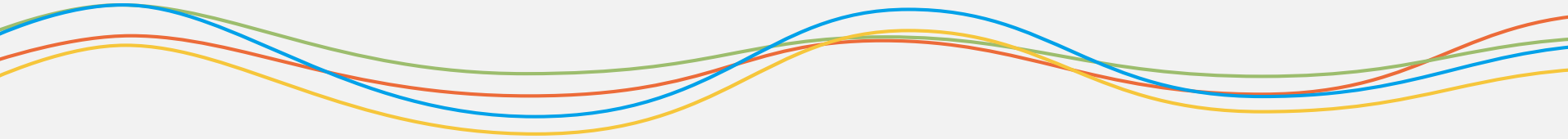
APMCon

Greenplum开源数据仓库

实现100亿监控数据的秒级分析

萧少聪(铁庵) 2016年8月19日



Greenplum开源数据仓库 – APM监控行业应用

APMCon

- [Greenplum重返开源的10个月]
- 100亿级监控数据秒级分析
- PostGIS结合地理信息监控数据
- 数学函数及MADlib实现SQL复杂分析
- 结合OSS云存储扩展海量级数据

Greenplum重返开源的10个月

APMCon

- 2005 Bizgres基于PostgreSQL结合BI特性的开源数据库

- 2005 推出Greenplum商业版本的MPP分布式数据仓库

- 2010 被EMC收购

- 2013 成为EMC旗下Pivotal公司核心产品

- 2015年10月 正式重回开源，基于Apache协议

Greenplum重返开源的10个月

APMCon

- 在GitHub中Fork出432个新版本
- 共有28607次commit
- 吸引了全球90位contributor代码贡献者
- 已经解决的request 820个，解决中的问题33个
- 在阿里云于2016年7月11日正式对外公测 [云数据库Greenplum版]
- 以上数据截止至2016年8月14日

Greenplum开源数据仓库 – APM监控行业应用

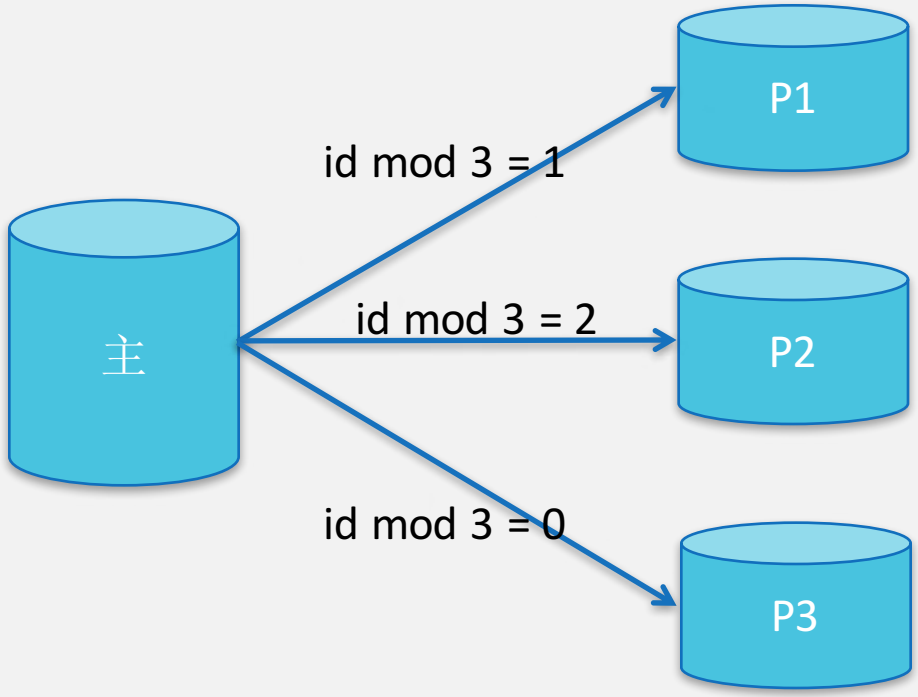
APMCon

- Greenplum重返开源的10个月
- **[100亿级监控数据秒级分析]**
- PostGIS结合地理信息监控数据
- 数学函数及MADlib实现SQL复杂分析
- 结合OSS云存储扩展海量级数据

