数据科学的挑战

Endeavor of TalkingData

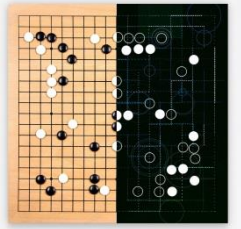
张夏天　Chief Data Scientist
TalkingData

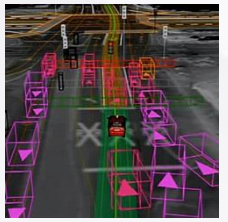
数据科学家的一天

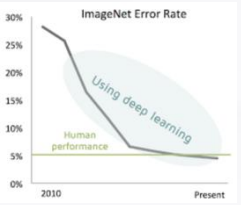
TALKINGDATA

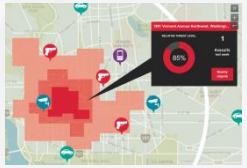
数据科学的盛世

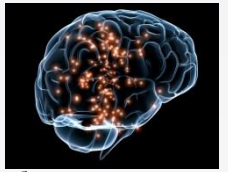
BigData孕育出一个又一个奇迹



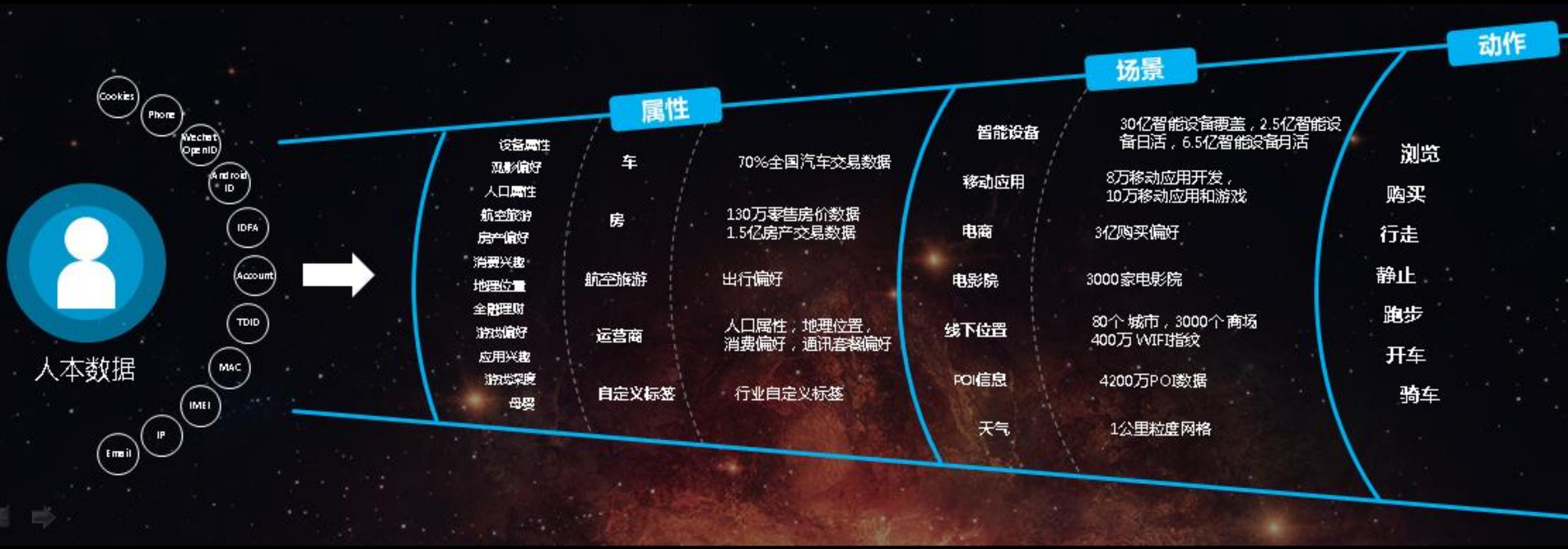




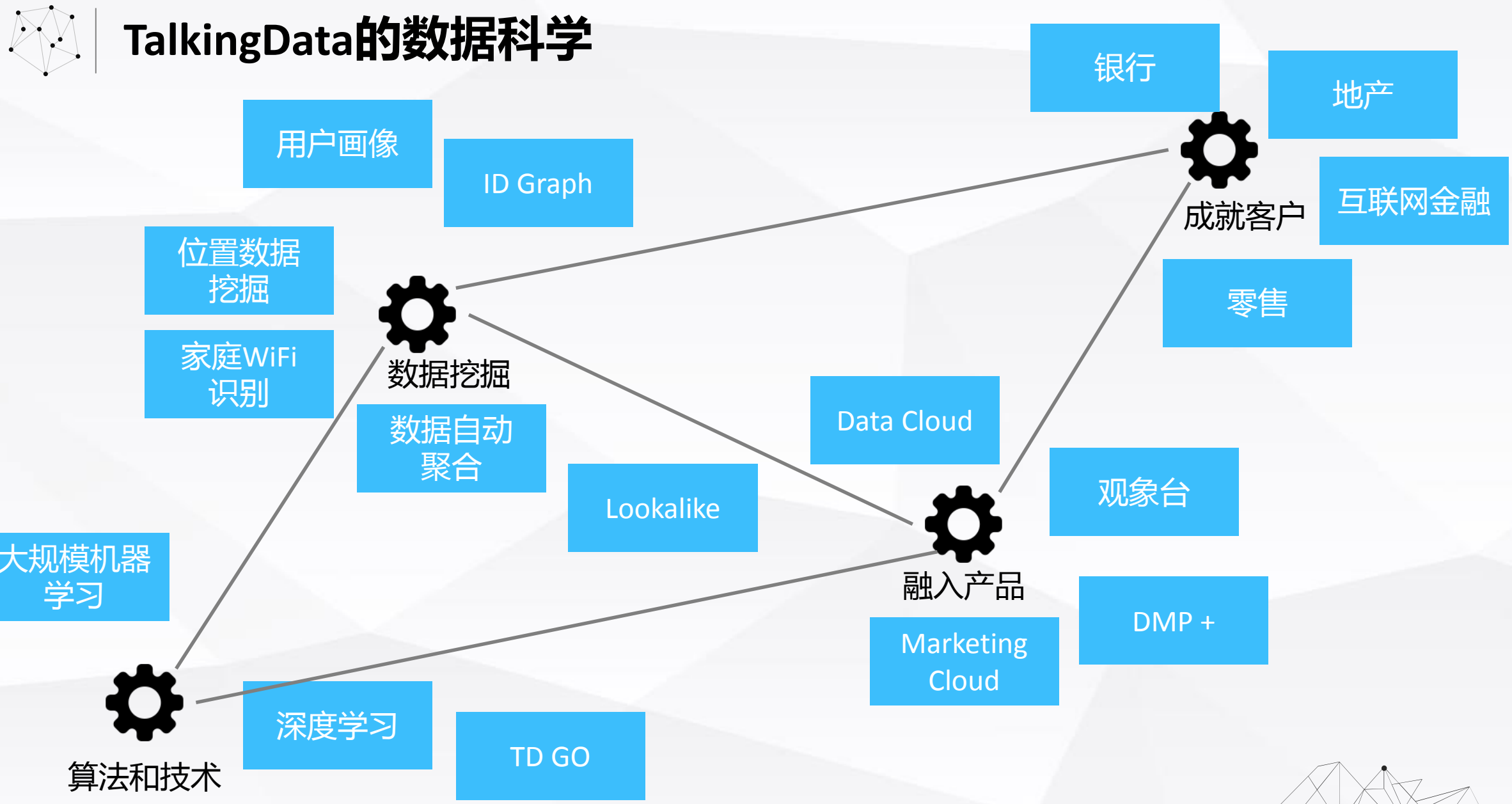




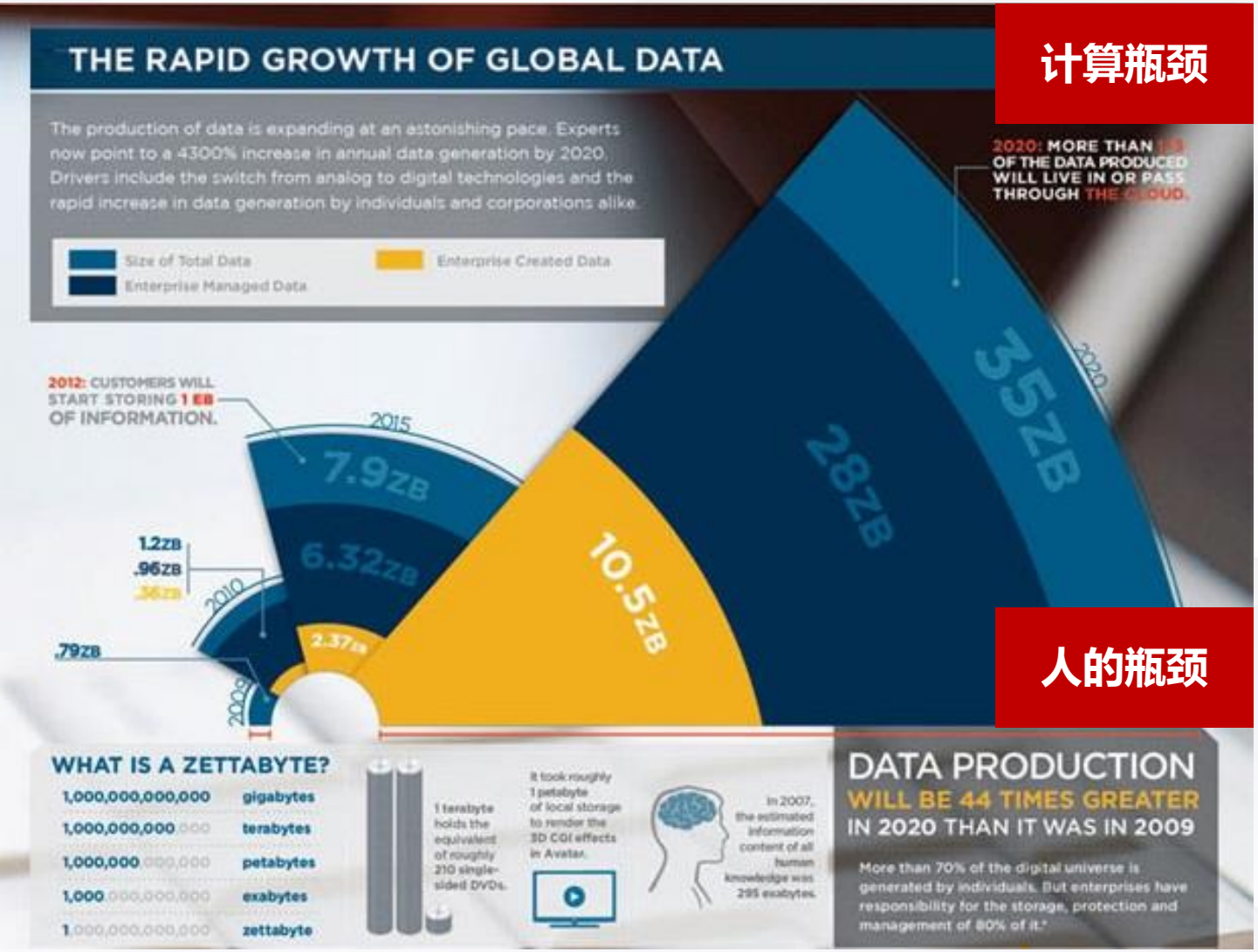
TalkingData 的数据



TalkingData的数据科学



数据科学的挑战



计算瓶颈

人的瓶颈

Data

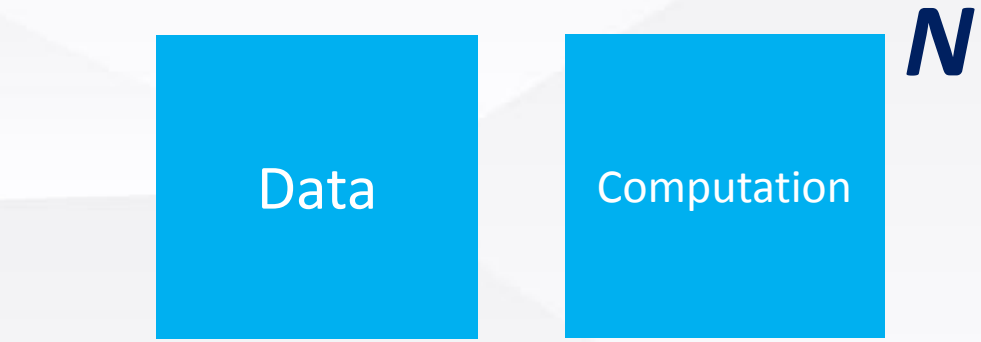
Computation

190,000

PROJECTED SHORTAGE IN DATA SCIENTISTS BY 2018

