Greenplum 5：内核增强、广泛适用的大数据平台

Greenplum产品经理　高小明

1 内核增强

2 跨云部署

3 高级分析与机器学习

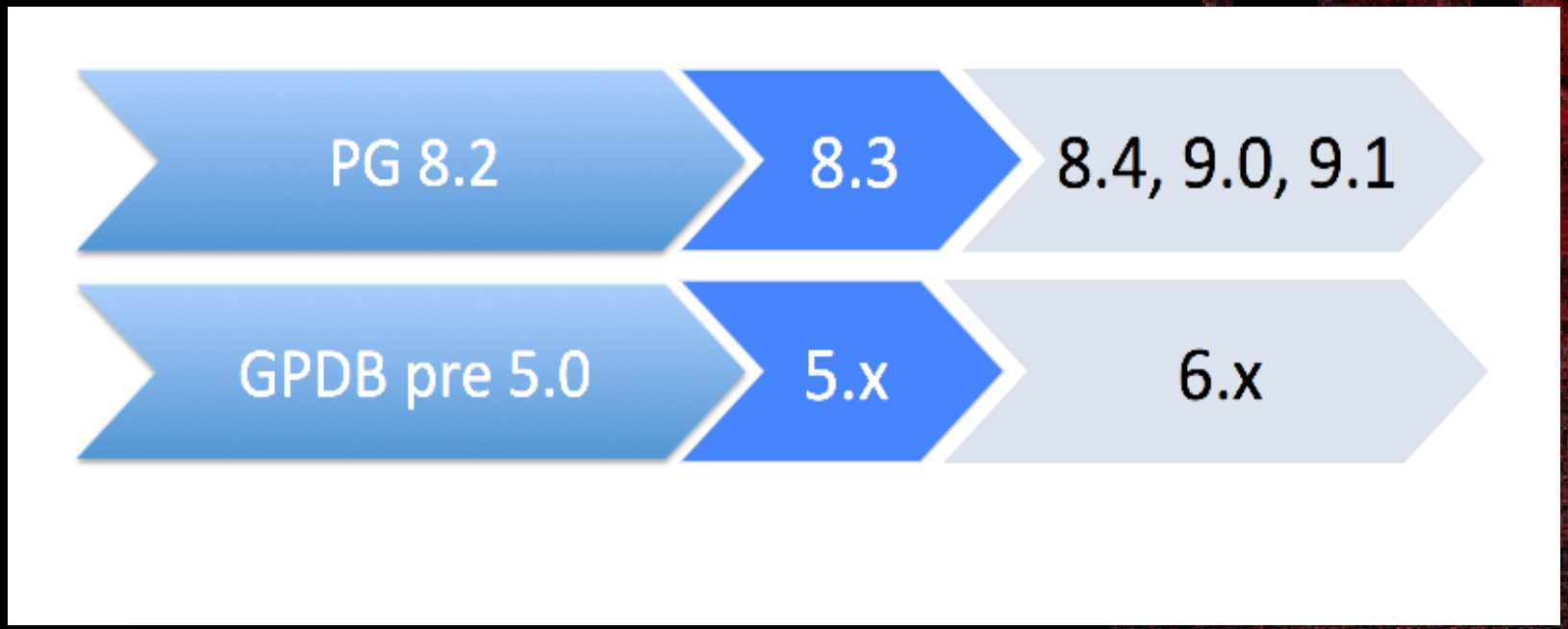
4 OLAP与OLTP混合负载

5 广泛适用

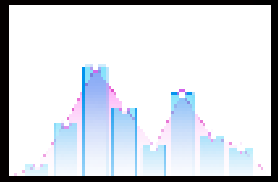
内核增强

持续集成PostgreSQL和Greenplum两大社区的创新

PostgreSQL合并

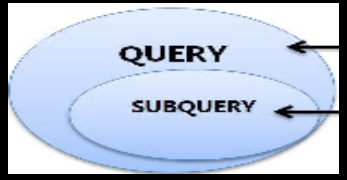


内核新特性



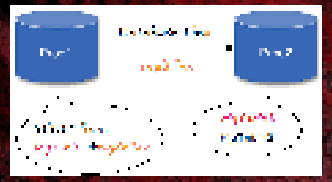
Analyze
加速

ORCA
增强



子查询
优化

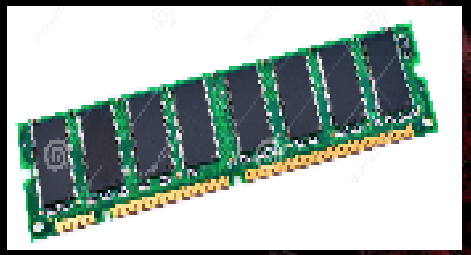
延迟事
务ID



DBLink

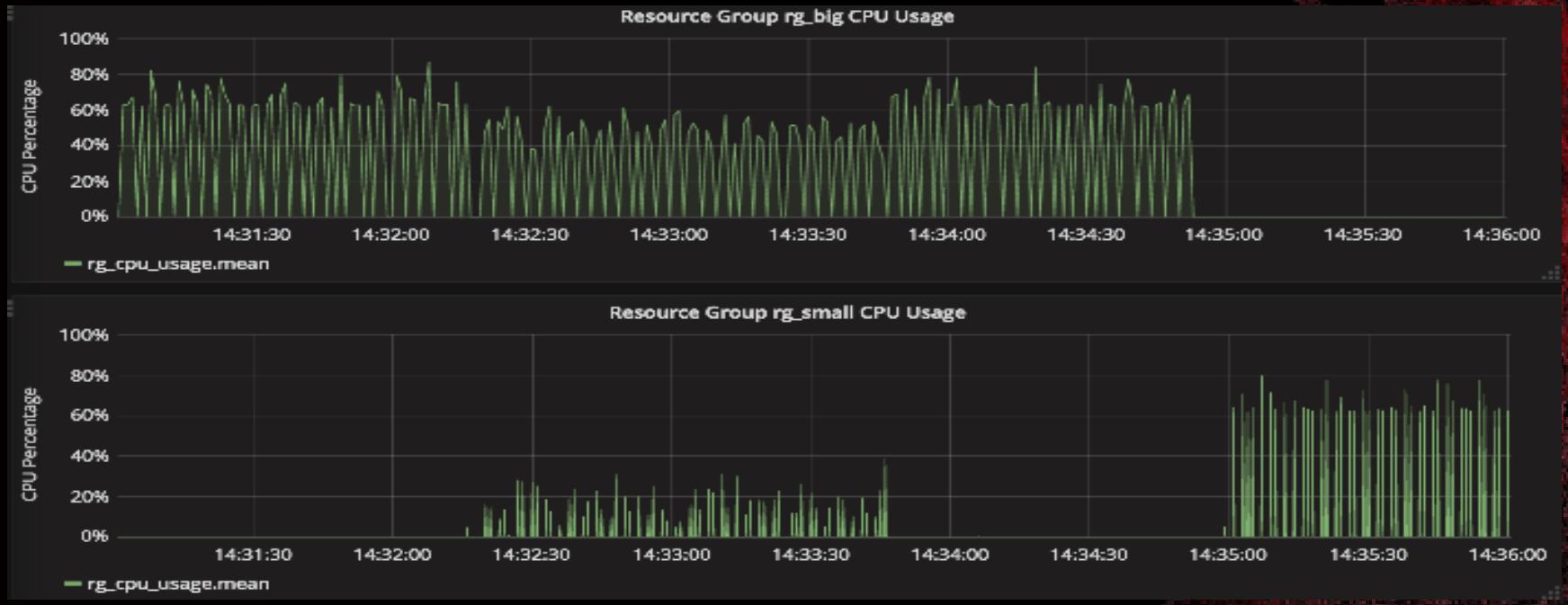
资源管理

精细

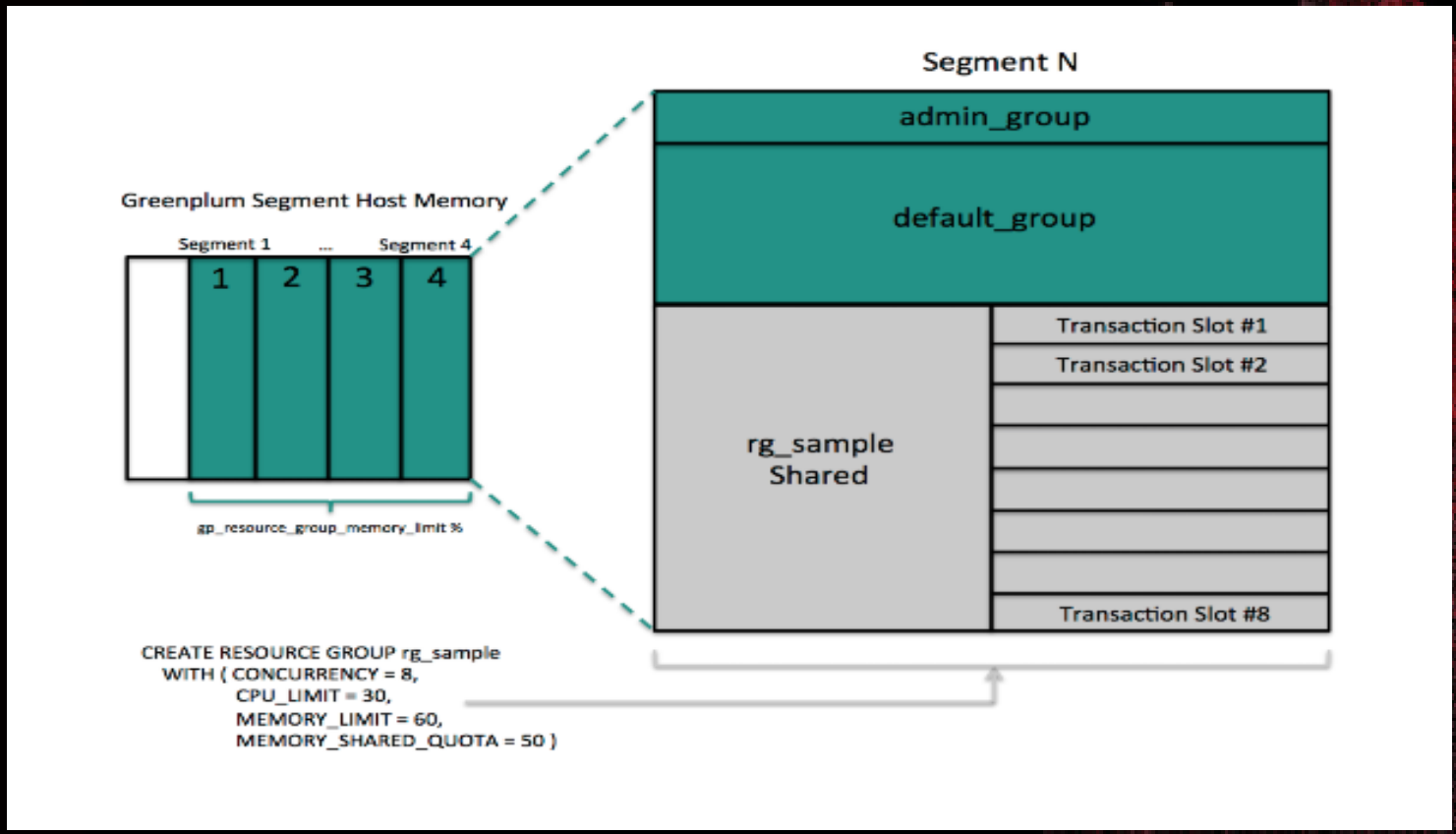


