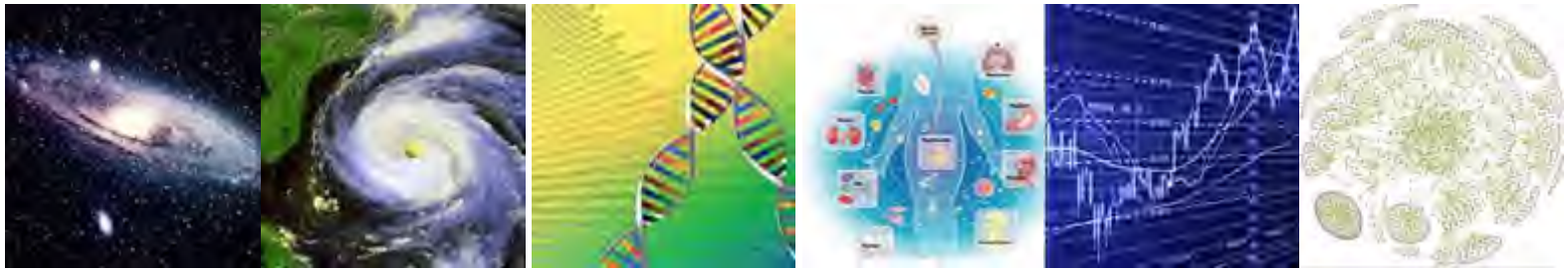
BDTC 2016中国大数据技术大会
Big Data Technology Conference 2016