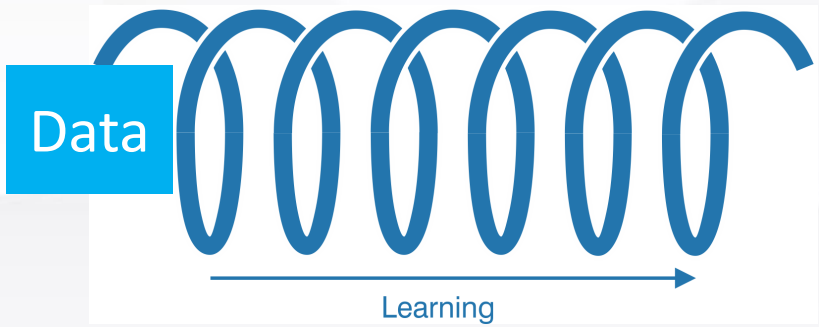
突破计算的瓶颈

算法的计算瓶颈



	single	multi
LWLR	$O(mn^2+n^3)$	$O(\frac{mn^2}{P}+\frac{n^3}{P'}+n^2\log(P))$
LR	$O(mn^2+n^3)$	$O(\frac{mn^2}{P}+\frac{n^3}{P'}+n^2\log(P))$
NB	$O(mn+nc)$	$O(\frac{mn}{P}+nc\log(P))$
NN	$O(mn+nc)$	$O(\frac{mn}{P}+nc\log(P))$
GDA	$O(mn^2+n^3)$	$O(\frac{mn^2}{P}+\frac{n^3}{P'}+n^2\log(P))$
PCA	$O(mn^2+n^3)$	$O(\frac{mn^2}{P}+\frac{n^3}{P'}+n^2\log(P))$
ICA	$O(mn^2+n^3)$	$O(\frac{mn^2}{P}+\frac{n^3}{P'}+n^2\log(P))$
k-means	$O(mnc)$	$O(\frac{mnc}{P}+mn\log(P))$
EM	$O(mn^2+n^3)$	$O(\frac{mn^2}{P}+\frac{n^3}{P'}+n^2\log(P))$
SVM	$O(m^2n)$	$O(\frac{m^2n}{P}+n\log(P))$

