DTCC

2013中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2013

大数据，商业智能的挑战

搬运大数据的亨利 2013.4.20

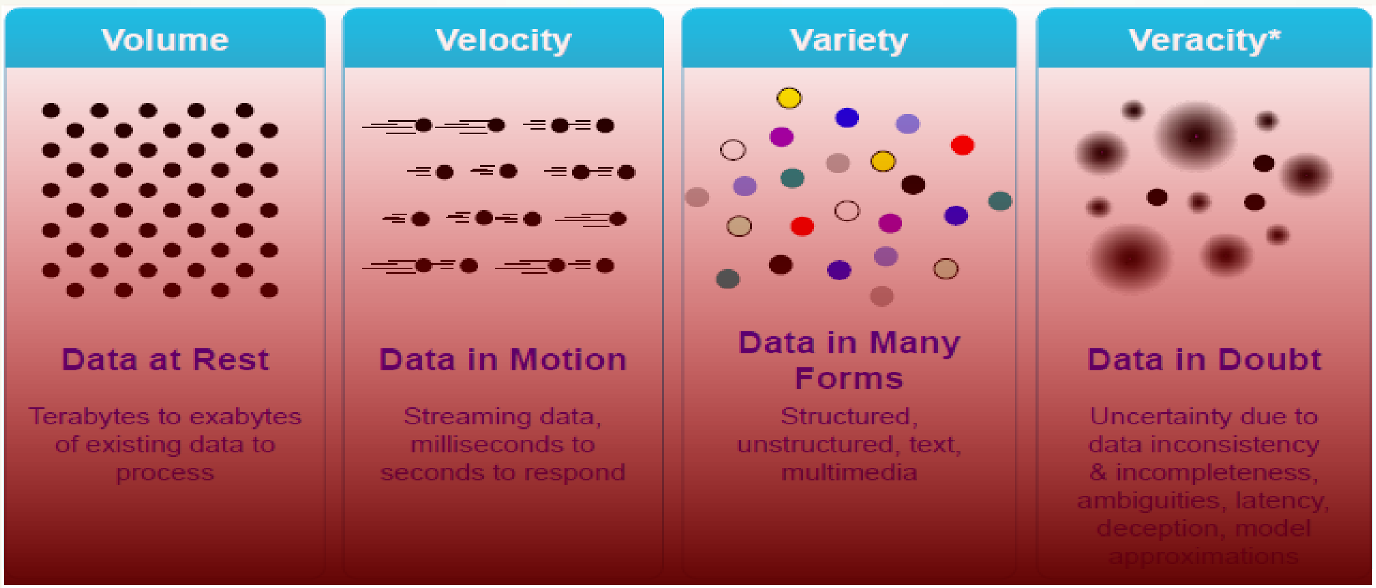
目录

1. 大数据背景
2. 开源大数据生态圈和商用大数据生态圈
3. 开源前端生态圈和商用前端生态圈
4. 大数据BI系统的底层技术
5. 大数据BI系统的演变过程
6. 建设大数据BI系统的思考

大数据背景

大数据的4V

1. 数据量大(Volume)
2. 速度快(Velocity)
3. 类型多(Variety)
4. 价值密度低(Value)



大数据背景

更多的数据？

谷歌自始至终都认为数据是越多越好，用谷歌产品研发总监Peter Norvig的话就是：更多的数据胜过更好的算法

开源大数据生态圈

诞生：谷歌

大数据领袖谷歌，于2003年起发布一系列论文:

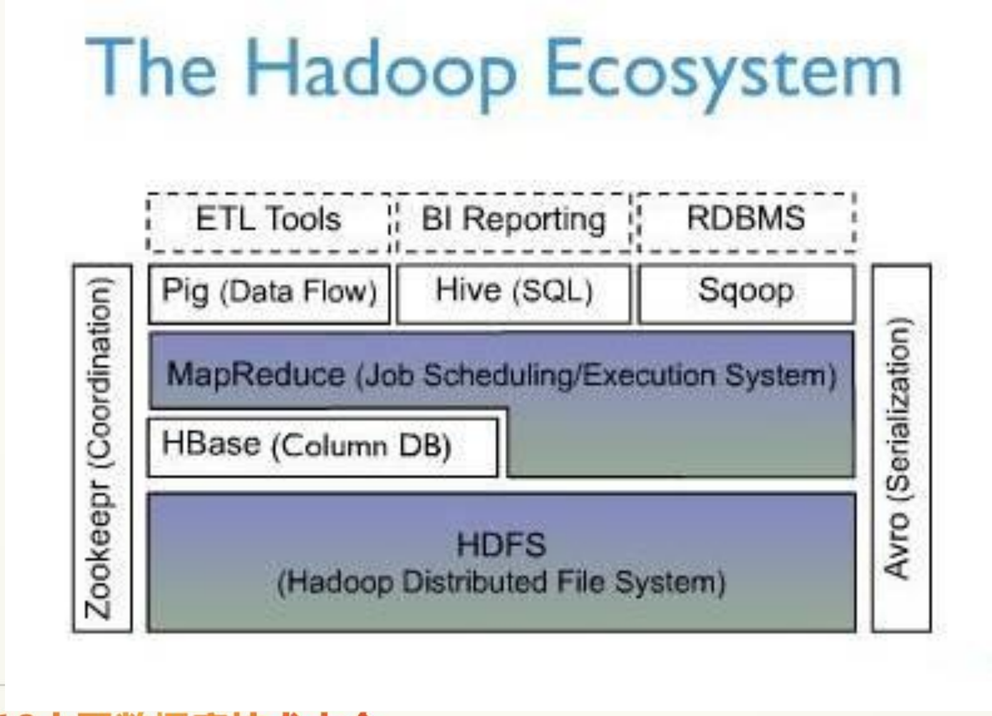
1. 《The Google File System 》
2. 《MapReduce: Simplified Data Processing on Large Clusters》
3. 《Bigtable: A Distributed Storage System for Structured Data》