- 1年有525600分钟，如果每台设备有25个要监控的指标

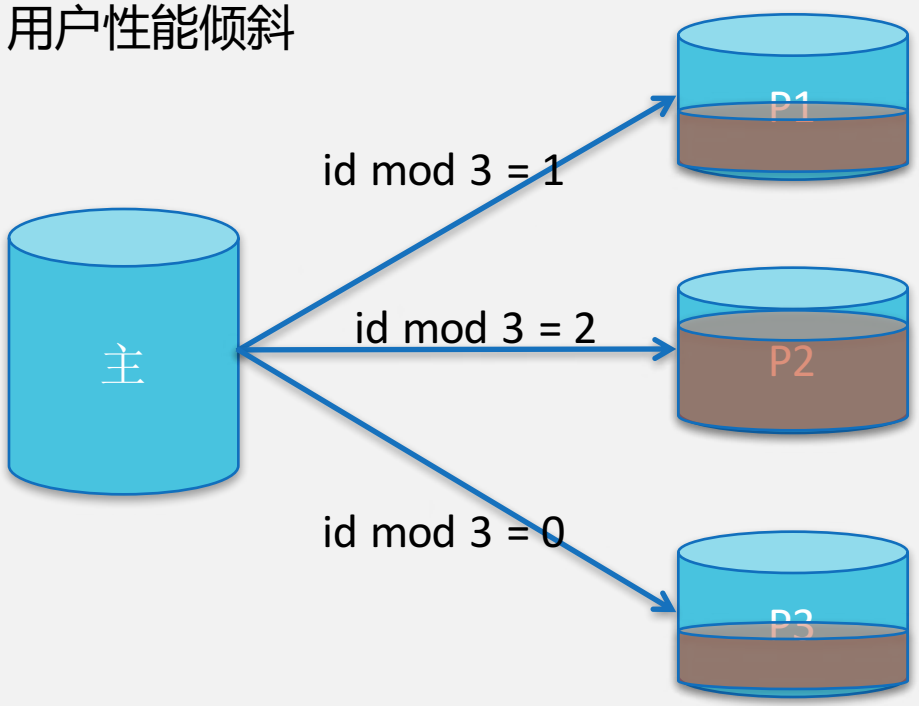
- 100亿 / 525600 / 25，约761台设备每分钟采样

分布式分库分表

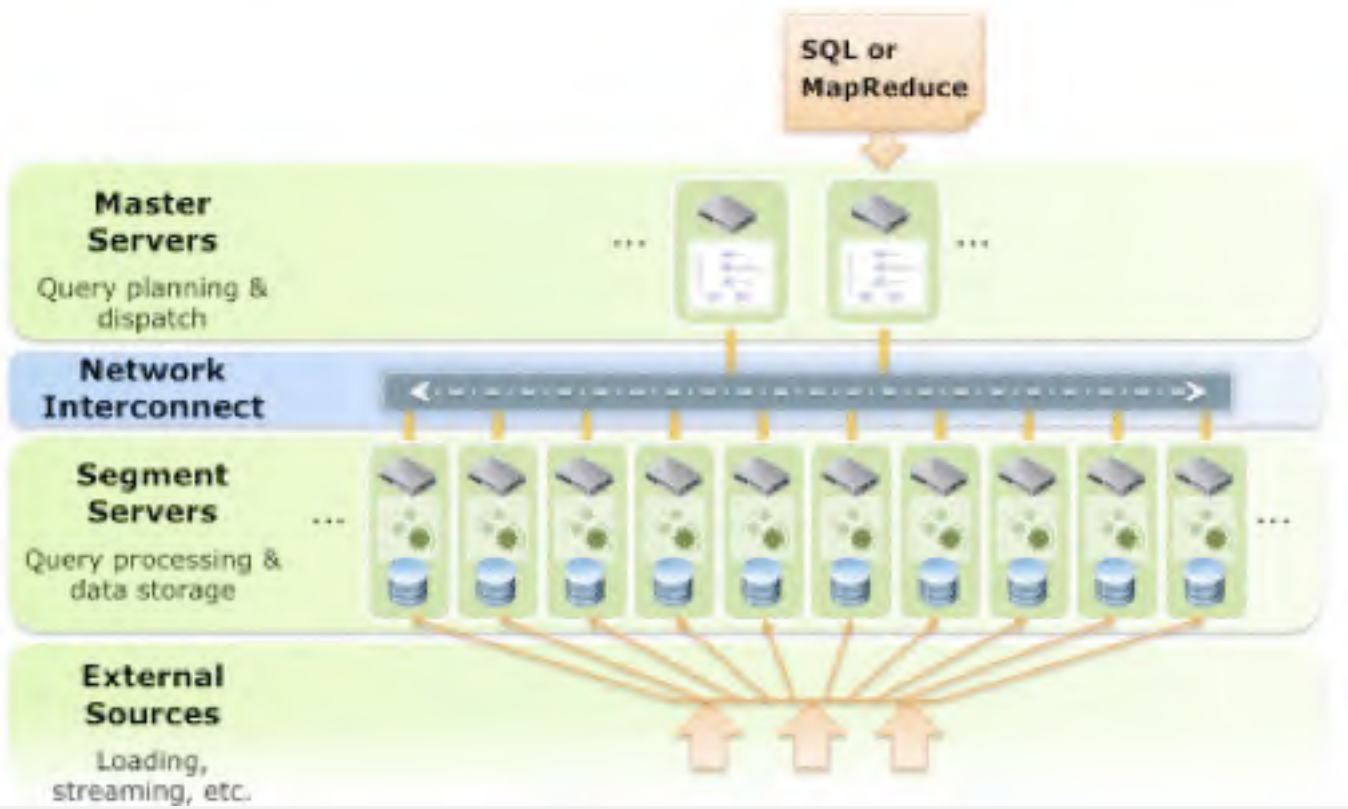


- 数据倾斜：
  - 每个设备活跃度不同，如关机、无信号
  - 不同用户查询频率不同

- 导致问题：服务器压力倾斜 / 用户性能倾斜



- Greenplum中的分片处理



图片来源：http://www.cubrid.org/wp-content/uploads/2011/2-/greenplum-system-configuration.png

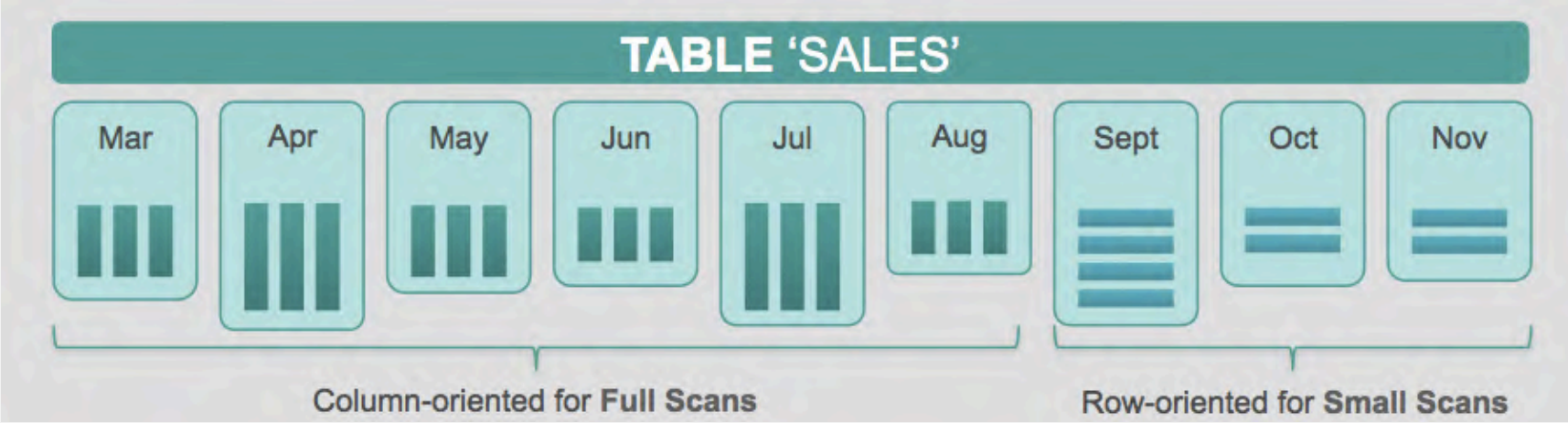
- Greenplum中的CREATE TABLE语法

```
CREATE [[GLOBAL | LOCAL] {TEMPORARY | TEMP}] TABLE
table_name (
[ { column_name data_type [ DEFAULT default_expr ]
   [column_constraint [ ... ]
[ ENCODING ( storage_directive [,...] ) ]
]
   | table_constraint
   | LIKE other_table [{INCLUDING | EXCLUDING}
                      {DEFAULTS | CONSTRAINTS}] ...}
   [, ... ] ]
   )
   [ INHERITS ( parent_table [, ... ] ) ]
   [ WITH ( storage_parameter=value [, ... ] )
   [ ON COMMIT {PRESERVE ROWS | DELETE ROWS | DROP} ]
   [ TABLESPACE tablespace ]
   [ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED
RANDOMLY ]
```

```
[ PARTITION BY partition_type (column)
      [ SUBPARTITION BY partition_type (column) ]
        [ SUBPARTITION TEMPLATE ( template_spec ) ]
      [...]
   ( partition_spec )
      | [ SUBPARTITION BY partition_type (column) ]
        [...]
   ( partition_spec
     [ ( subpartition_spec
         [(...)]
       ) ]
   )
```