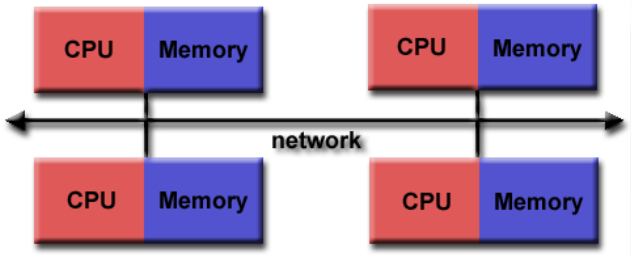
Cheng T. Chu, Sang K. Kim, Yi A. Lin, Yuanyuan Yu, Gary R. Bradski, Andrew Y. Ng, Kunle Olukotun , **Map-Reduce for Machine Learning on Multicore, NIPS, 2006.**



计算量倍数增长

IO开销巨大

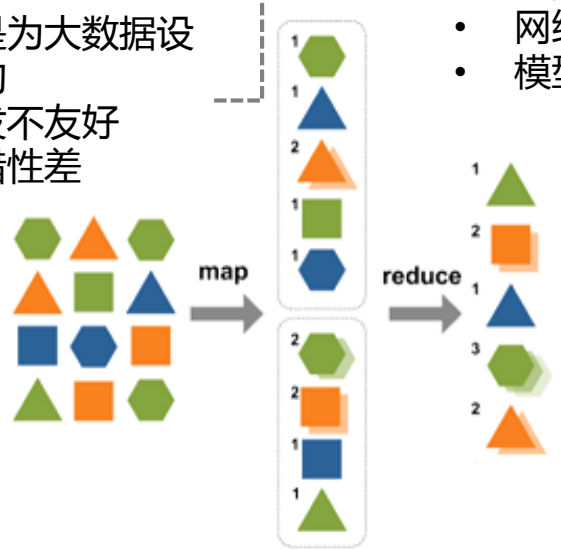
机器学习的并行模式



- 兼容各种并行模式

MPI

- 不是为大数据设计的
- 开发不友好
- 容错性差



- 开发友好

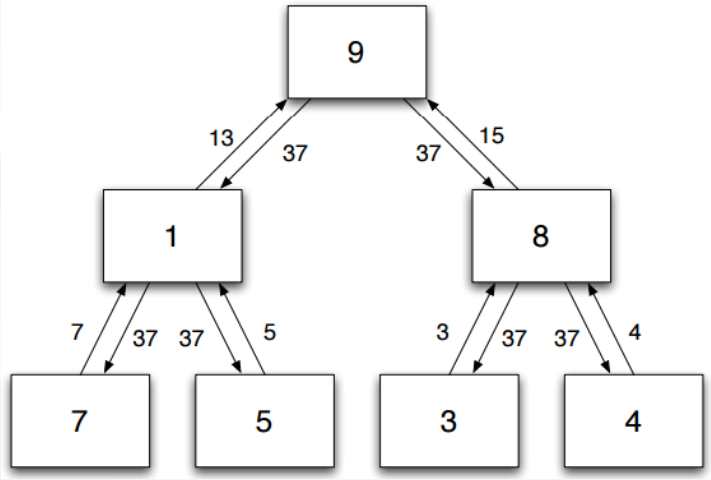
MapReduce

- 同步代价大
- 网络瓶颈大
- 模型规模有限制

- 开发友好

AllReduce

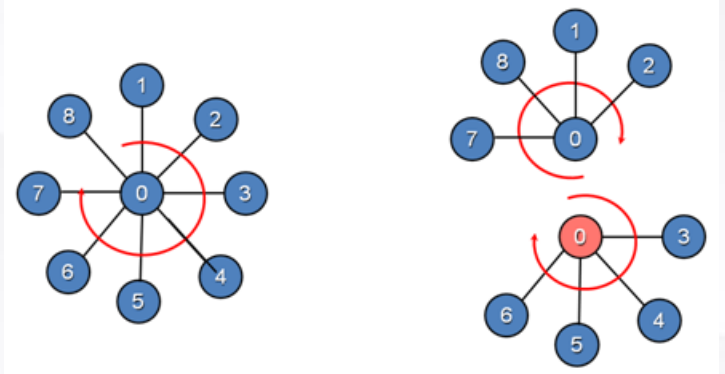
- 同步代价大
- 网络瓶颈较大



- 表达能力强
- 支持大模型
- 比MR灵活

Graph-Base

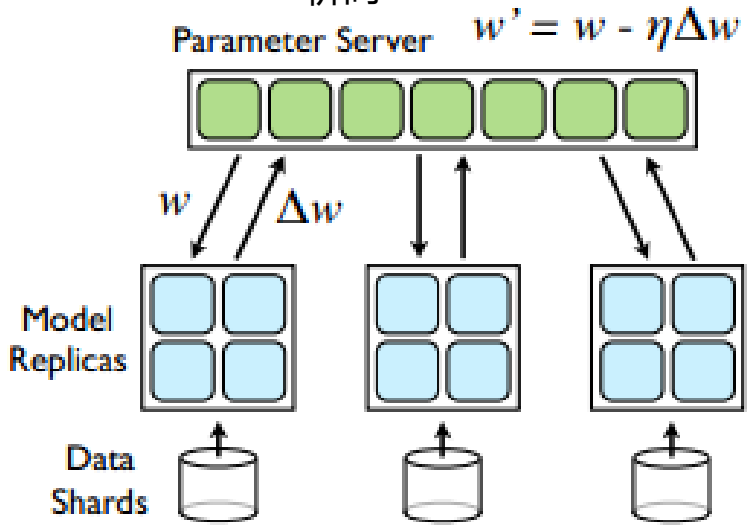
- 学习曲线较高
- 部分算法效率低