战火被点燃，从此进入大数据时代

开源大数据生态圈

诞生：开源大数据生态圈

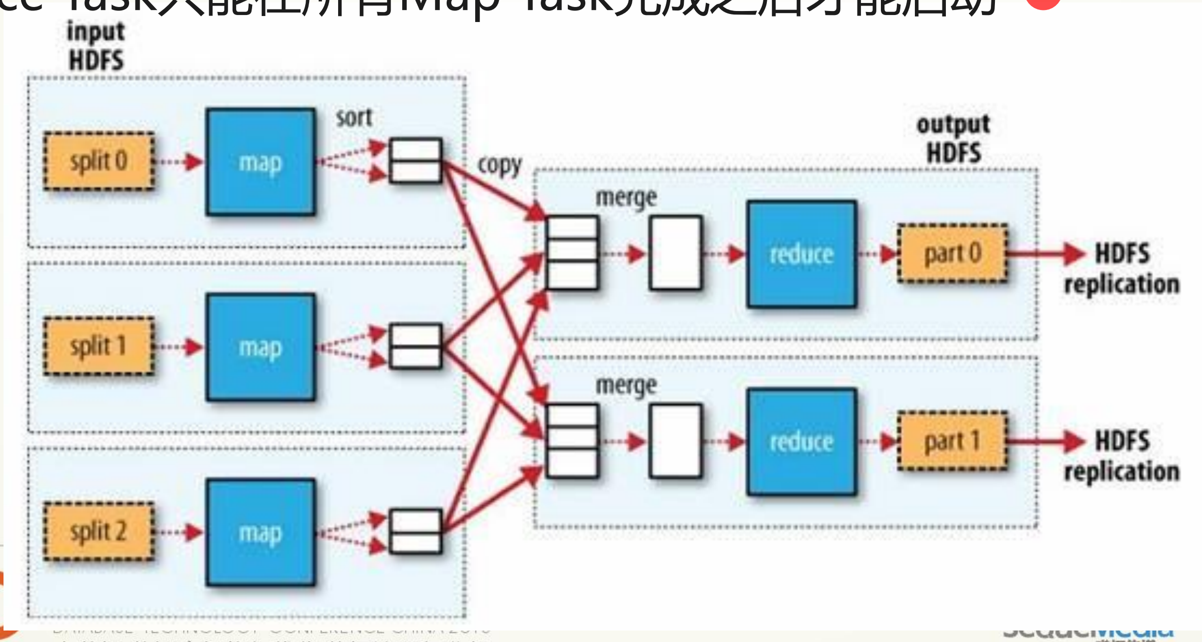
1. Hadoop HDFS、Hadoop MapReduce, HBase、Hive渐次诞生，早期Hadoop生态圈逐步形成
2. Hypertable是另类。它存在于Hadoop生态圈之外，但也曾经有一些用户



开源大数据生态圈

Hadoop MapReduce不适合实时计算？

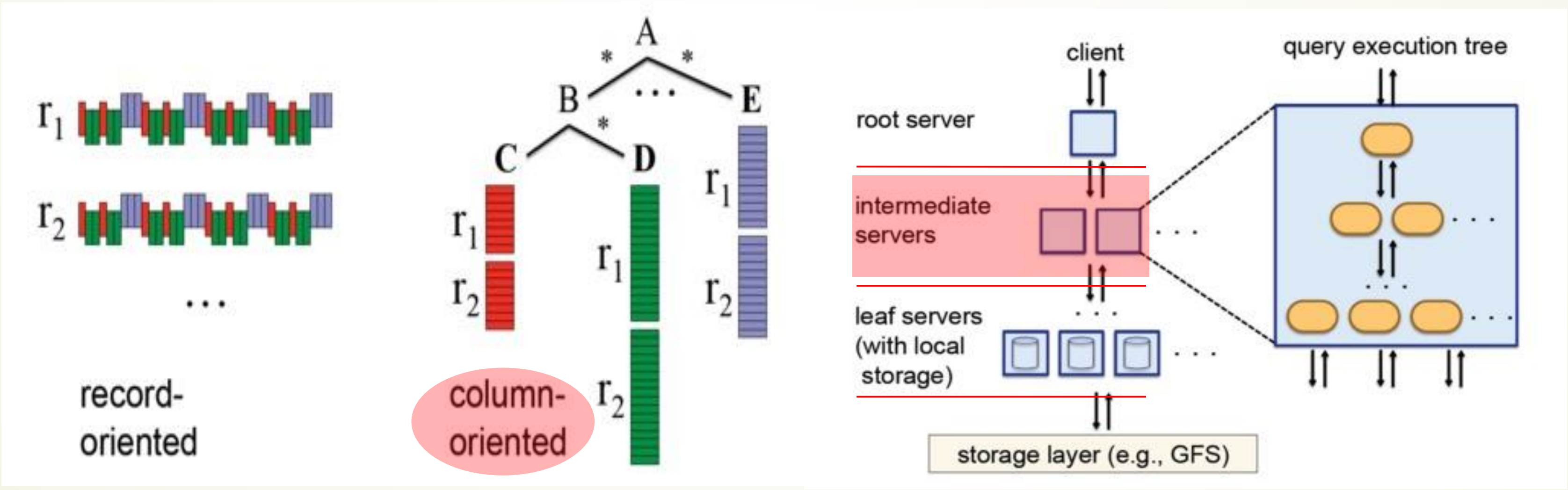
1. 任务分配Server不会将信息Push到计算Node，而是让计算Node通过心跳去Pull任务
2. 基于框架的通用性，MapReduce代码也会在HDFS中传送，在各计算Node展开，再通过启动新JVM进程装载并运行
3. 类似的JVM进程启停有5、6次之多
4. Reduce Task只能在所有Map Task完成之后才能启动