让分析计算时，每台服务器计算量趋于平衡

- Greenplum中的行列混存储合支持



图片来源：http://blog.pivotal.io/wp-content/uploads/2014/10/Polymorphic.png

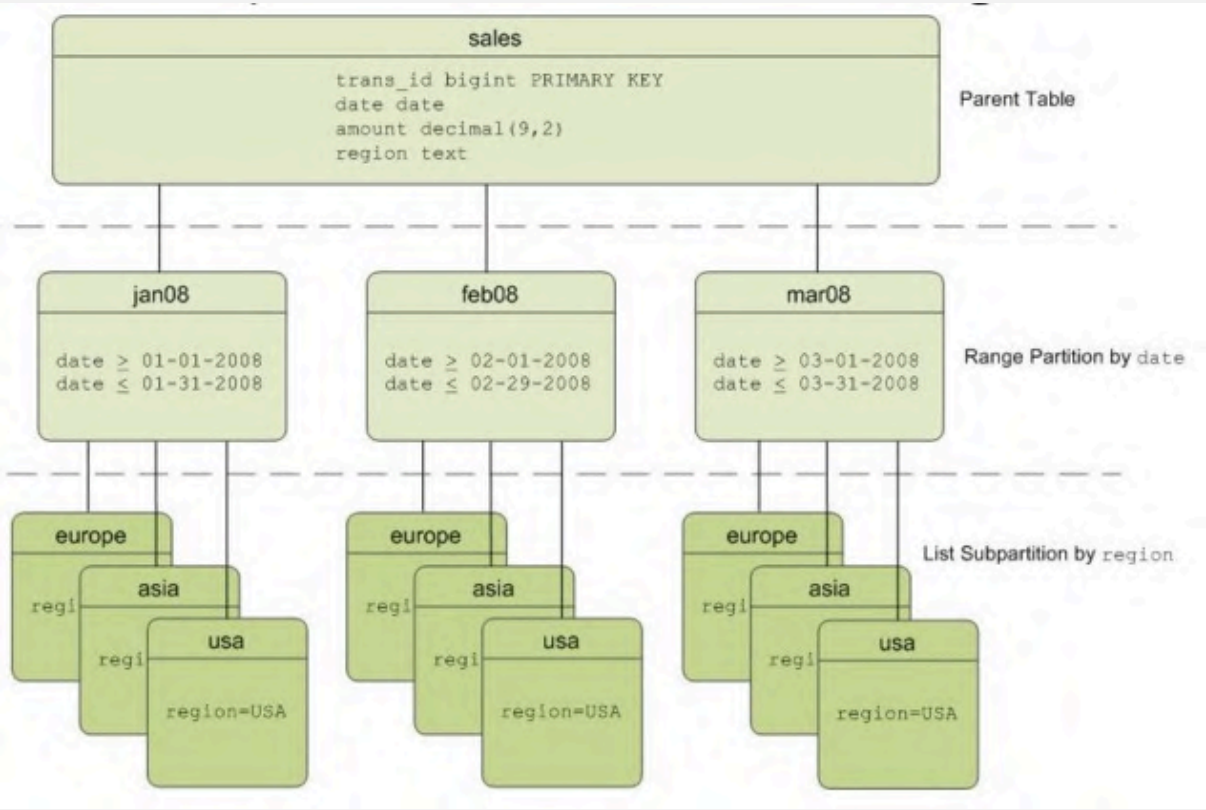
- Greenplum中的CREATE TABLE语法

```
CREATE [[GLOBAL | LOCAL] {TEMPORARY | TEMP}] TABLE table_name (
[ { column_name data_type [ DEFAULT default_expr ]
   [column_constraint [ ... ]
[ ENCODING ( storage_directive [,...] ) ]
]
   | table_constraint
   | LIKE other_table [{INCLUDING | EXCLUDING}
                      {DEFAULTS | CONSTRAINTS}] ...}
   [, ... ] ]
   )
   [ INHERITS ( parent_table [, ... ] ) ]
   [ WITH ( storage_parameter=value [, ... ] )
   [ ON COMMIT {PRESERVE ROWS | DELETE ROWS | DROP} ]
   [ TABLESPACE tablespace ]
   [ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED RANDOMLY ]
```