隔离

CPU资源分配


Resource Group rg_big CPU Usage
CPU Percentage
100%
80%
60%
40%
20%
0%
14:31:30
14:32:00
14:32:30
14:33:00
14:33:30
14:34:00
14:34:30
14:35:00
14:35:30
14:36:00
rg_cpu_usage.mean
Resource Group rg_small CPU Usage
CPU Percentage
100%
80%
60%
40%
20%
0%
14:31:30
14:32:00
14:32:30
14:33:00
14:33:30
14:34:00
14:34:30
14:35:00
14:35:30
14:36:00
rg_cpu_usage.mean

内存资源分配


Segment N
admin_group
default_group
Transaction Slot #1
Transaction Slot #2
rg_sample
Shared
Transaction Slot #8
Greenplum Segment Host Memory
Segment 1
...
Segment 4
1
2
3
4
gp_resource_group_memory_limit %
CREATE RESOURCE GROUP rg_sample
WITH (CONCURRENCY = 8,
CPU_LIMIT = 30,
MEMORY_LIMIT = 60,
MEMORY_SHARED_QUOTA = 50)

备份恢复和并行copy

- 重构gpbackup/gprestore
  - 不再锁pg_class表
  - 普通表只加ACCESS SHARE锁
  - 借助事务和MVCC
- 并行copy
  - COPY <table> TO <file> ON SEGMENT
  - COPY <table> FROM <file> ON SEGMENT

跨云部署

跨云是数据分析的当前需求和未来趋势

Infrastructure-Agnostic

Bare-Metal

Private Cloud

vmware vSphere

openstack.