开源大数据生态圈

中期：谷歌

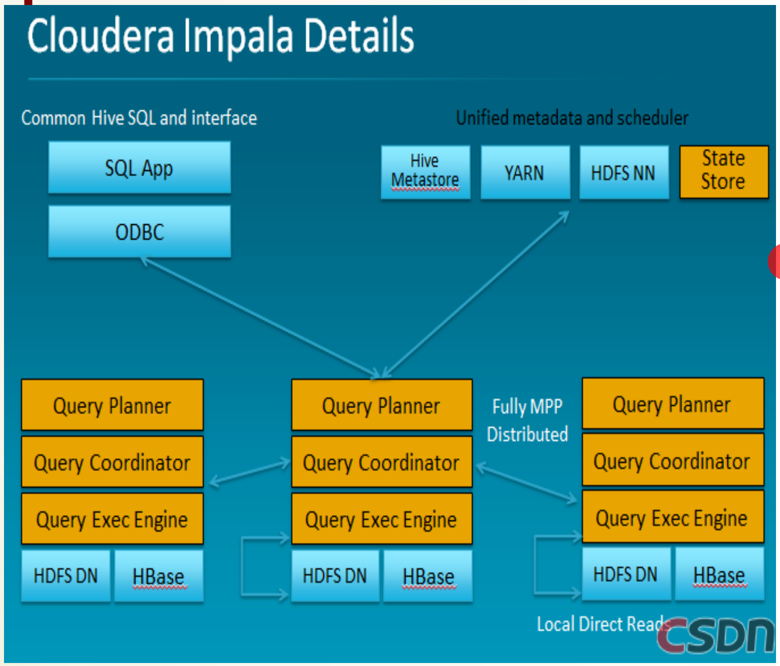
. 为了高扩展性，我们容忍了高延迟●
. 2010年，谷歌又发布了论文:《Dremel: Interactive Analysis of Web Scale Datasets》
. 论文撰写于2006年●●●
. 平静再次被打破

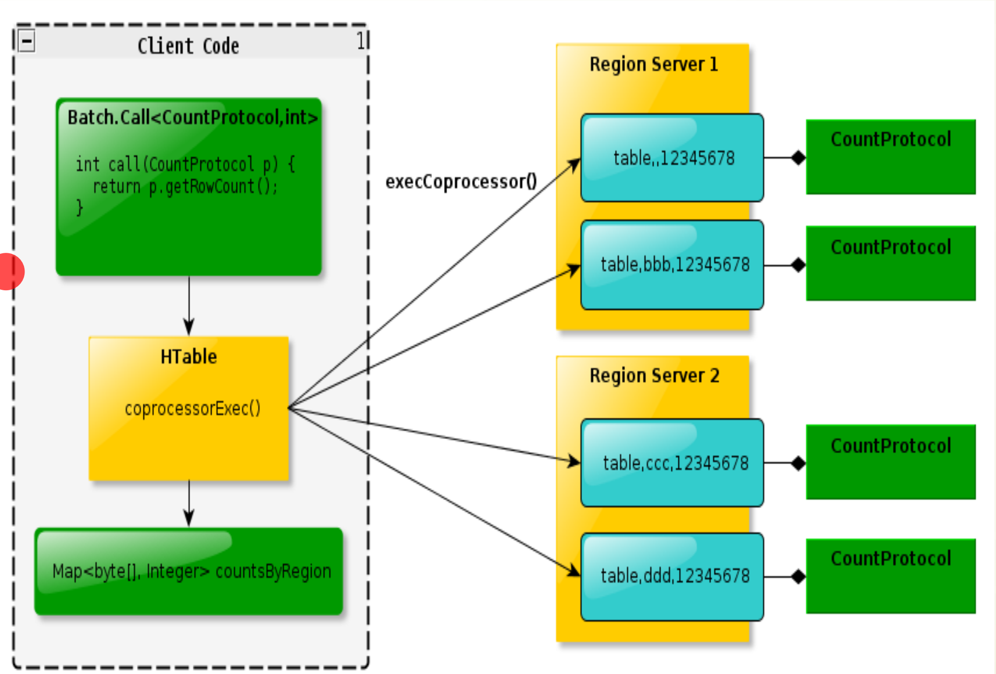


开源大数据生态圈

中期：开源 Caution!

在Hadoop MapReduce框架之外，衍生出了一批低延迟的大数据分析项目：Cloudera Impala, MapR Drill, HBase Coprocessor, Spark, etc.





开源大数据生态圈

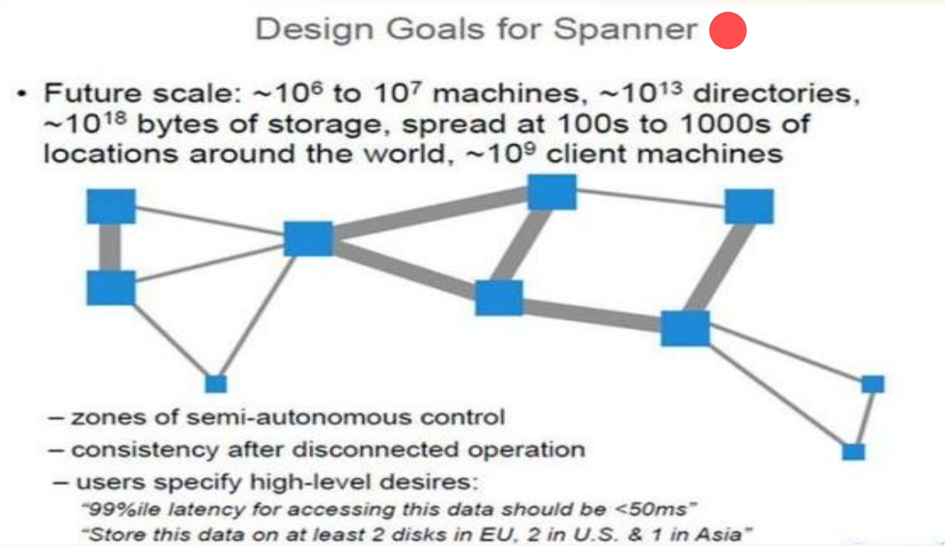
后来：谷歌

NoSQL引领着IT界时尚圈，几乎家喻户晓；低延迟的大数据分析项目在开源界刚出现没多久，谷歌又烧了第三把火。这次它发表的两篇论文是：

1. 《F1 - The Fault-Tolerant Distributed RDBMS Supporting Google's Ad Business》
2. 《Spanner: Google's Globally-Distributed Database》

