿条数据中检索不同条数的时间

检索数据量	时间					
	Time1	Time2	Time3	Time4	Time5	Average Time
50	0.642873959	0.538596825	0.630257802	0.560858922	0.529351668	0.580387835
500	3.423413949	3.435389952	3.477381901	3.443081812	3.482598738	3.452373271
5000	30.02885075	29.41999598	31.28160067	30.62959595	30.56743474	30.38549562
50000	290.6492464	284.7561411	302.7746129	296.4638592	295.8622009	294.1012121

基于PostgreSQL的设计与实现

- 性能测试——锥形检索
  - 22亿条数据中检索不同条数的时间



结论：检索性能基本呈线性增长，不满足需求，且经常出现无响应的情况。

基于SciDB的设计与实现

SciDB简介

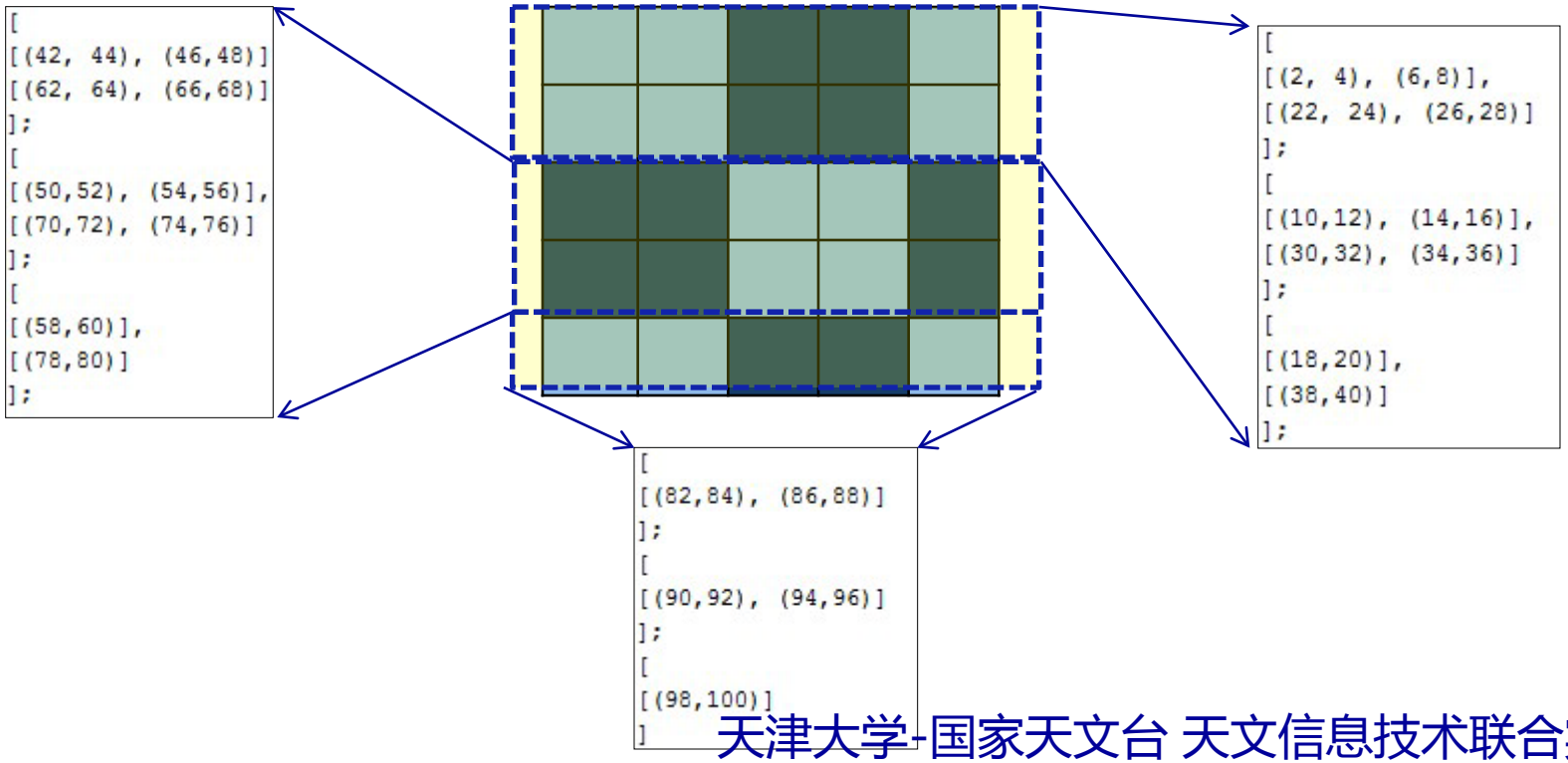
- 2008年由斯通布雷克与David DeWitt以及来自于布朗大学，MIT，威斯康星大学麦迪逊分校的研究人员创建
- 以科学研究为应用的开源数据管理和分析系统（DMAS）
- 围绕数据分析建立的，数据存储的特点是一次写入，多次读取
- 数据的呈现形式是多维矩阵，包括维度和数据单元的属性两部分