Public Cloud

amazon web services

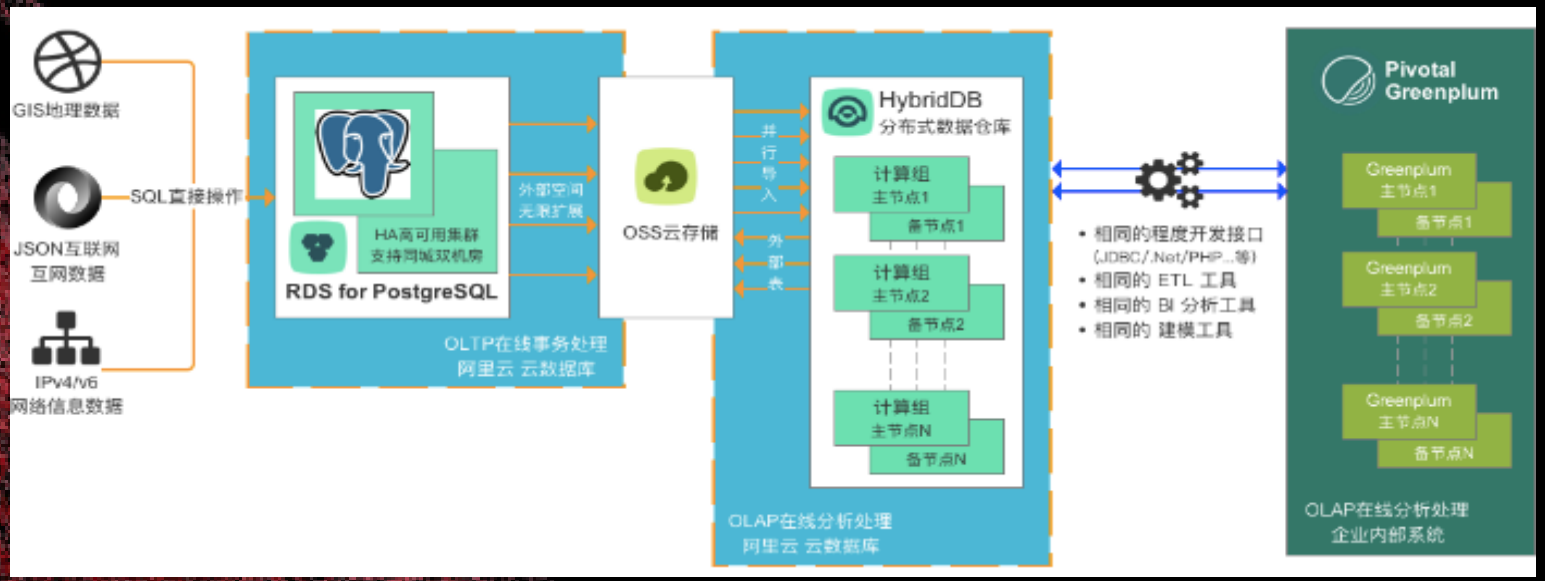
Microsoft Azure

Google Cloud Platform

“我们使用在AWS上运行的Greenplum构建了一个广告解决方案，给我们行业带来了巨大的变化。我们对于Greenplum 5具有的跨云功能和新分析能力感到非常兴奋，希望继续与Pivotal保持紧密的合作关系。”