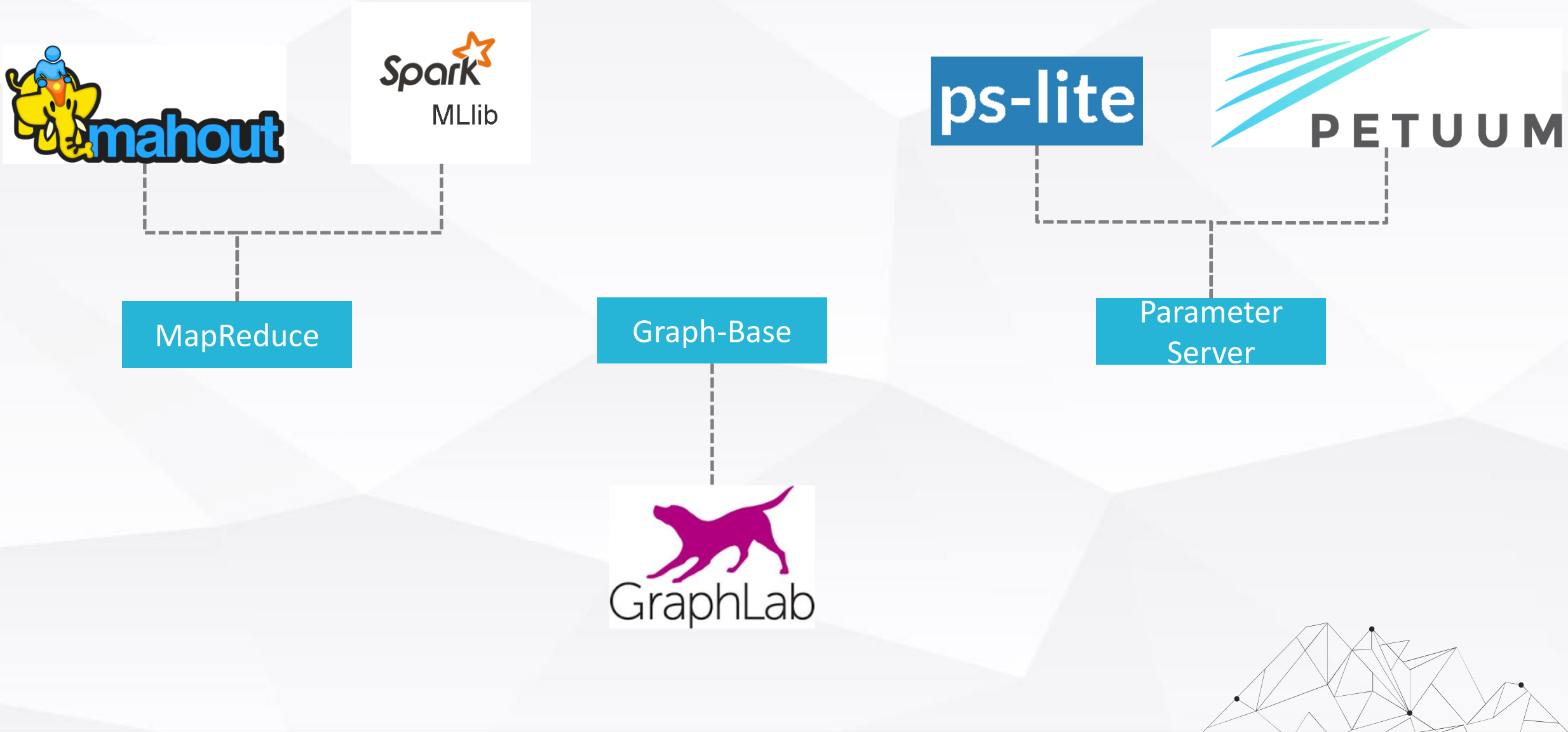
- 支持大模型
- 稀疏数据效率高

Parameter Server

- 稠密数据通信代价高

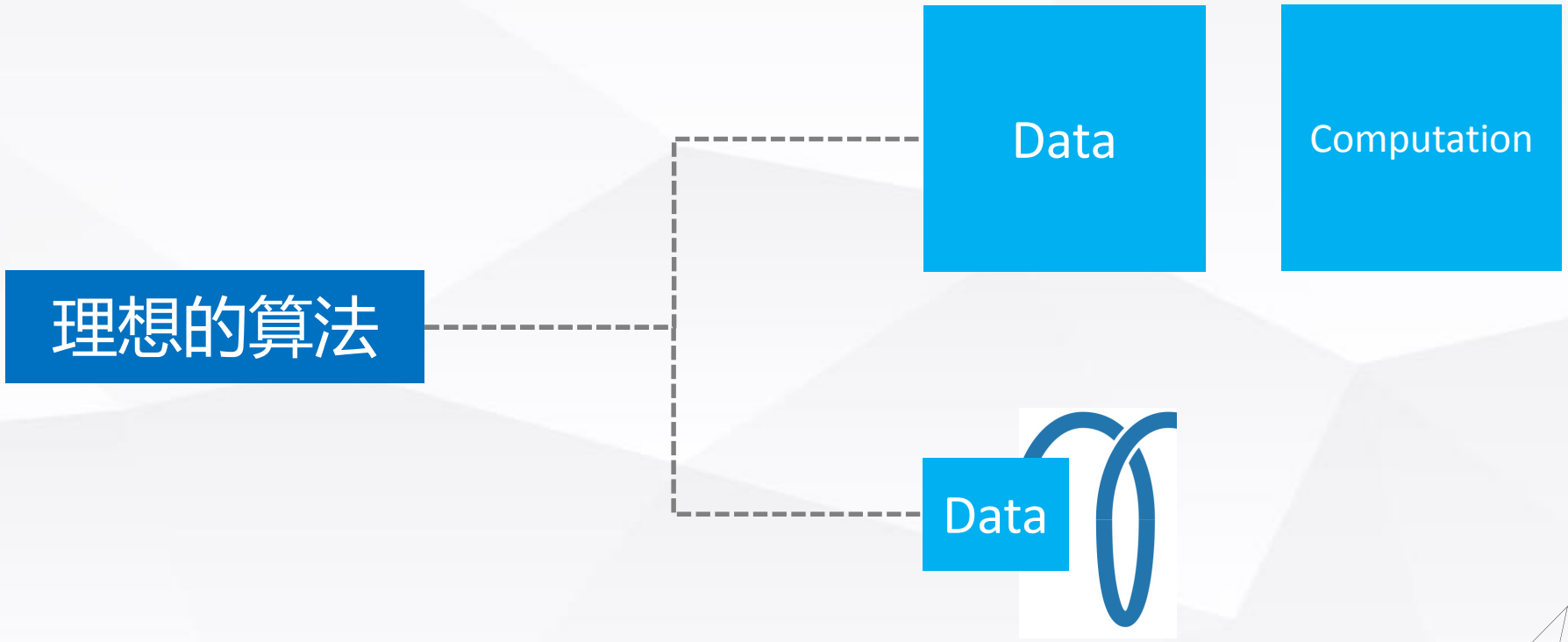


大规模机器学习平台简介

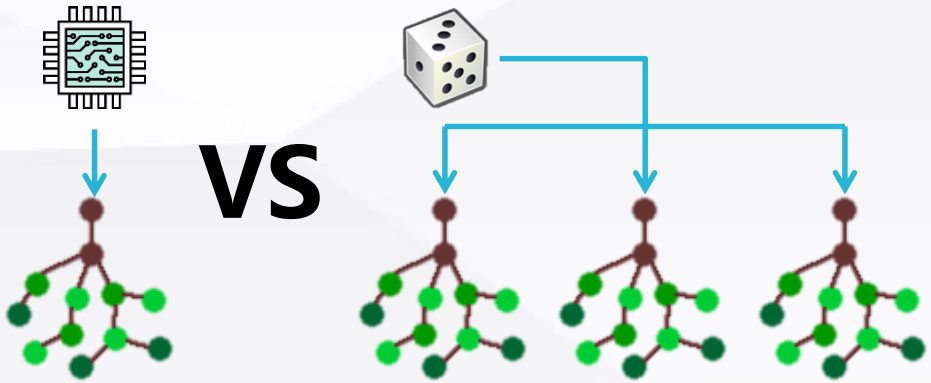


大规模机器学习的反思

仅通过增加计算和内存资源是否能解决计算的瓶颈问题？



随机决策树和随机决策哈希算法



线性算法

比决策树快两个数量级

更精确

更稳定

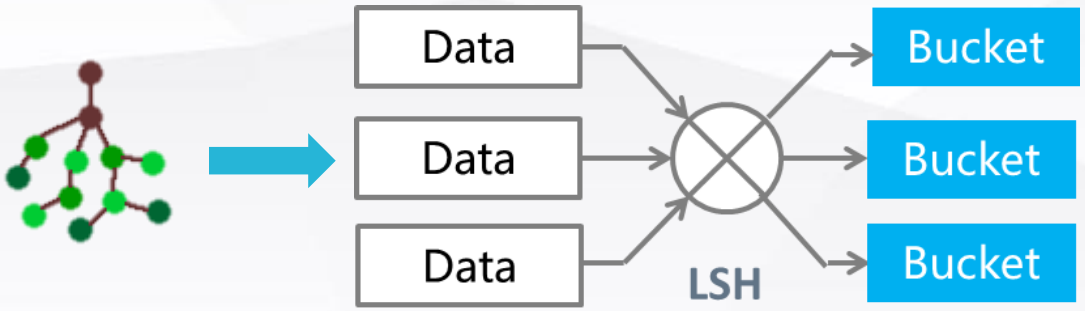
Fan, W., Wang, H., Yu, P. S. and Ma, S. Is random model better? On its accuracy and eciency, IEEE ICDM, 3 (2003).

Xiatian Zhang, Quan Yuan, Shiwan Zhao, Wei Fan, Wentao Zheng, and Zhong Wang, Multi-label classication without the multi-label cost, SDM, 2010.