where *storage_parameter* for a partition is:

```
APPENDONLY={TRUE|FALSE}
BLOCKSIZE={8192-2097152}
ORIENTATION={COLUMN|ROW}
CHECKSUM={TRUE|FALSE}
COMPRESSTYPE={ZLIB|QUICKLZ|RLE_TYPE|NONE}
COMPRESSLEVEL={1-9}
FILLFACTOR={10-100}
OIDS[=TRUE|FALSE]
```

- Greenplum中的表分区
  - 针对Where条件查询
  - 按条件减少查询范围
  - 降低磁盘IO提高性能



图片来源：http://greenplum.org/gpdb-sandbox-tutorials/

- Greenplum中的CREATE TABLE语法

```
CREATE [[GLOBAL | LOCAL] {TEMPORARY | TEMP}]
TABLE table_name (
[ { column_name  data_type [ DEFAULT default_expr ]
   [column_constraint  [ ... ]
[ ENCODING ( storage_directive [,...] ) ]
]
   | table_constraint
   | LIKE other_table [{INCLUDING | EXCLUDING}
                      {DEFAULTS | CONSTRAINTS}] ...}
   [, ... ] ]
   )
   [ INHERITS ( parent_table [, ... ] ) ]
   [ WITH ( storage_parameter=value [, ... ] )
   [ ON COMMIT {PRESERVE ROWS | DELETE ROWS |
DROP} ]
   [ TABLESPACE tablespace ]
   [ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED
RANDOMLY ]
```

```
[ PARTITION BY partition_type (column)
      [ SUBPARTITION BY partition_type (column) ]
        [ SUBPARTITION TEMPLATE ( template_spec ) ]
      [...]
   ( partition_spec )
      | [ SUBPARTITION BY partition_type (column) ]
        [...]
   ( partition_spec
     [ ( subpartition_spec
         [(...)]
       ) ]
   )
```