—— John Conley, 数据仓库副总裁，Conversant

与阿里云HybridDB for PostgreSQL实现混合云打通



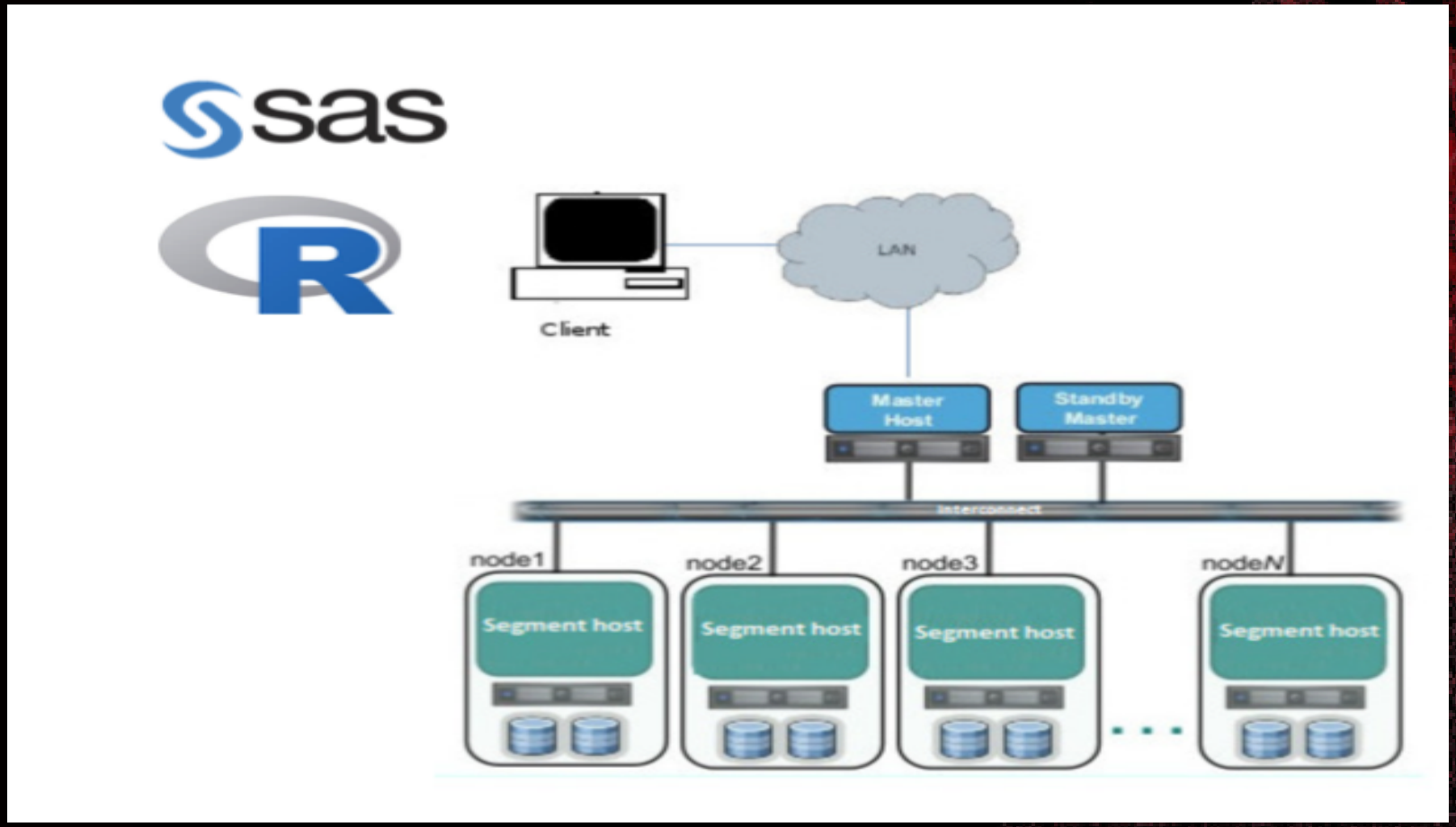
跨云部署的优点

- 避免供应商、硬件锁定
- 快速随处分析
- 多种配置、存储选项
- 成本控制

高级分析与机器学习

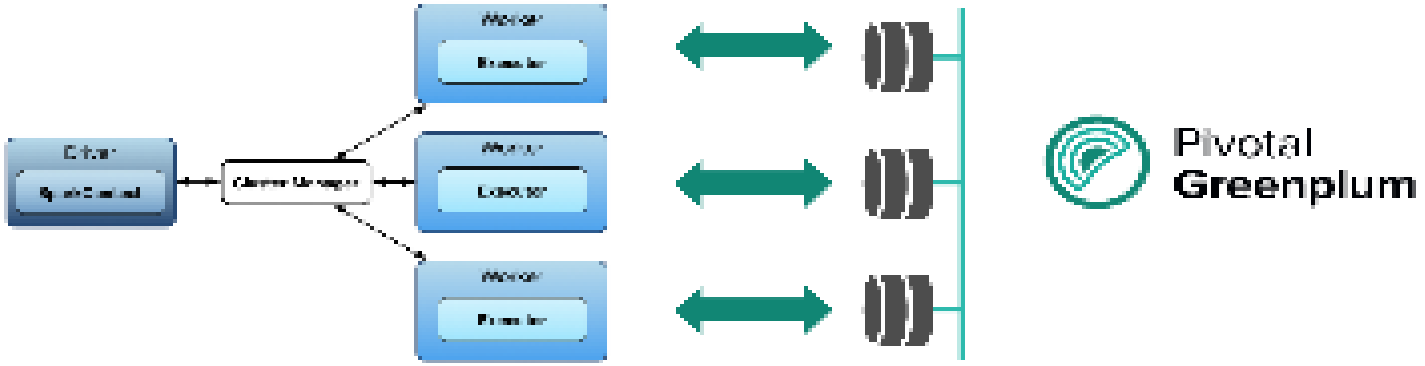
4种方式灵活应用

1 数据库外内存分析


sas
R
Client
LAN
Master Host
Standby Master
Interconnect
node1
node2
node3
nodeN
Segment host
Segment host
Segment host
Segment host

2 数据库外Spark集群分析

Data Transfer between Spark Executor(s) and GPDB segment(s)



Read committed

- Entire table
- Entire GPDB segment (GPDB only)
- Partition of table rows (JDBC)