. 多年以来，很多人以NoSQL为时尚。到后来才发现，原来谷歌早就在用SQL语句自如地访问自己的MPP数据库：F1和Spanner了

. NoSQL定义曾经有很多版本，谷歌的最新定义是：No Time to Support SQL



开源大数据生态圈

后来：开源界整体失声

开源大数据生态圈

总结：开源大数据生态圈落后，努力！

基本套路：

1. 2003年，谷歌点燃了大数据战火
2. 2010年，谷歌又给大数据放了一把火
3. 2012年，谷歌再给大数据放了第三把火

开源大数据生态圈

几个另类：MySQL, MonetDB, PostgreSQL

想过打造属于自己的MPP 数据仓库吗？

商用大数据生态圈

一体机数据库/数据仓库:

国外	IBM PureData(Netezza), Oracle Exadata, SAP Hana等等
国内	

Scale-out 数据仓库:

国外	Teradata AsterData, EMC GreenPlum, HP Vertica 等等
国内	GBase等等

Scale-up 数据集市:

国外	QlikView、 Tableau 等等
国内	

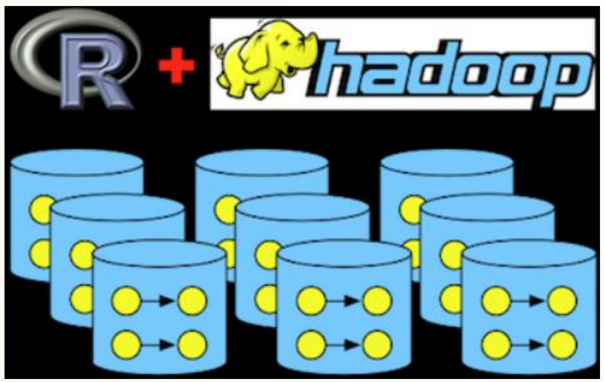
Scale-out 数据集市:

国外	Style Intelligence 等等
国内	Yonghong Data Mart 等等

开源前端生态圈

开源:

统计分析和数据挖掘	R, RHadoop, Mahout等等
展现分析	JasperSoft，Pentaho, Openi, Birt等等



商用前端生态圈

统计分析和数据挖掘:

国外	SAS，SPSS等等
国内	

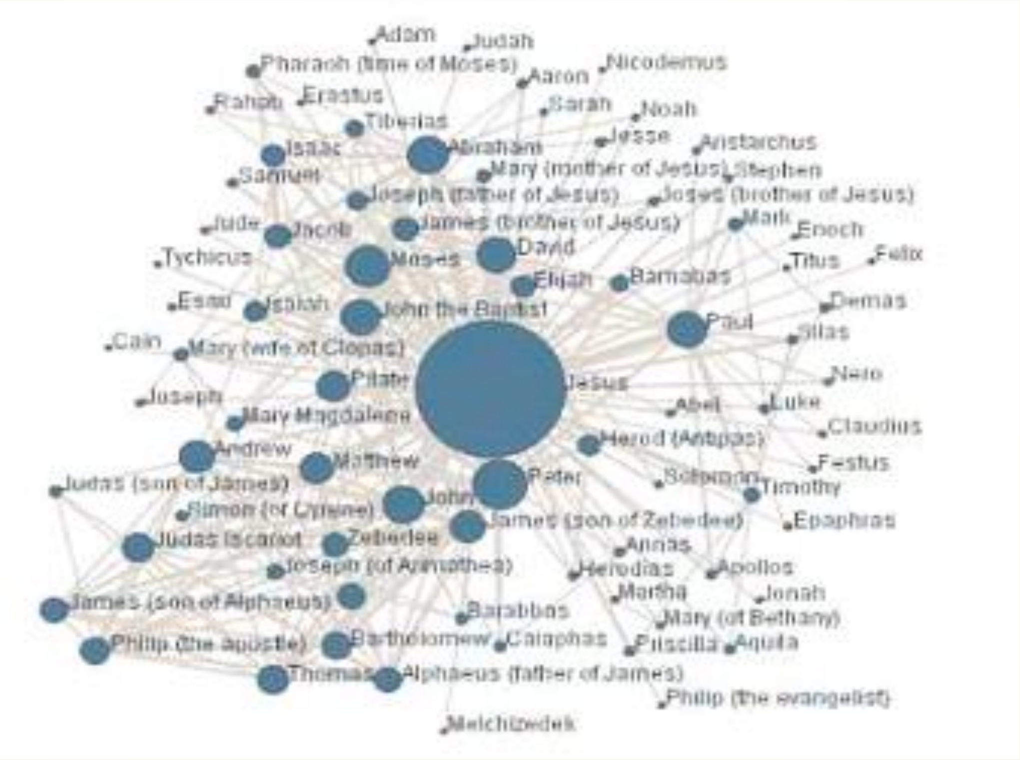
展现分析:

国外	Cognos, BO, Microsoft, Oracle, Microstrategy等等
	QlikView,Tableau, Spotfire, Style Intelligence等等 ●●●
国内	SmartBI, Yonghong Z-Suite, CellSoft等等 ●

大数据BI系统的底层技术

必要的底层技术

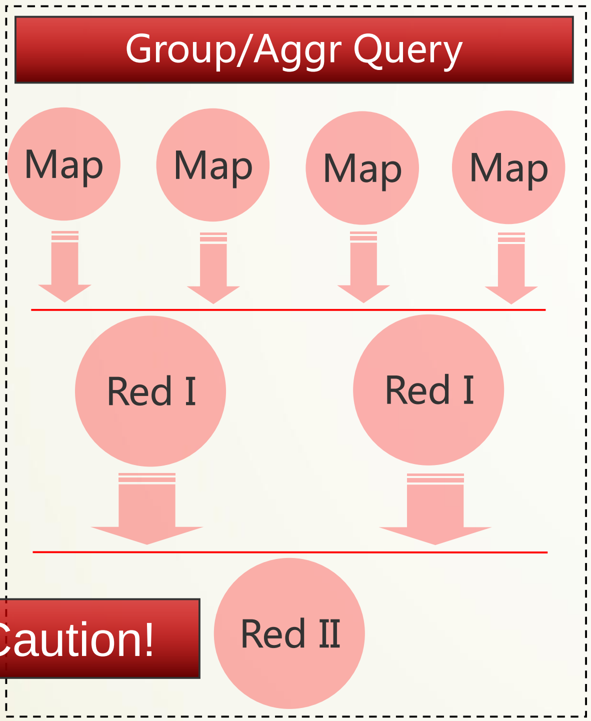
1. 分布式计算
2. 分布式通信
3. 内存计算
4. 列存储
5. 库内计算