Greenplum开源数据仓库 – APM监控行业应用

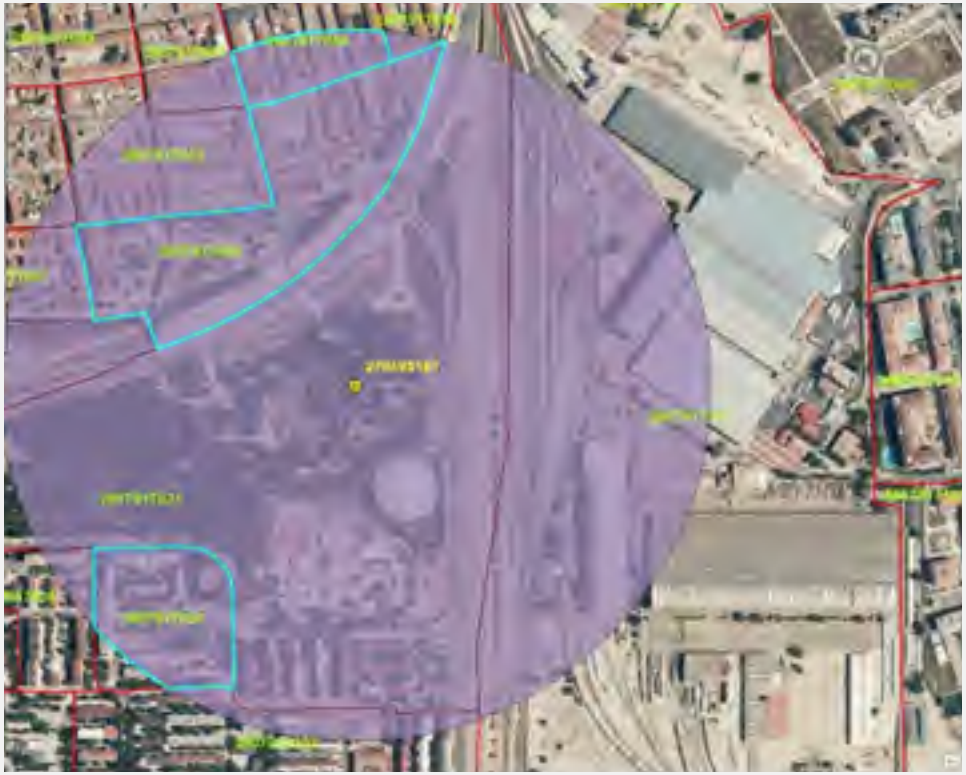
APMCon

- Greenplum重返开源的10个月
- 100亿级监控数据秒级分析
- **[PostGIS结合地理信息监控数据]**
- 数学函数及MADlib实现SQL复杂分析
- 结合OSS云存储扩展海量级数据