Repeatable read

- Entire table

3 数据库内存储过程：数据科学包R/Python

Module Name	Description/Used For
Beautiful Soup	Navigating HTML and XML
Gensim	Topic modeling and document indexing
Keras	Deep learning
Lifelines	Survival analysis
lxml	XML and HTML processing
NLTK	Natural language toolkit
NumPy	Scientific computing
Pandas	Data analysis
Pattern-en	Part-of-speech tagging
pyLDAvis	Interactive topic model visualization
PyMC3	Statistical modeling and probabilistic machine learning
scikit-learn	Machine learning data mining and analysis
SciPy	Scientific computing
spaCy	Large scale natural language processing
StatsModels	Statistical modeling
Tensorflow	Numerical computation using data flow graphs
XGBoost	Gradient boosting, classifying, ranking

3 数据库内存储过程：TensorFlow

- Table:

  列: col1 & col2

  线性关系: col2 = w*col1 + b

- 计算参数:

  Select tfTrain(agg_train(col1),

  agg_train(col2)) from test;

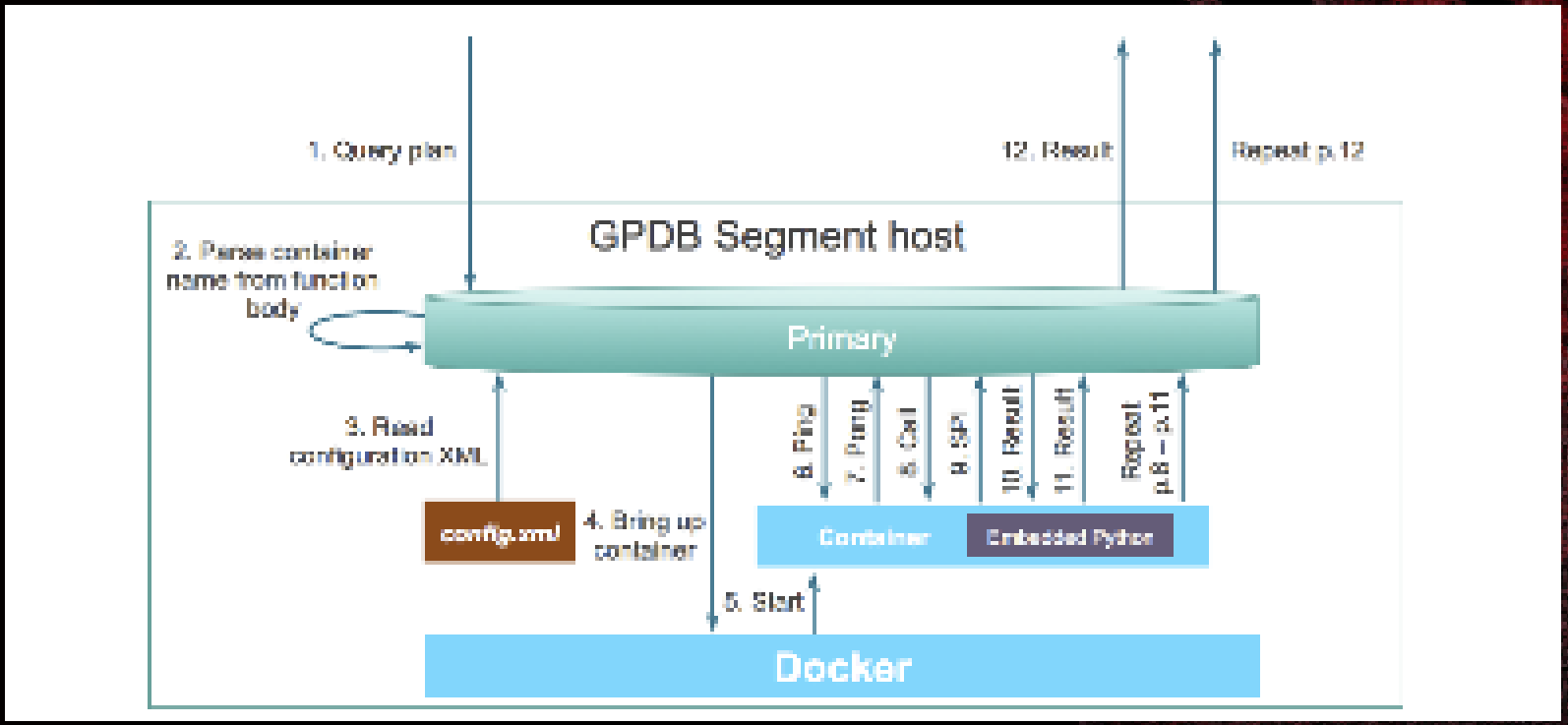
col1	col2
1	0.4
2	0.5
5	0.8
...	...