SciDB存储方式

- 多维矩阵
- 数据：以二进制形式存放于文件系统
- 元信息：保存在PostgreSQL数据库public模式中

多维矩阵特点

- 矩阵分块（chunk）⟶ 相关数据相邻
- 致密矩阵：
  - 每个单元都必须存放数据，数据的导入形式必须严格符合矩阵的结构
  - create array example1 <a:int64, b:int64>[x=1:5,2,0, y=1:5,2,0]



多维矩阵特点

- 稀疏矩阵：
  - 存在没有数据的单元，通过指定每个数据单元的坐标导入数据
  - create array example2 <a:int32> [y=0:3,2,0,x=0:8,2,0]

```
[[
{0,0} (1)
{1,0} (11)
{0,1} (2)
{1,1} (12)
 ]];
[[
{0,2} (3)
{1,2} (13)
{0,3} (4)
{1,3} (14)
 ]];
```

SciDB数据操作特点

- Data Definition Language (DDL)
  - 建立矩阵 create array
  - 导入数据 load()
  - 导出数据 save()
  - 根据已有矩阵构建新矩阵 build(), build_sparse()
  - 保存矩阵的变更 store() (与build()结合使用)
  - 添加新数据 input() (与store()结合使用)
  - 删除矩阵 remove()
  - 重命名矩阵 rename()
  - 更改维度名，属性名 cast()
  - 查看元信息 list(), dimensions(), attributes(), versions()

SciDB数据操作特点

- Data Manipulation Language (DML)
  - 查看所有数据 scan()
  - 根据维度值提取子集 subsample()
  - 选择指定区域的数据 between()
    - 与subsample的区别：结果中不会改变元素的位置
  - 根据属性值选择子集 filter()
  - 根据模式选择矩阵 lookup()
  - 提取部分矩阵的部分属性构建新矩阵 project()
  - 联合两个矩阵 join()
  - 更新属性值 apply()
  - 统计操作 aggregate(), avg(), count(), max(), min(), sum()
  - 添加/删除维度 adddim(), deldim()
  - …

基于SciDB的设计与实现

- 基于SciDB的原因
  - 分布式环境
  - 专门针对海量科学数据
  - 数据一次写入、多次读取
- 基于SciDB的设计方案
  - 检索参数作为维度
  - 大矩阵分块
  - 不同矩阵应对不同检索

具体设计方案示例

- 最佳检索方式：以维度值为检索参数
- 检索函数：subsample()

create array cstarTest<…>[oid=1204300800:1217520000,1,0, id=0:20000,500,0]

subsample(cstarTest, 1212554381 , 0, 1212554381, 20000)

create array starTest<…>
[RA=0:35999999,144000,8400, DECL=0:17999999,72000,4200]
（ 将天空划分为250*250个块，每个块大约是1/2弧度（arc-degree））

subsample (starTest, (126.734*10000), (-43.3245 * 10000),
(126.956*10000), (-41.9483*10000));