PostGIS结合地理信息监控数据

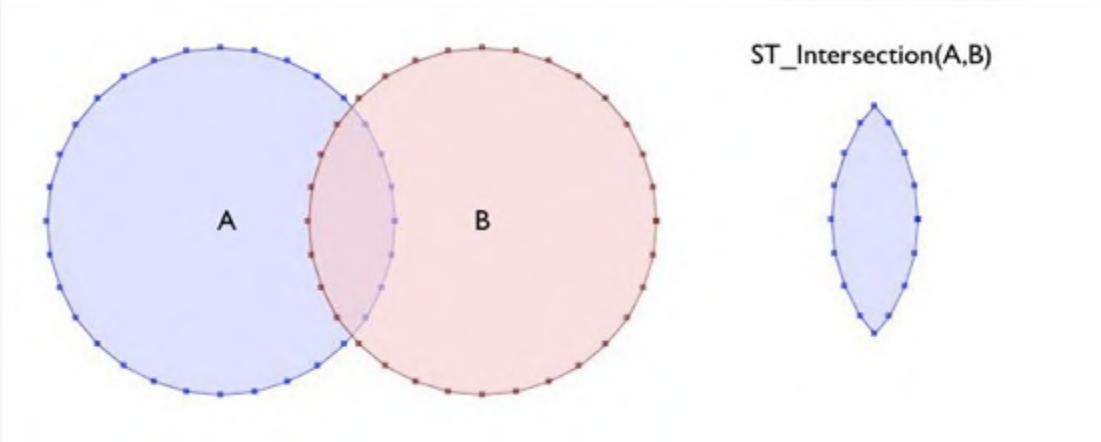
APMCon

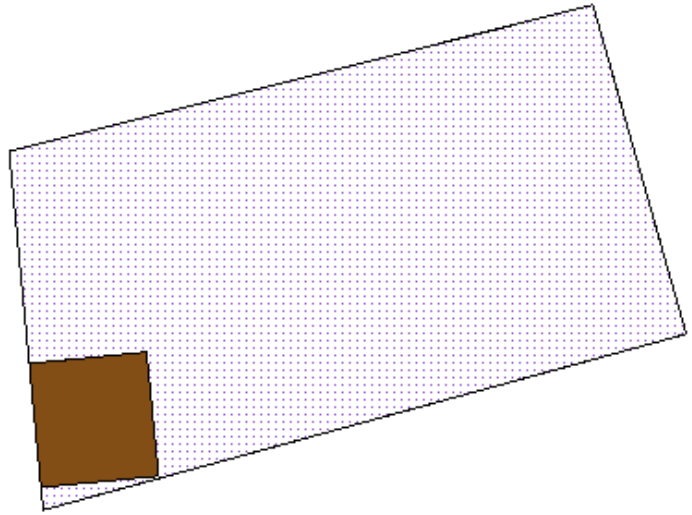
- 用户在不同区域（公园、商场、地铁）的App使用频率
- 各个区域用户的App组合使用习惯
- 区域化精准用户App推荐



图片来源：http://geospatialtraininges.com/2015/04/24/postgis-o-arcgis-comparando-rendimientos/

PostGIS结合地理信息监控数据





```
SELECT ST_AsText(ST_Intersection(
  ST_Buffer('POINT(0 0)', 2),
  ST_Buffer('POINT(3 0)', 2)
));
```

```
SELECT b.the_geom As bgeom,
p.the_geom As pgeom,
       ST_Intersection(b.the_geom,
p.the_geom) As intersect_bp
    FROM buildings b INNER JOIN
parcels p ON ST_Intersection(b,p)
    WHERE
ST_Overlaps(b.the_geom,
p.the_geom)
    LIMIT 1;
```

图片来源：http://gis.stackexchange.com/questions/25797/select-bounding-box-using-postgis

Greenplum开源数据仓库 – APM监控行业应用

APMCon

- Greenplum重返开源的10个月
- 100亿级监控数据秒级分析
- PostGIS结合地理信息监控数据
- **[数学函数及MADlib实现SQL复杂分析]**
- 结合OSS云存储扩展海量级数据

- 求方差（一），Variance

求总体方差：

```
postgres=# select var_pop(c1) from (values(1),(2),(3),(4),(5)) as t(c1);
      var_pop
--------------------
 2.0000000000000000
(1 row)
postgres=# select var_pop(c1) from (values(1),(2),(3),(4),(5),(1000)) as t(c1);
       var_pop
---------------------
 138058.472222222222
(1 row)
```

- 求方差（二），Variance

求样本方差：

```
postgres=# select var_samp(c1) from (values(1),(2),(3),(4),(5)) as t(c1);
      var_samp
--------------------
 2.5000000000000000
(1 row)
postgres=# select var_samp(c1) from (values(1),(2),(3),(4),(5),(1000)) as t(c1);
      var_samp
---------------------
 165670.166666666667
```

- 相关性, 线性相关性, Correlation

表示两组数据的相关性, 相关值从0到1取值

趋向1表示完全相关, 趋向0 表示完全不相关

```
postgres=# select corr(c1,c2) from (values(1,2),(2,3),(3,4),(4,5),(5,6),(1000,1001))  as t(c1,c2);
 corr
------
    1
(1 row)
postgres=# select corr(c1,c2) from (values(1,2),(2,3),(3,4),(4,5),(5,6),(1000,1))  as t(c1,c2);
        corr
--------------------
 -0.652023240836194
(1 row)
```