3 数据库内存储过程：TensorFlow

```
create function sfunc_train(state float[], a float)
returns float[] as
$$
    state.append(a)
    return state
$$ language plpythonu;

create aggregate agg_train(float)
(
    sfunc=sfunc_train,
    stype=float[],
    initcond='{}'
)
```

```
create function tfTrain(x_data float[], y_data float[])
returns numeric[] as
$$
    import tensorflow as tf
    import numpy as np

    W = tf.Variable(tf.random_uniform([1], -1.0, 1.0))
    b = tf.Variable(tf.zeros([1]))

    y = W * x_data + b

    loss = tf.reduce_mean(tf.square(y - y_data))
    optimizer = tf.train.GradientDescentOptimizer(0.5)
    train = optimizer.minimize(loss)

    init = tf.initialize_all_variables()
    sess = tf.Session()
    sess.run(init)
    for step in range(201):
        sess.run(train)
    return np.append(sess.run(W)[0], sess.run(b)[0])
$$ language plpythonu;
```

3 数据库内存储过程： PL/Container



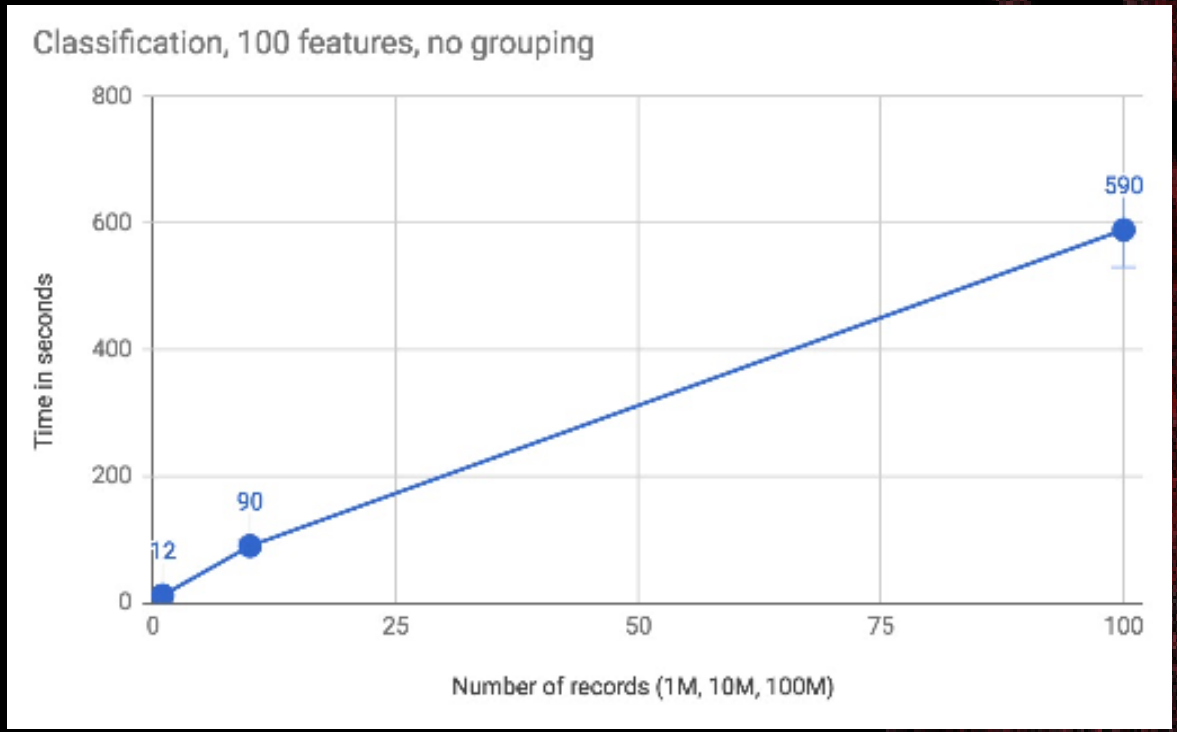
4 数据库内Madlib

```
SELECT madlib.linregr_train('houses',                    --- Input table
                            'houses_out',                --- Output table
                            'price',                     --- Variable to predict
                            'ARRAY[1, tax, bath, size]', --- Features in data
                            'bedroom'                    --- Group data to create
                                                         ---    multiple models
                            )
```

```
SELECT houses.*,
    madlib.linregr_predict(ARRAY[1, tax, bath, size],   --- Use same features
                           model.coef)as predict
FROM houses_test, houses_out as model;                  --- Combine test data
                                                        ---    and model table
```