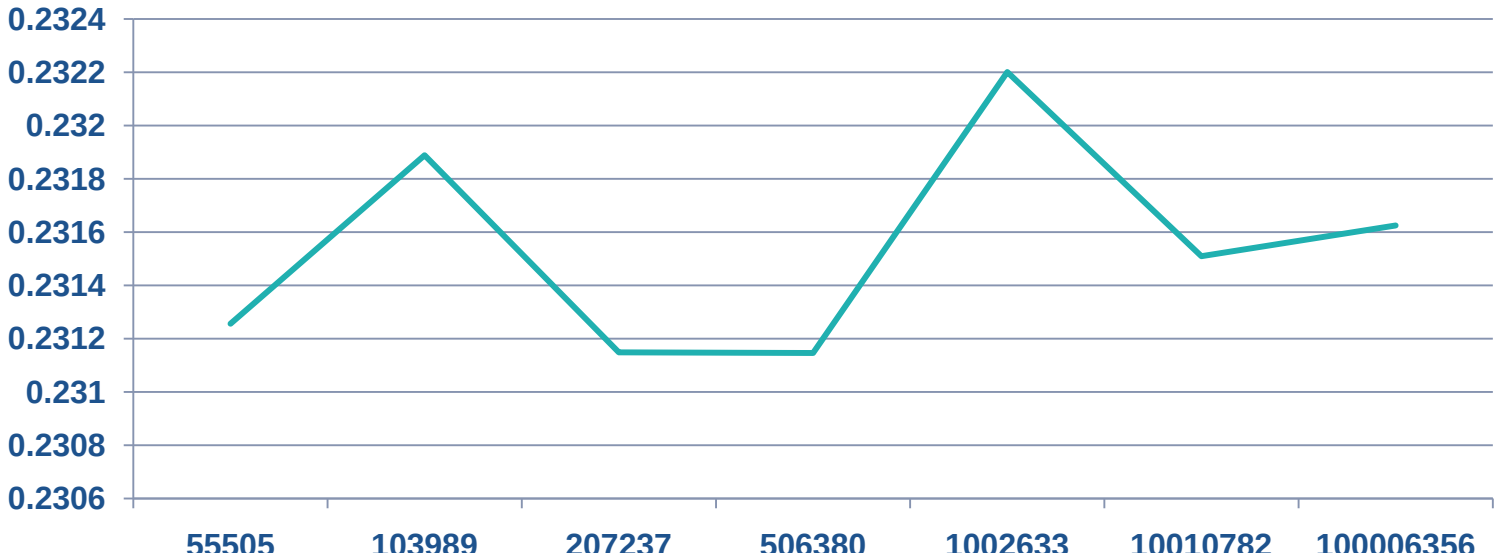
基于SciDB的设计与实现

- 性能测试——ID检索
  - 导入的数据量不同时，检索50条数据的性能测试

存储	时间					
数据量	Time1	Time2	Time3	Time4	Time5	Time6
55505	0.234064102	0.2336092	0.233534098	0.232512951	0.232944965	0.233333063
103989	0.233119011	0.232089043	0.231813908	0.231522083	0.230900049	0.231888819
207237	0.233609915	0.231089115	0.230577946	0.229914904	0.230551958	0.231148767
506380	0.236222029	0.22903204	0.229988098	0.23020196	0.230288029	0.231146431
1002633	0.233501174	0.234106235	0.233193202	0.228129209	0.233073235	0.232400611
10010782	0.23216328	0.231248025	0.229200107	0.231920174	0.233015512	0.231509419
100006356	0.23021583	0.235206218	0.231563025	0.227116021	0.234023137	0.231624846

基于SciDB的设计与实现

- 性能测试——ID检索
  - 导入的数据量不同时，检索50条数据的性能测试



结论：检索性能与数据基数无关，检索时间基本为常量（0.23s)