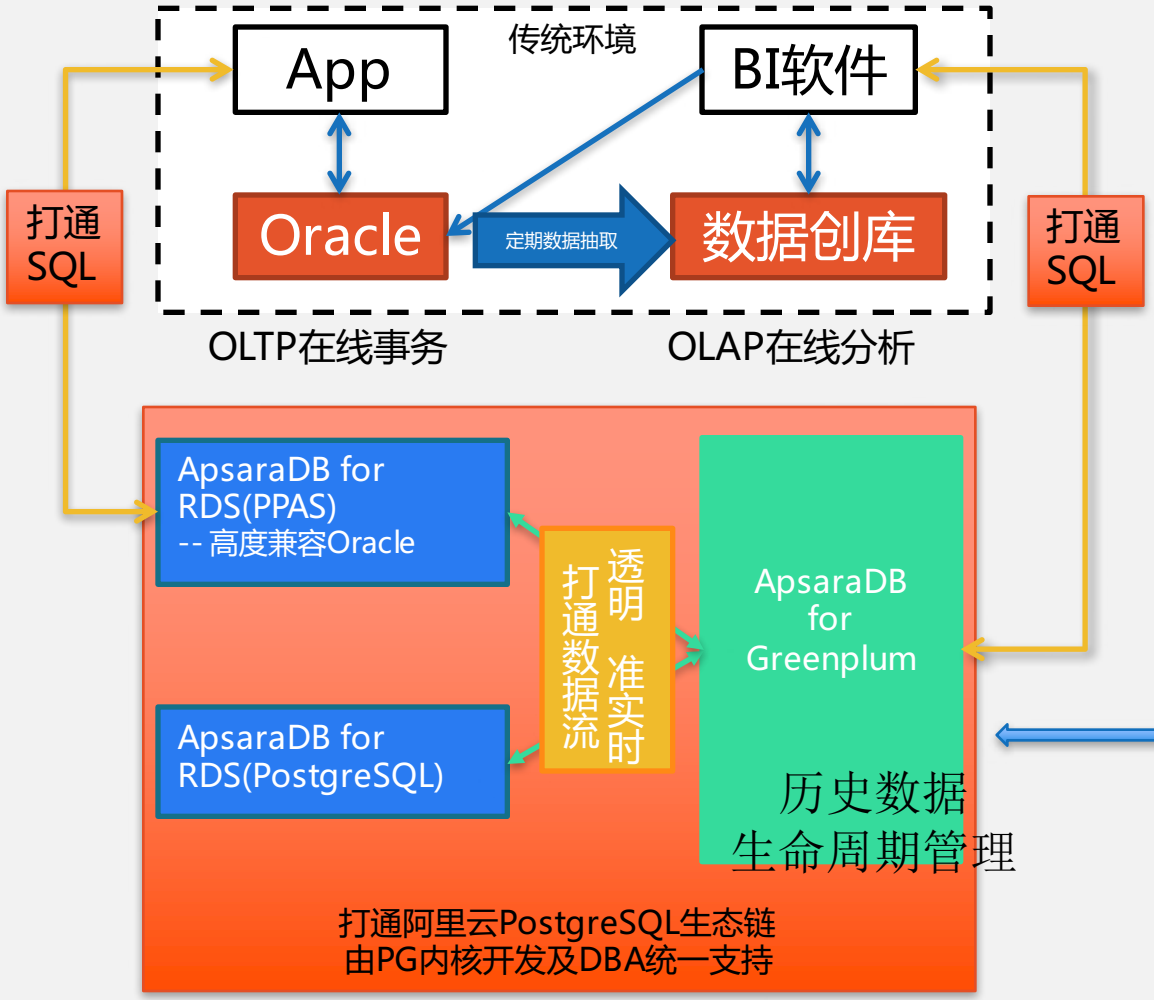
Greenplum开源数据仓库 – APM监控行业应用

APMCon

- Greenplum重返开源的10个月
- 100亿级监控数据秒级分析
- PostGIS结合地理信息监控数据
- 数学函数及MADlib实现SQL复杂分析
- **[结合OSS云存储扩展海量数据]**

在阿里云中Greenplum的扩展

APMCon



- 传统商业数据库上云最佳路径
- 最大程度SQL语法避免重新学习
- PB级历史数据按需随时可查询

THANK YOU