4 数据库内Madlib: 特点

- 简单易用
- 更大的并行性
- 更好的可扩展性



- SVM分类性能
- 1 master
  4 segments

OLAP和OLTP混合负载

同时满足不同类型业务需求

混合负载需求

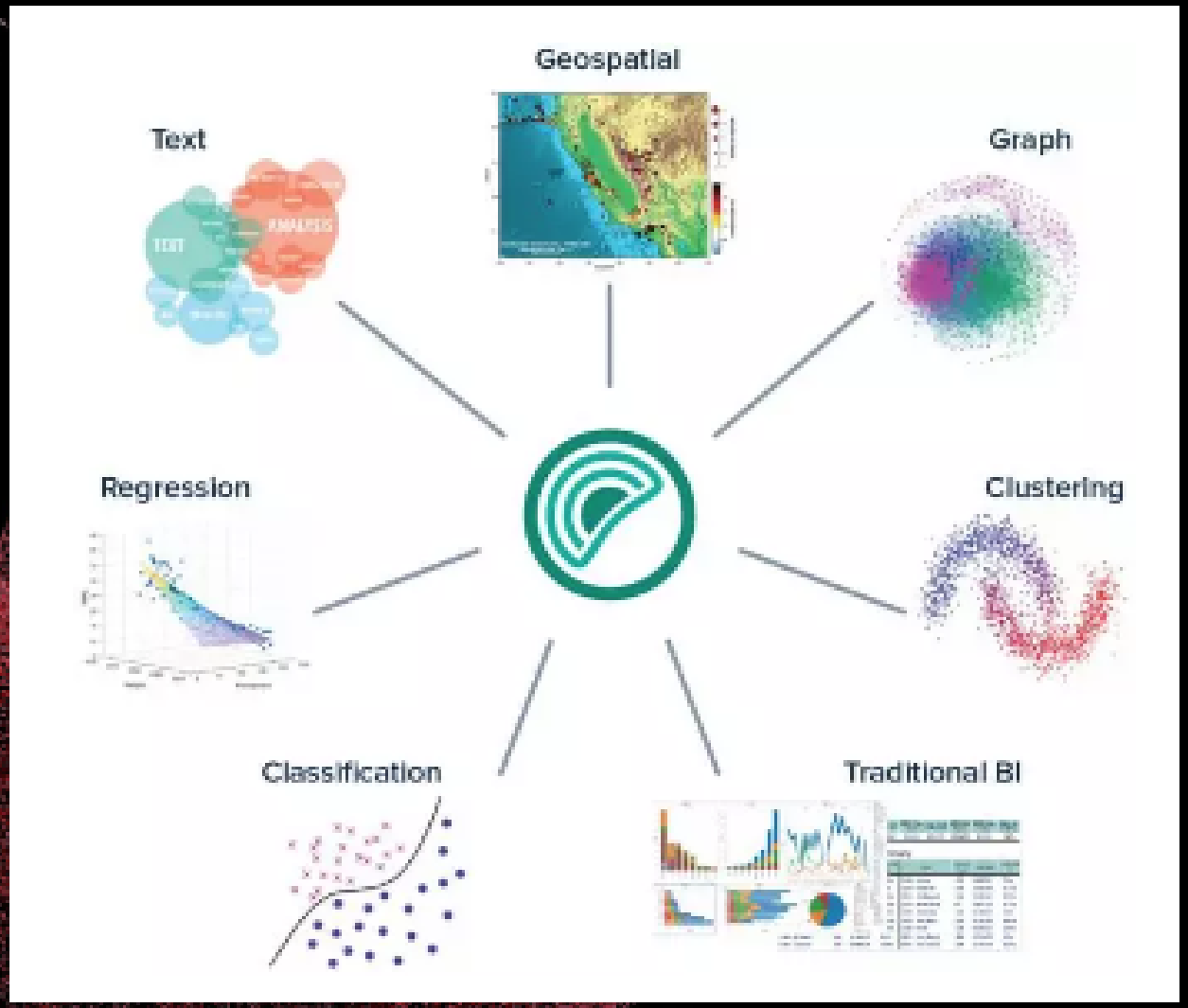
- 客户应用的需求
- 避免数据导入导出
- 社区的声音

支持混合负载

- ORCA优化改进
- 偏向短查询
- SELECT - INSERT - UPDATE/DELETE

广泛适用

适合不同背景用户和各种场景的大数据平台


Geospatial
Text
Graph
Regression
Clustering
Classification
Traditional BI

2017 杭州·云栖大会
THE COMPUTING CONFERENCE

Pivotal

飞天 / 智能
APSARA INTELLIGENCE

欢迎关注Pivotal中国研发中心

THANK YOU

扫码参与云栖大会调查问卷，赢取大会限量纪念品