并行化困难

We found out:

$$\widehat{P}_+(x) \approx P_+(x) + \frac{dh^2 P''_+(x)}{24} + \frac{dh^2}{24a^d}[P'_+(x+a/2) - P'_+(x-a/2)]$$



Xiatian Zhang, Wei Fan, Nan Du, Random Decision Hashing for Massive Data Learning, BigMine 2015 of KDD 2015

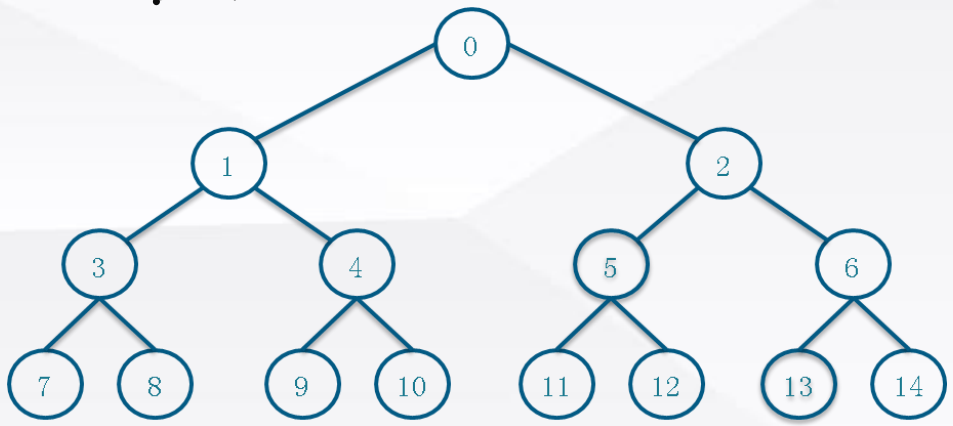
RDT和RDH的精度

Data	RDH	RDT	J48	SMO	LR
a1a	0.881	0.879	0.712	0.760	0.751
a9a	0.886	0.890	0.755	0.761	0.763
mushrooms	1.000	1.000	1.000	1.000	1.000
w1a	0.909	0.953	0.613	0.748	0.732
w8a	0.894	0.997	-	0.797	0.822
splice	0.966	0.909	0.935	0.843	0.853
cod-rna	0.971	0.969	0.944	0.944	0.937
covtype	0.761	0.768		-	0.705
gisette		0.934			-

RDT和RDH的训练时间

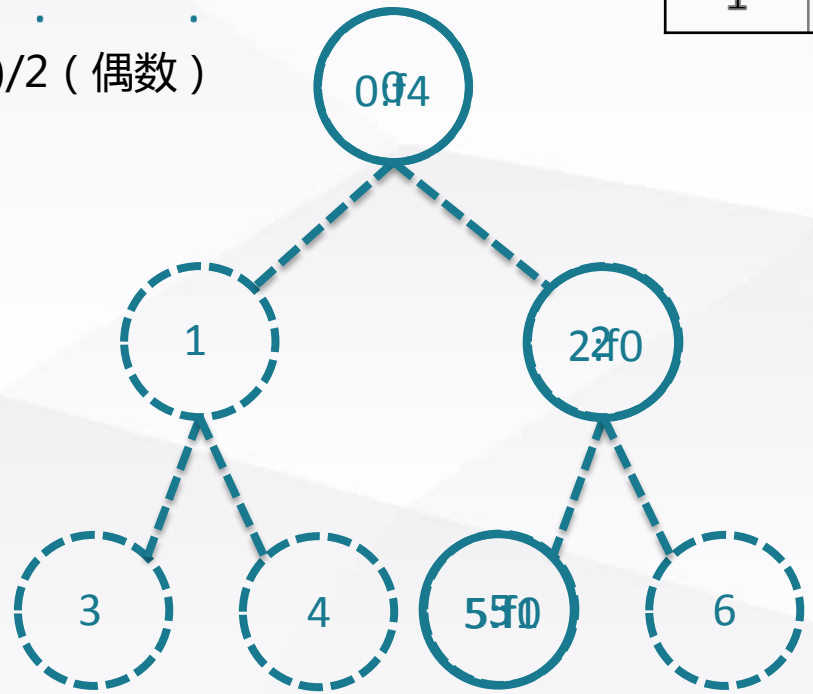
Data	RDH	RDT	J48	SMO	LR
a1a	0.194	0.569	1.861	1.574	1.010
a9a	1.709	24.171	647.013	1637.011	35.901
mushrooms	0.481	3.608	13.651	1.665	3.993
w1a	0.383	0.934	23.022	0.759	6.561
w8a	18.838	33.759	-	487.39	371.836
splice	0.499	0.387	0.770	1.742	0.819
cod-rna	10.933	7.799	155.763	62.705	4.271
covtype	68.545	240.392	-	-	299.667
gisette		82.513			-