基于图计算的高性能大数据分析系统 Gemini

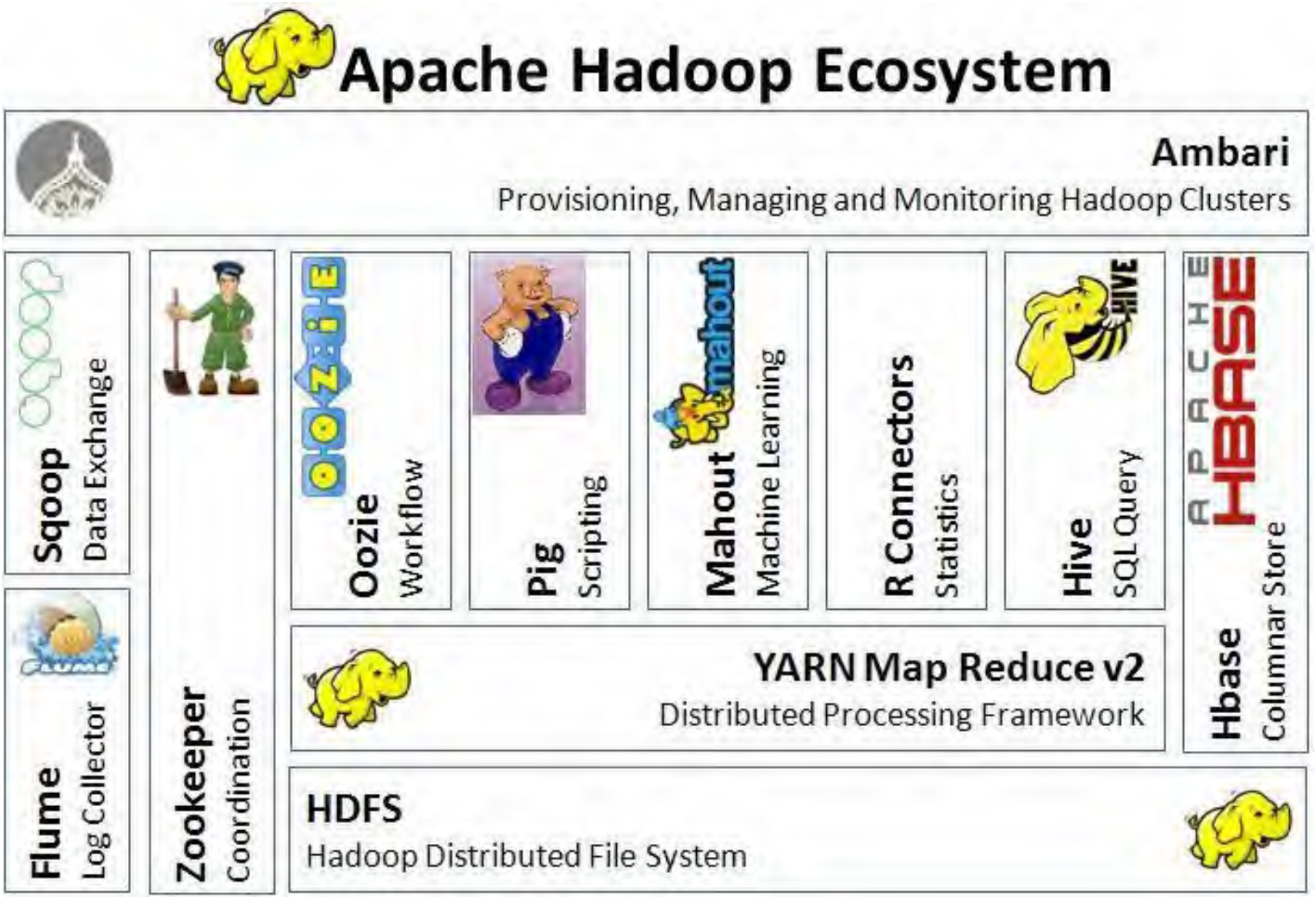
清华大学 陈文光

大数据对分析平台的挑战

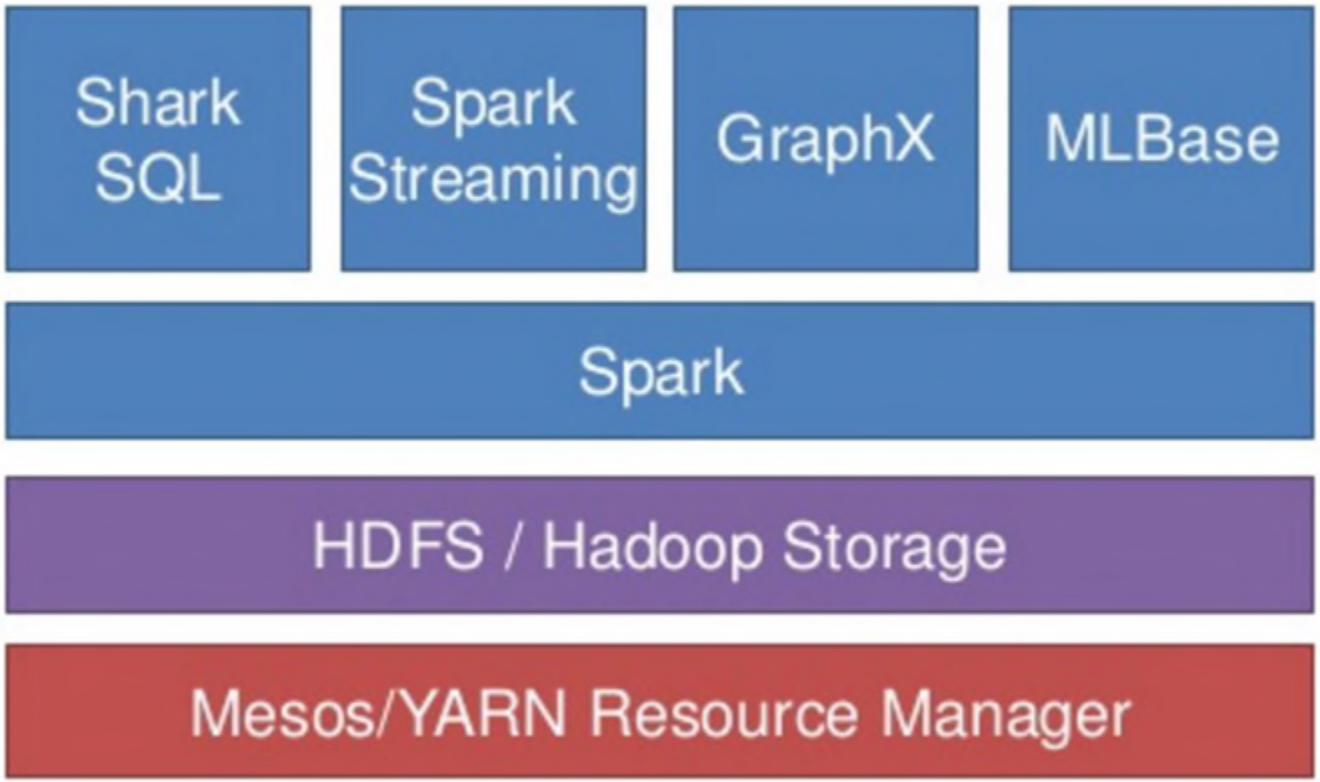
- 大数据是指无法在一定时间内用常规软件工具对其内容进行抓取、管理和处理的数据集合（维基百科定义）
- 大数据 = “海量数据”+“复杂类型的数据”
- 大数据的特性（ Volume，Variety，Velocity）
  - 数据量大：PB、TB、EB、ZB级别的数据量
  - 种类多：包括文档、视频、图片、音频、数据库、层次状数据等
  - 速度快：数据生产速度很快；要求对数据处理和I/O速度很快