大数据BI系统的底层技术

分布式计算

1. Offline MPP
2. Online MPP



Caution!

hadoop MapReduce — 1. Offline

Spark Lightning-Fast Cluster Computing

POWERED BY cloudera IMPALA

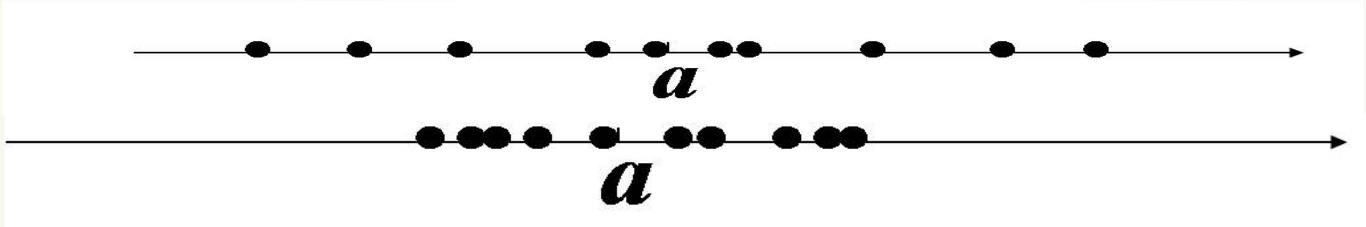
Google Dremel — 2. Online

大数据BI系统的底层技术

分布式通信

1. Performance
2. Load
3. Variance

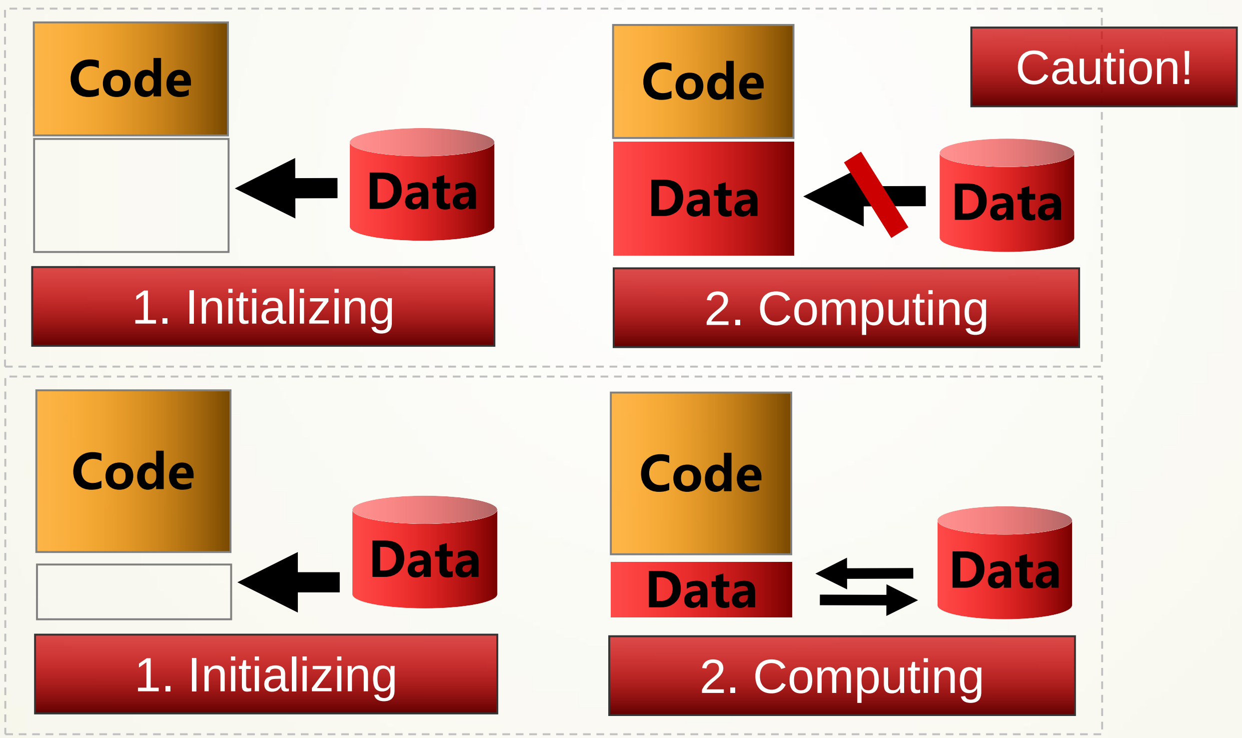
Average(a) 是相同值5，但是Variance(a)差距很大



大数据BI系统的底层技术

内存计算

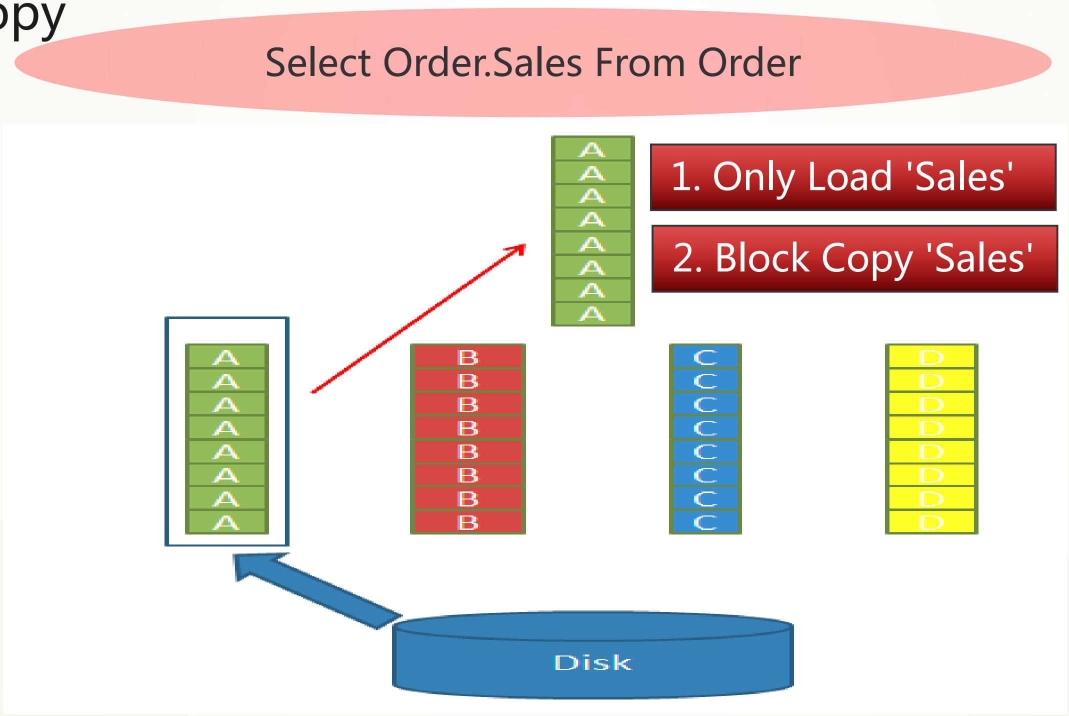
1. All Data
2. Hot Data



大数据BI系统的底层技术

列存储

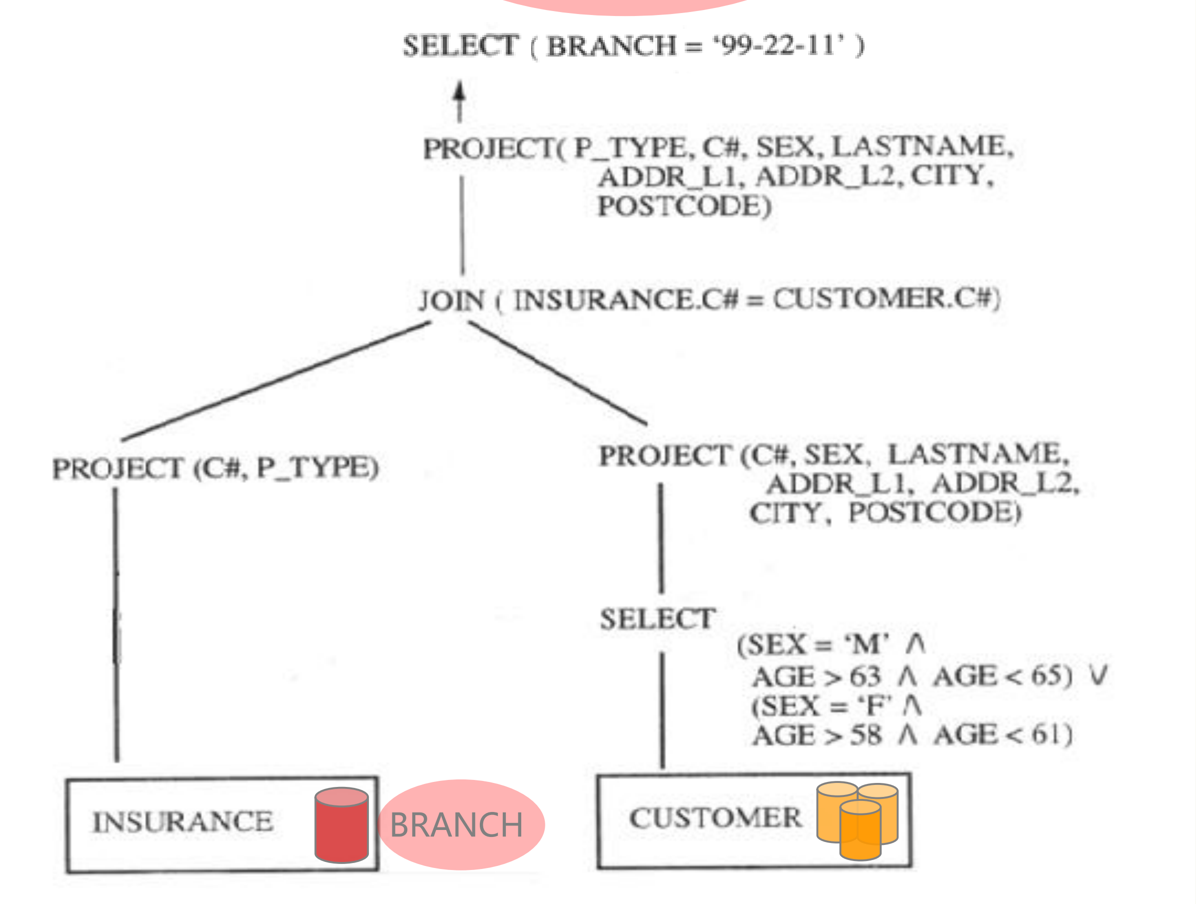
1. Compact (Run Length, Dict, Range, LZW，Delta, etc.)
2. Partial I/O
3. Block Copy



大数据BI系统的底层技术

库内计算 - 条件查询

条件能否压下去？



大数据BI系统的底层技术

库内计算 - 聚集查询

Sum →
1. **Map**: Sum
2. **Reduce**: Sum_Sum

Count →
1. **Map**: Count
2. **Reduce**: Sum_Count

Variance →
0. **Formula**: (n * Σx2 – (Σx)2) / (n * (n – 1))
1. **Map**: Count, Sum, SumSQ
2. **Reduce**: Sum_Count, Sum_Sum, Sum_SumSQ

Covariance →
0. **Formula**: 1/n * Σxy - μx * μy
1. **Map**: Sum_x, Sum_y, SumWT_xy,Count_x
2. **Reduce**: Sum_Sum_x, Sum_Sum_y, Sum_SumWT_xy, Sum_Count