RDT的并行化（For Binary Feature Data）

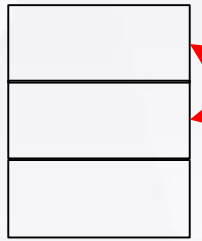


父节点： (p-1)/2(奇数) ， (p-2)/2（偶数）
左子节点：2*p+1
右子节点：2*p+2

f0	f1	f2	f3	f4	f5	f6	f7
1	0	1	0	0	1	0	1

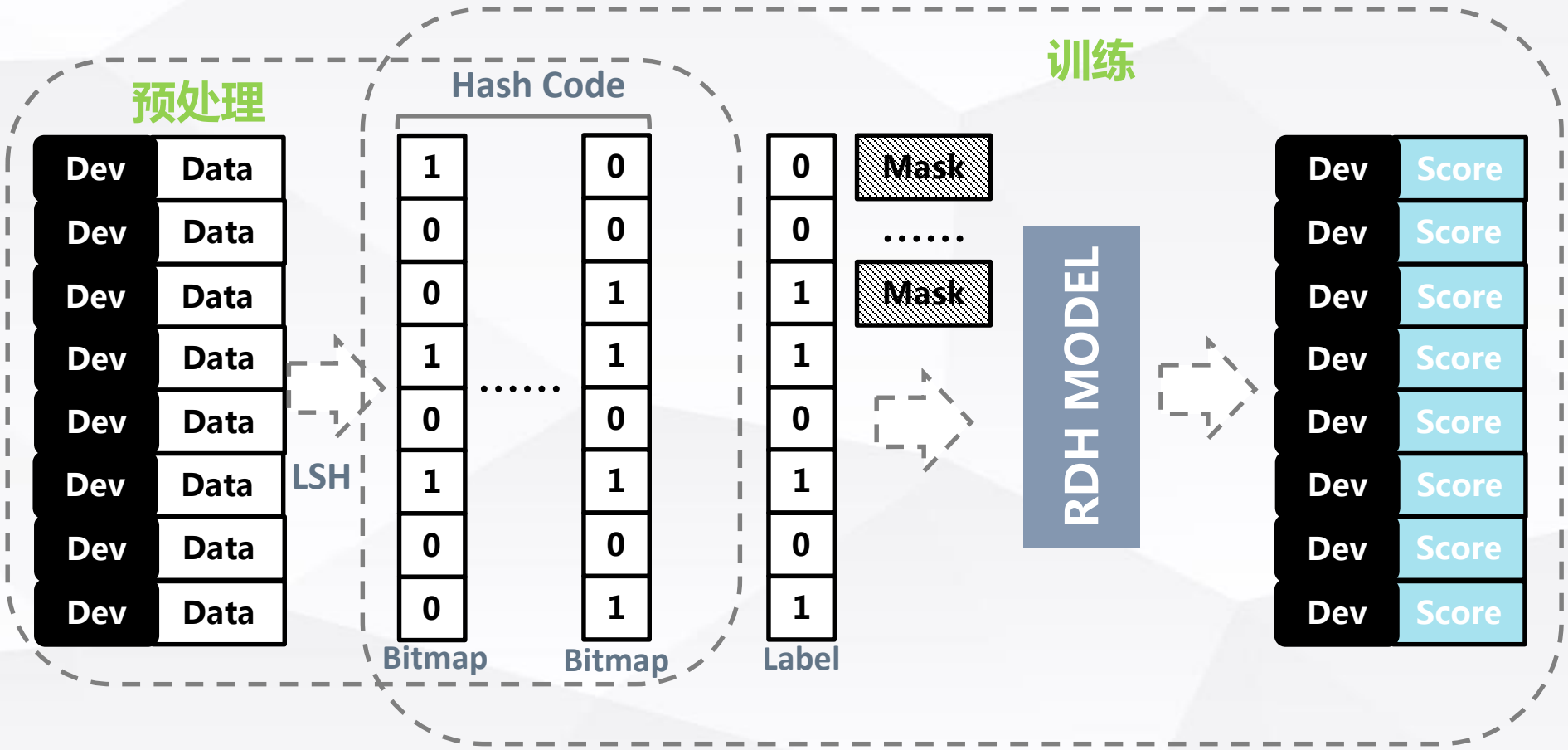


(0+1) mod 8 = 1



Conflict！

RDH的并行化和Bitmap加速



SGD方法

Batch Gradient Decent

$$w := w - \eta \nabla Q(w)$$

敏感参数

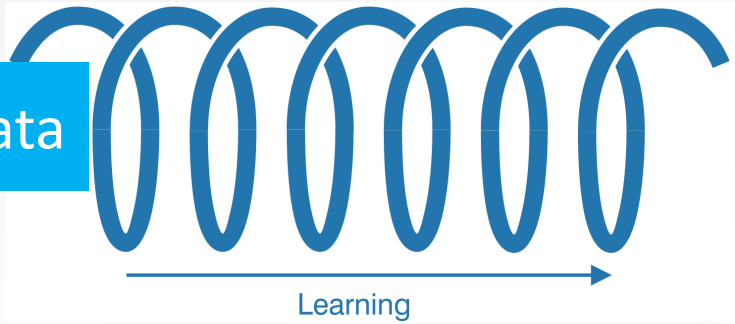
Stochastic Gradient Decent

$$w := w - \eta \nabla Q_i(w)$$

Data

Computation

Data



Data

Computation

SGD方法的优化

无参数一次迭代收敛

利用每一步的梯度信息动态调节学习率，让模型快速收敛

无参数稀疏正则化

在学习过程过根据各系数的重要程度和内存容量动态确定模型稀疏度

SGD的并行化

Spark MLlib

梯度平均

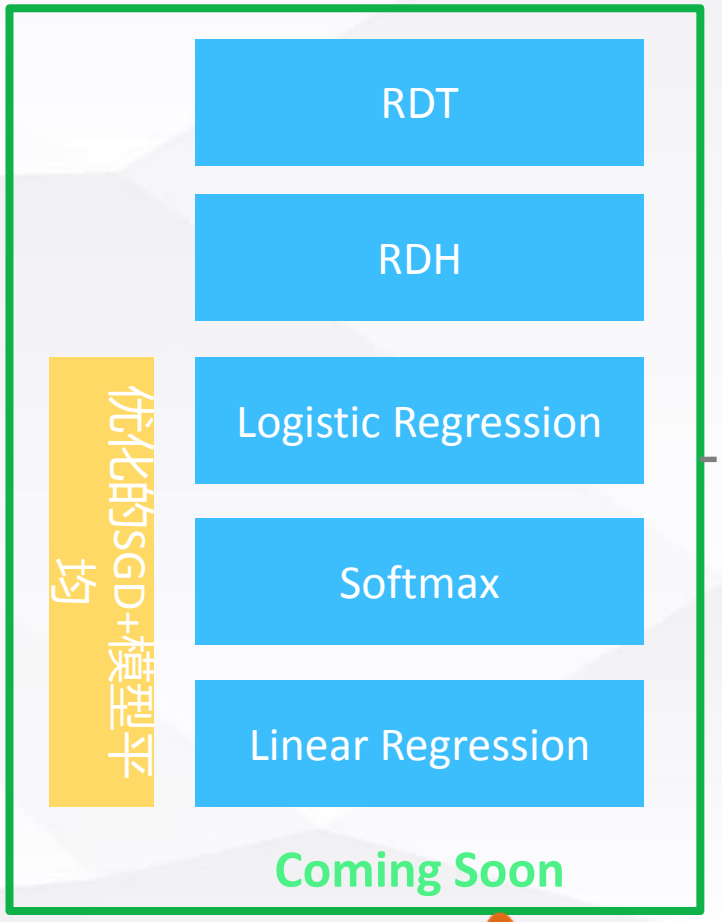
$$w_t = w_{t-1} - \frac{\eta}{n}\sum_{i=0}^{n}\nabla Q_i(w_{t-1})$$