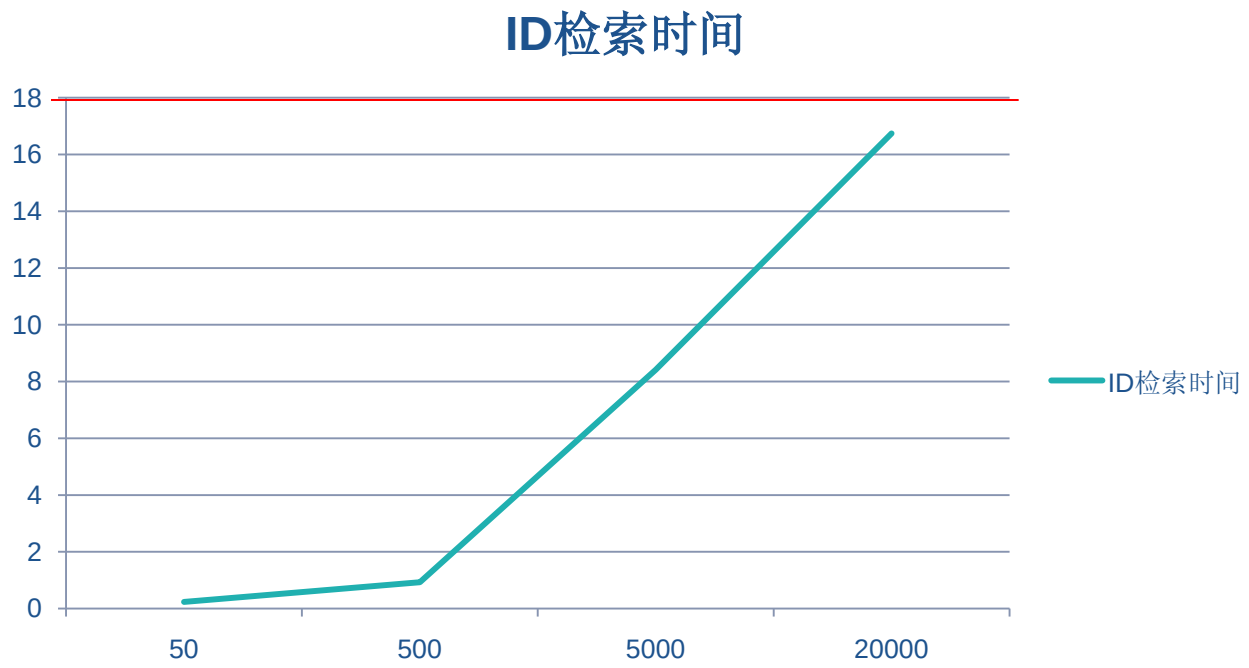
基于SciDB的设计与实现

- 性能测试——ID检索
  - 22亿条数据中检索不同条数的时间

检索数据量	时间					
	Time1	Time2	Time3	Time4	Time5	Average Time
50	0.234064102	0.2336092	0.233534098	0.232512951	0.232944965	0.233333063
500	0.882818937	0.861493111	0.954677105	0.995584965	0.943105936	0.927536011
5000	8.736562014	8.380357981	8.457309008	8.084507942	8.429337025	8.417614794
20000	17.23726678	16.83882284	16.33315682	16.82430792	16.46629405	16.73996968

基于SciDB的设计与实现

- 性能测试——ID检索
  - 22亿条数据中检索不同条数的时间



结论：检索性能基本呈线性增长，满足需求(<20s)

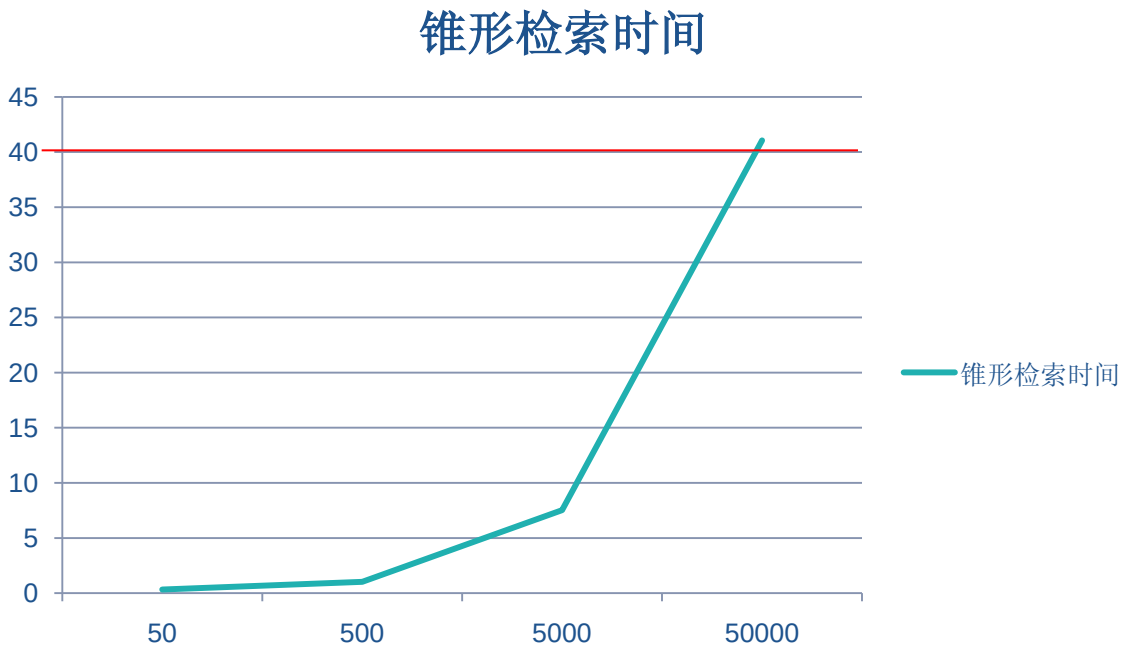
基于SciDB的设计与实现

- 性能测试——锥形检索
  - 22亿条数据中检索不同条数的时间

检索数据量	时间					
	Time1	Time2	Time3	Time4	Time5	Average Time
50	0.336900278	0.336445375	0.336370273	0.335349126	0.335781141	0.336169239
500	0.985655113	0.964329286	1.05751328	1.09842114	1.045942111	1.030372186
5000	7.839398189	7.483194156	7.560145183	7.187344118	7.5321732	7.520450969
50000	42.80311411	40.85824009	41.2783927	39.24289888	41.12566567	41.06166229