大数据BI系统的底层技术

库内计算 子查询

子查询怎么转化为Map Reduce？

有一些论文专门研究子查询的改写。

例如：Won Kim:
《On Optimizing an SQL-like Nested Query》

例如：C´esar A. GalindoLegaria & Milind M. Joshi:
《Orthogonal Optimization of Subqueries and Aggregation》

子查询大致可以分为以下几类：

1. Type-A, 2. Type-N, 3. Type-J, 4. Type-JA

例如，Type-J 子查询可改写为Join(关联):

Select t1.col1 from t1 where t1.col2 in (select t2.col2 from t2 where t1.jcol=t2.jcol)

1. Merge the tables in the subquery to the ‘from’ clause of the main query.
2. Add subquery conditions (including joins) to the main query’s ‘where’ clause. The conditions should be combined using an AND operator.
3. Replace Ri.Ci op (select Rj.Cj) with a join: Ri.Ci nop Rj.Cj, and AND it with the entire condition.
   - ⇨ *If op is IN, the condition is Ri.Ci = Rj.Cj*
   - ⇨ *If op is NOT IN, the condition is NOT (Ri.Ci = Rj.Cj)*

大数据BI系统的演变过程

第一阶段

1. OLAP on OLTP

第二阶段

1. Materialized View
2. Stage, ODS, DWD, DWA
3. ROLAP, MOLAP, HOLAP

第三阶段

1. 开源社区
2. 一体机 , MPP DW, MPP DM

这也是几乎每个BI研发人员的进化之旅，直到...

建设大数据BI系统的思考

大数据BI系统并不仅仅是数据展现

1. 要能够掌握情况、分析问题、找到答案
2. 前端系统的交互和分析能力要强大：过滤(Filter)、钻取(Drill)、刷取(Brush)、缩放(Zoom)、关联(Associate)、变换(Transform)、动态计算（Dynamic Calculation）、链接(Link)
3. 数据挖掘、预测将变成重要的需求

中美日BI系统建设目的

1.把生产品质管理作为设立BI目的的企业，中国(45.9%)比美国、日本多2成左右。

2.中国有31.6%的企业把电子商务相关分析作为设立BI的目的，美国其次为23.3%，而日本这一比例仅为2.7%。

3.美国有33.2%企业是为了做供应链分析，中国其次29.5%，日本17.1%。

4.中美都是有2成多企业把分析网络上的客户行为作为设立BI的目的，而一直把客户看得很重的日本企业这一比例仅为3.1%。同样，将BI用于客户支持和呼叫中心上面的日本企业占比也很低，不足中国和美国的一半。