我们的方法

模型平均

$$w_t = \frac{1}{n}\sum_{i=0}^{n} w_{t-1,i}$$

Fregata – TD大规模机器学习算法库

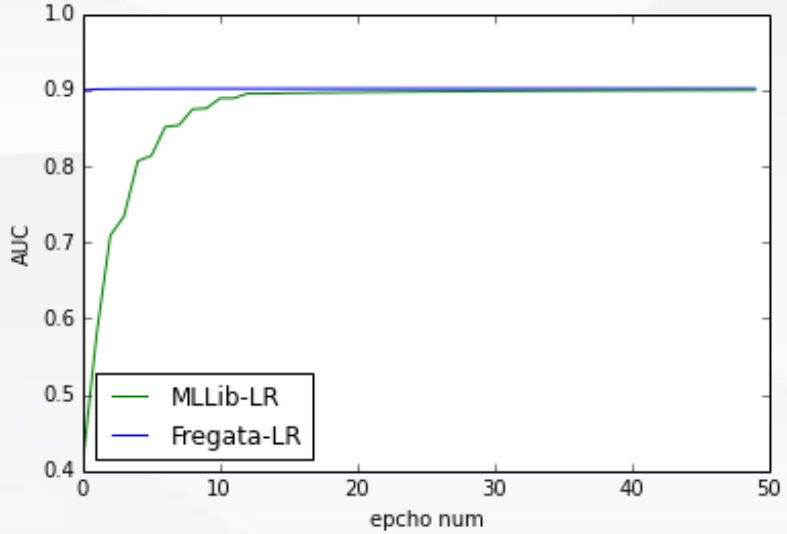


即将开源

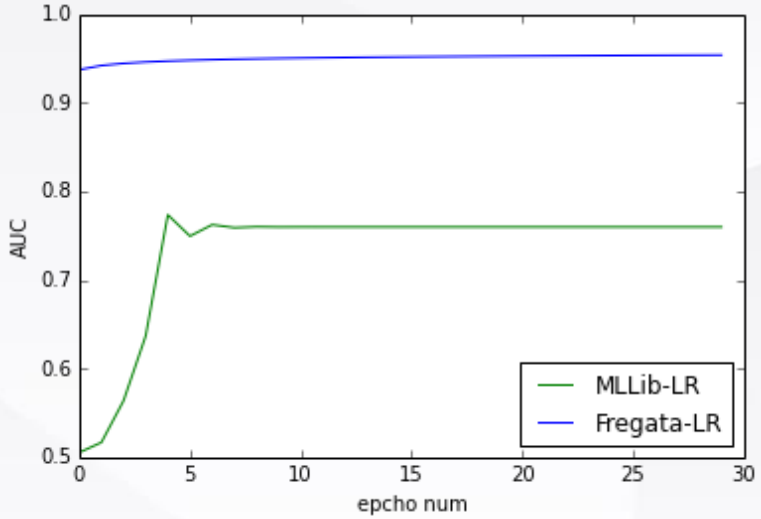
Fregata

Fregata VS MLLib

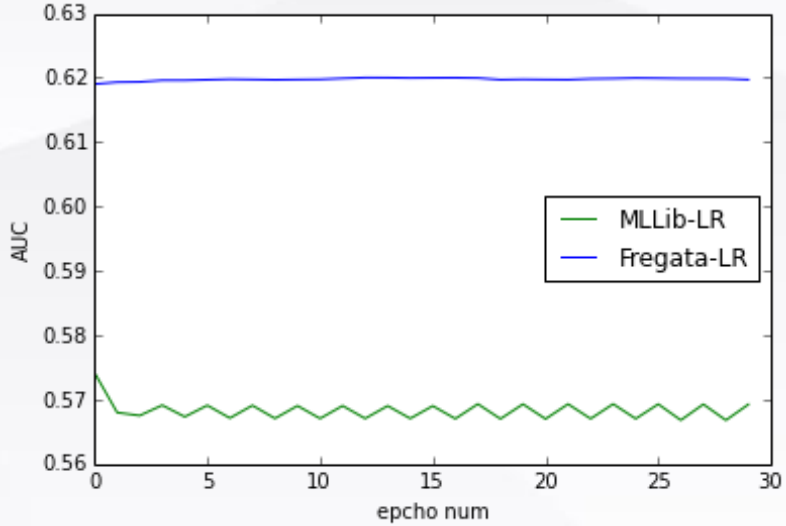
a9a



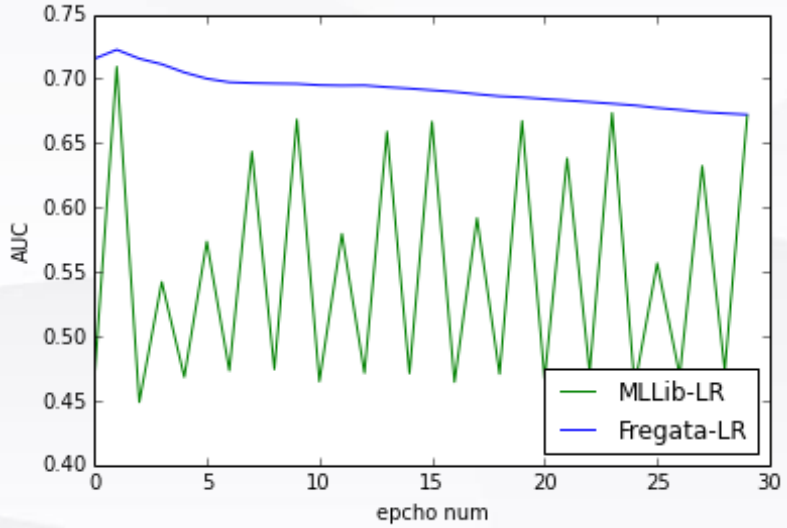
epsilon



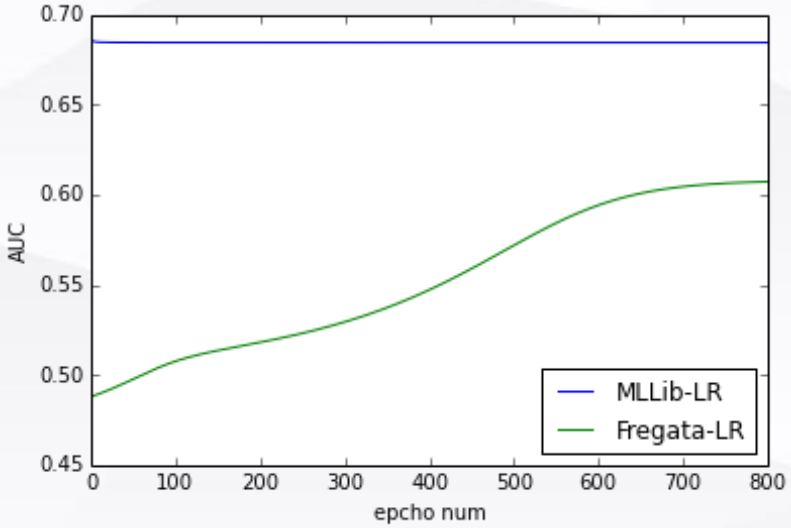
madelon



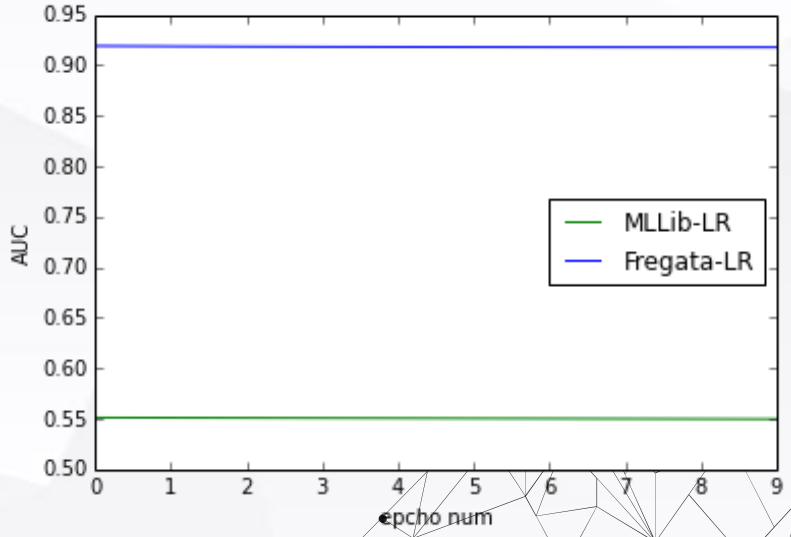
liver-disorders



higgs



lookalike



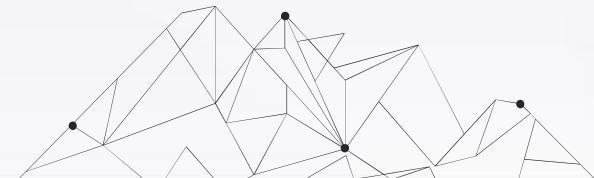
Fregata配置和接口

SBT配置

```
libraryDependencies += "com.talkingdata.datascience.fregata" %% "ml-spark" % "0.1.0"
```

训练代码

```
val conf = new SparkConf().setAppName("test LR")
val sc = new SparkContext(conf)
val trainData = LibSvmReader.read(sc,"/Volumes/takun/libsvm/a9a")
val lr = new LogisticRegression()
val model = lr.train(trainData)
```