基于SciDB的设计与实现

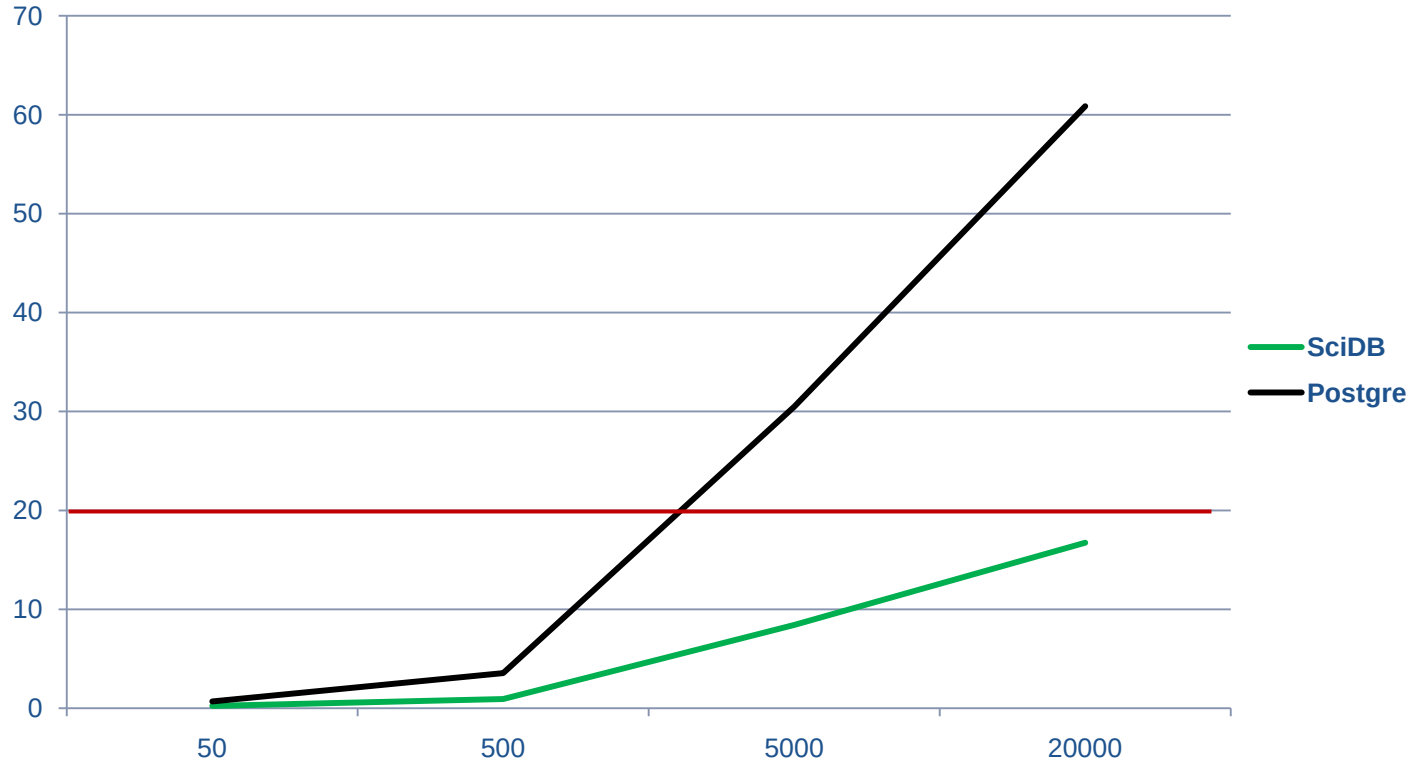
- 性能测试——锥形检索
  - 22亿条数据中检索不同条数的时间



结论：检索性能基本呈线性增长，满足需求（ <40s)