5.美国近4成企业目的是需求预测，而中国和日本这一比例都是2成左右。

6.中国和美国有2成左右的企业将BI用于社会影响力相关分析，而日本企业这一比例仅为2%

7.美国企业将BI解决方案用于风险管理、发现并防止营私舞弊的比例比中国和日本高很多。

建设大数据BI系统的思考

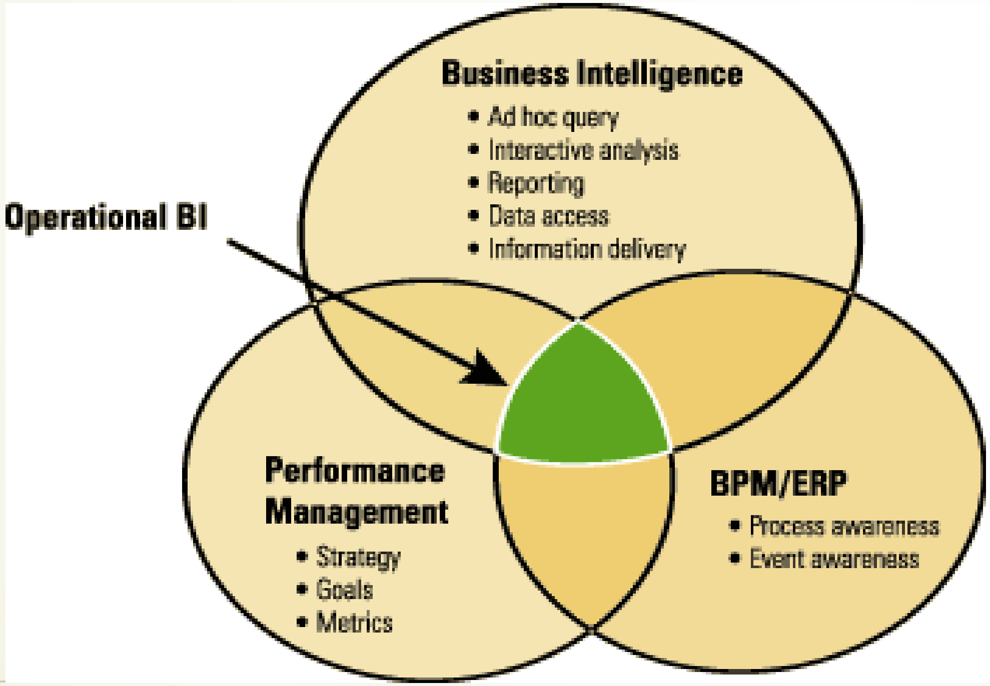
性能 — 最重要的功能？

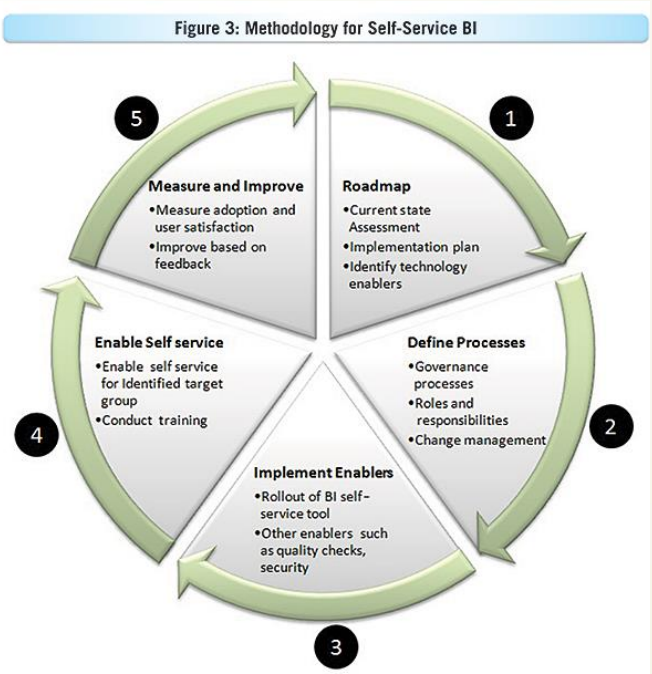
1. 性能需要衡量交付一个BI应用的整体周期及难易程度
2. 谨慎采用 Cache、Preaggregation等不合乎直觉的技术：优化效果不稳定；难以管理，可能失去控制
3. 性能最好不以牺牲粒度(Granularity)达到：粒度越细越好
4. 海量数据，实时处理

建设大数据BI系统的思考

建设方向

1. 业务驱动优于数据驱动
2. 自助式(Self-Service)的大数据BI系统





建设大数据BI系统的思考

建设原则

1. 敏捷开发原则（ADP）：Agile Development Principle. 好系统不是一蹴而就，先切入进去再持续迭代
2. 开闭原则（OCP）：Open to Extension, Close to Modification. 减少BI应用对数据层的污染
3. 依赖倒置原则（DIP）：Dependency Inversion Principle. 合理架构，降低数据与应用之间的依赖度
4. 里氏替换原则（LSP）：Liskov Substitution Principle. 开放标准，使各模块的替换性好

建设大数据BI系统的思考

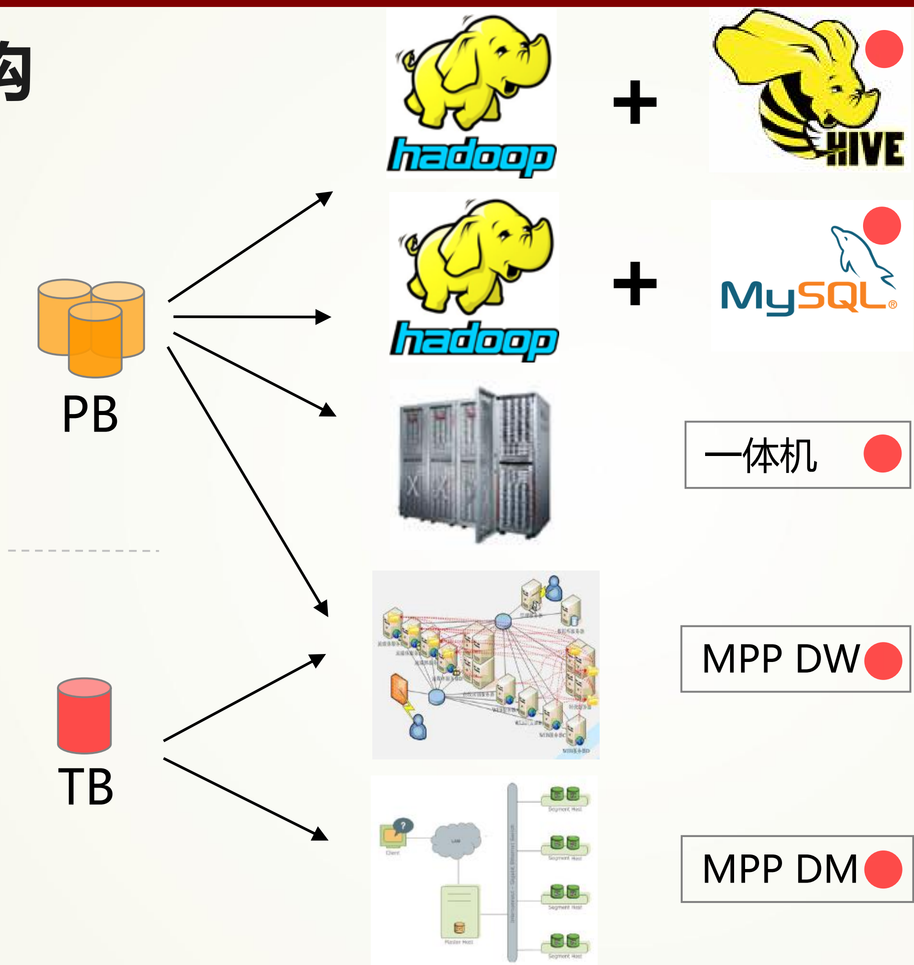
产品选型

1. 正确面对开源和商业产品：开源是一种商业模式，没有免费的午餐
2. 正确面对国际和国内产品：在 TCO、Service、Customization等方面仔细权衡
3. 大数据BI系统的TCO并不一定就高：曾经观察到一些项目以十万左右的开销支持了1 TB数据量的实时分析(包括软硬件)
4. 选择最适合自己的、服务最好的产品和技术

Scale-out | Scale-up | x86 Server | Non-x86 Server

建设大数据BI系统的思考

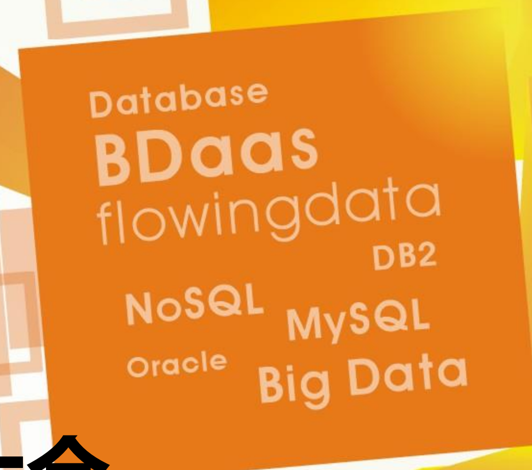
常见架构



传统的BI系统架构并非一无是处，同样值得我们学习借鉴

谢谢！欢迎莅临

2013中国数据库技术大会



搬运大数据的亨利 henryhe.bi@gmail.com