Fregata的优势

超大规模

- 10亿样本
- 10亿维度
- 根据内存容量动态确定模型稀疏度

智能训练

- 无需调参
- 一次训练取得最高精度

超高速度

- 内存加速10秒级训练
- 无内存加速分钟级训练
- 比MLLib快1000倍

成熟稳定

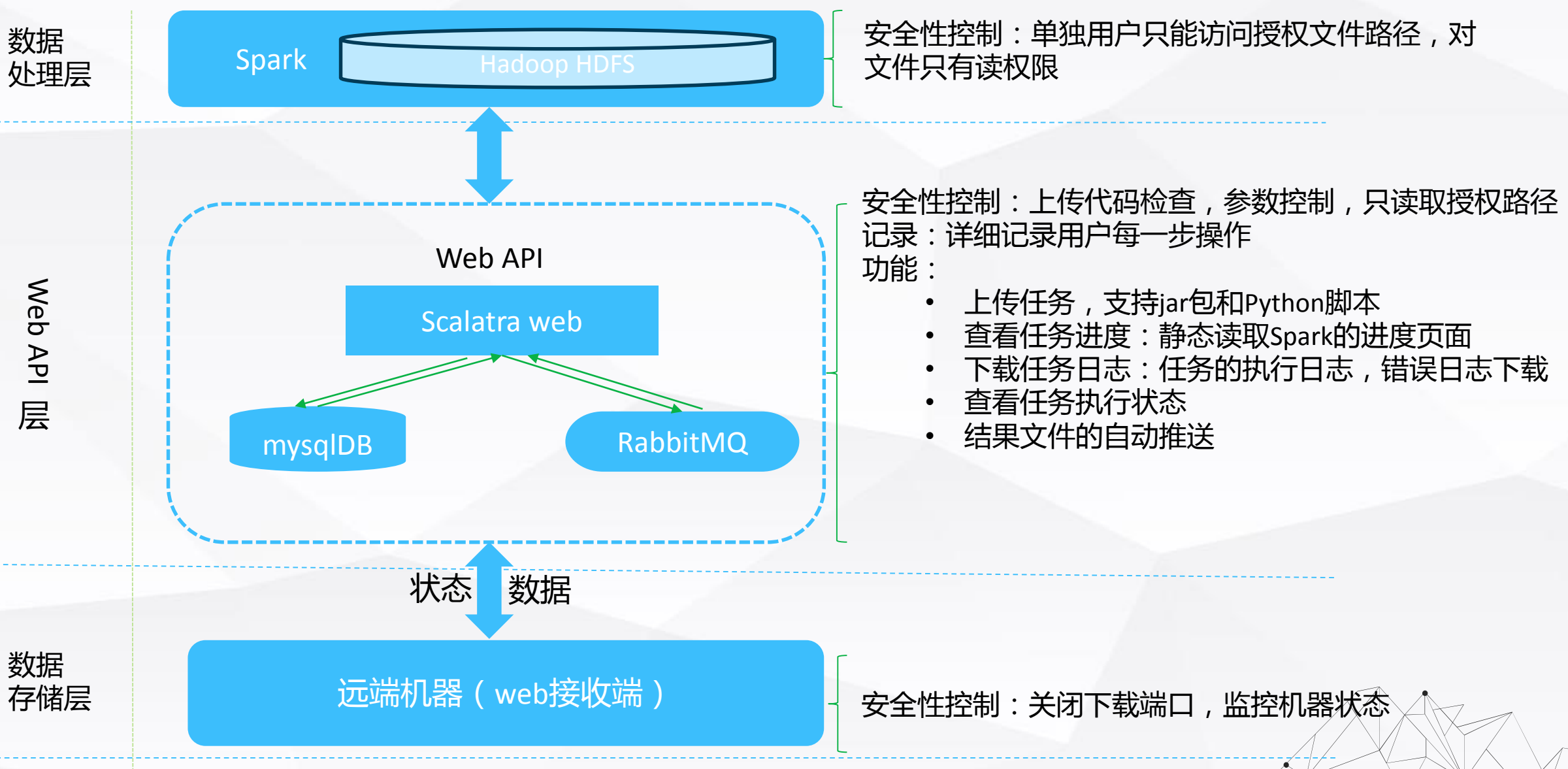
- 基于Spark原生版本开发
- 无缝接入数据处理流程
- 数千次不同任务测试
- 简单易用

超越个人的智慧

开放数据，众智成城

开放数据的权衡

TalkingData数据开放沙箱



成功案例

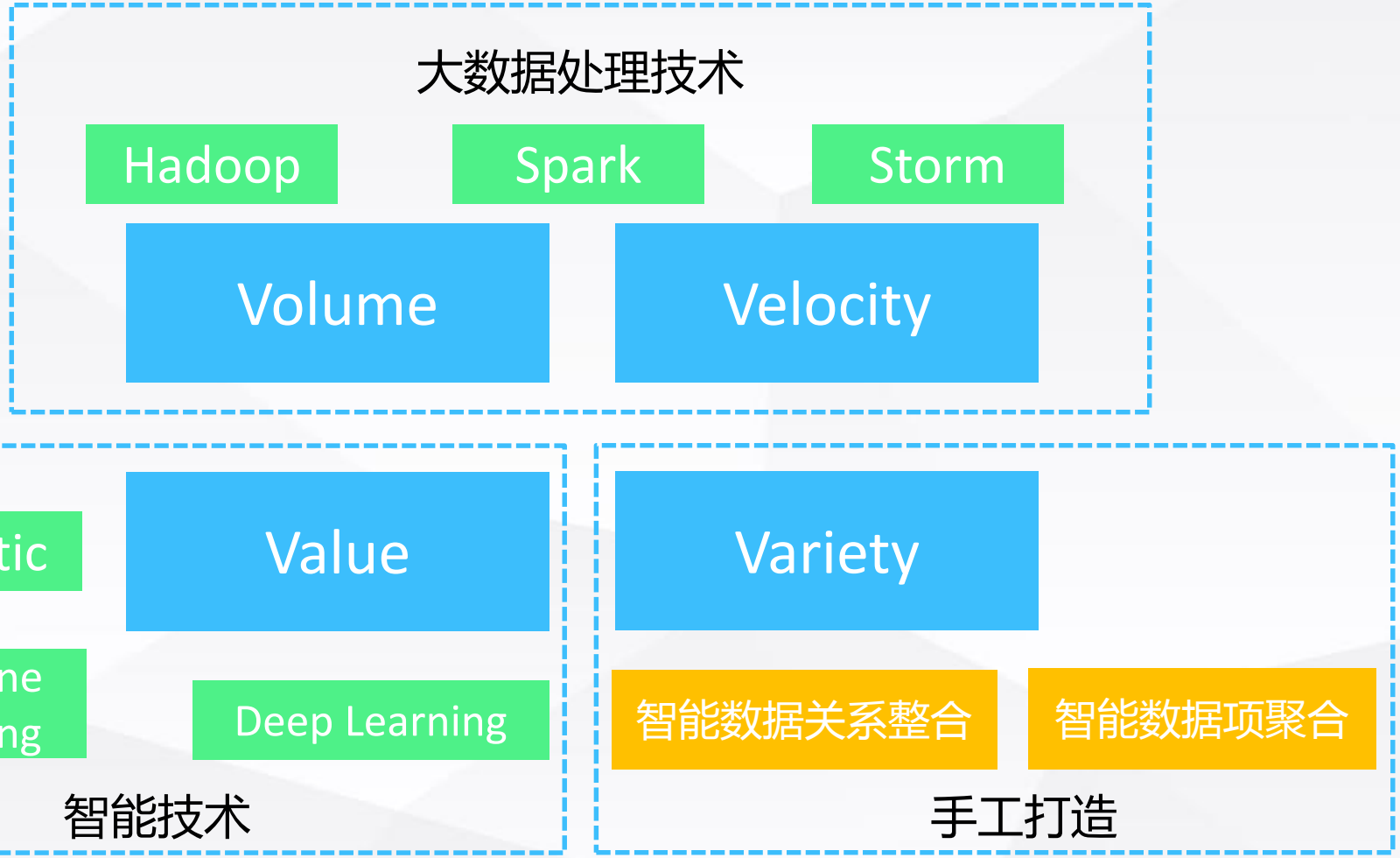
Thasos

通过沙箱服务访问了Talkingdata约8TB数据量。

清华大学大数据实践课程

通过沙箱服务访问了Talkingdata约100GB的数据量。

智能驾驭数据-大数据技术的现状



未来


T11
2016
THANKS
聘
Let’s rock data together!
hr@tendcloud.com
TalkingData