两种方案性能对比

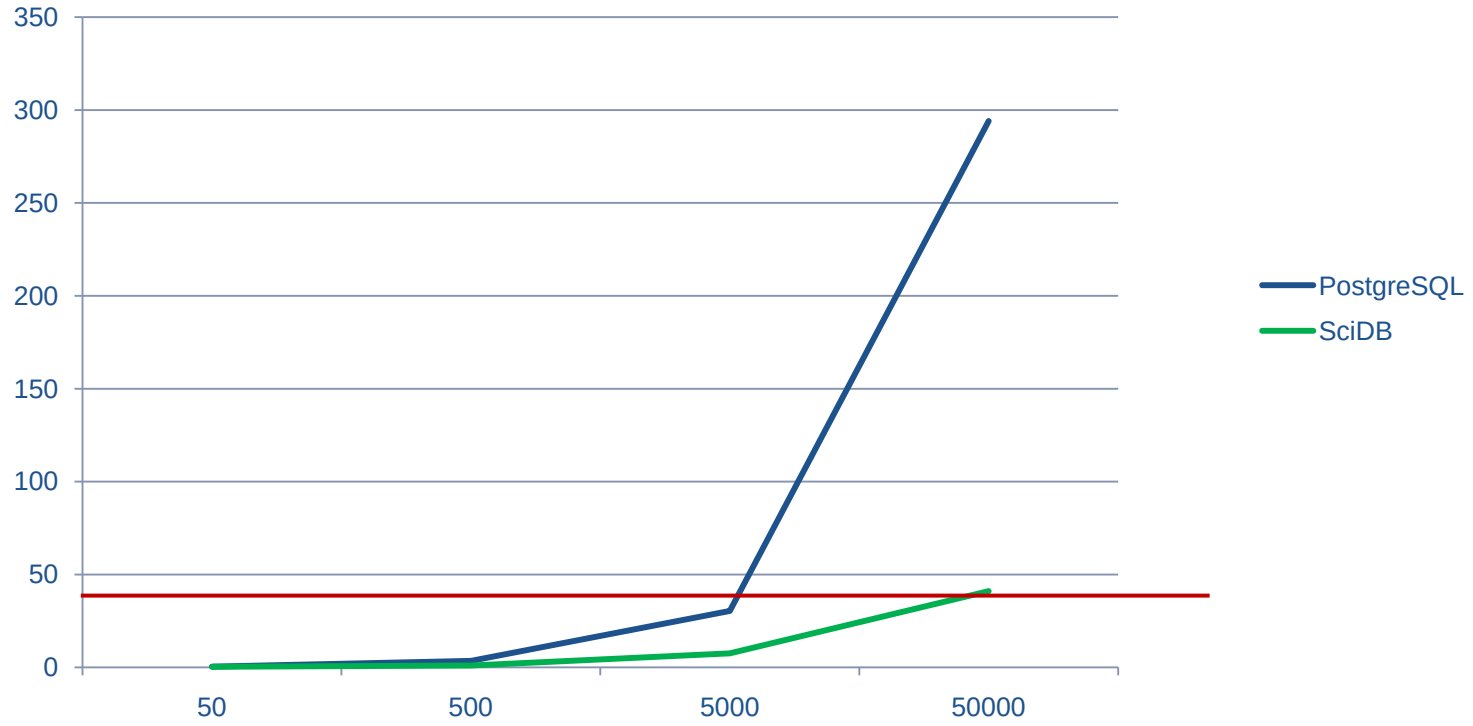
- 性能对比——ID检索
  - 22亿条数据中检索不同条数的时间



基本上均呈线性增长，增长速度SciDB<<PostgreSQL

两种方案性能对比

- 性能测试——锥形检索
  - 22亿条数据中检索不同条数的时间



基本上均呈线性增长，增长速度SciDB<<PostgreSQL

小结：两种方案优缺点

- SciDB优缺点
  - 优点：
    - 检索性能与数据量基本无关
  - 缺点：
    - 数据格式化过程较复杂
    - 不支持过于复杂的检索条件
- SciDB适用于数据量较大，查询条件较为简单的应用；
- PostgreSQL适用于数据量较小，查询条件较为复杂的应用。

小结：天文数据库建议方案

- PostgreSQL 与 SciDB 相结合
  - 数据量小，查询语句复杂 → PostgreSQL
    - 存放观测信息的表，查找与用户输入时间最接近的时间
    - 基准天体信息表，提供锥形检索，再根据天体 Id 检索SciDB
  - 数据量大，查询语句简单 → SciDB
    - 经过交叉证认后的原始观测数据
    - 以不同维度建立矩阵，满足不同需求
    - 按维度值分段，建立多个矩阵，平衡导入性能和查询性能

谢谢大家

Question & Answer