主流大数据平台 - Hadoop



基于内存的大数据分析平台 -Spark

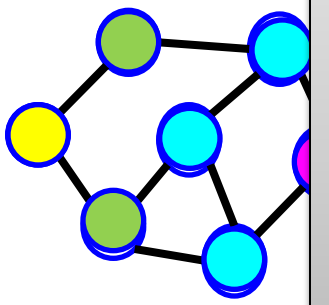


Spark的局限性-数据模型层面

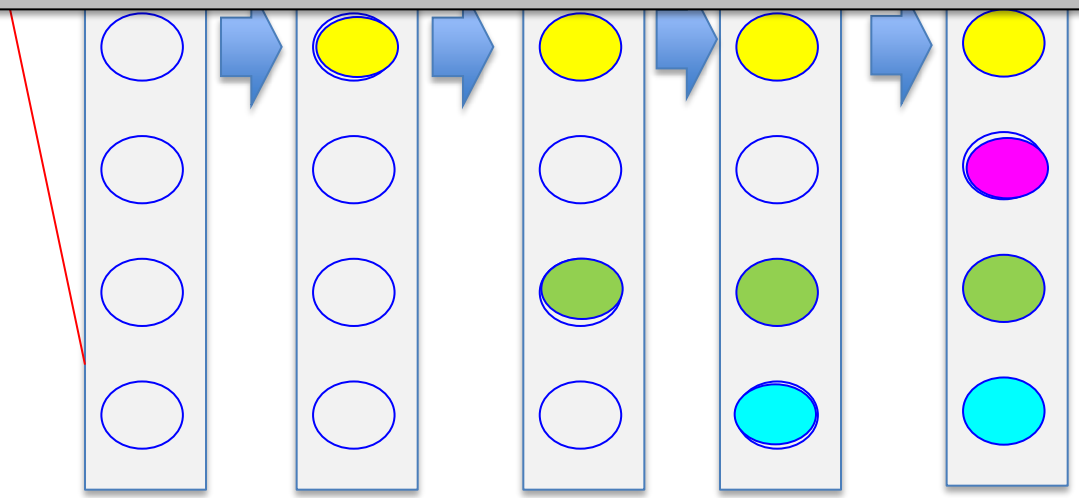
大数据应用：

部分数据更

图遍历（BFS）



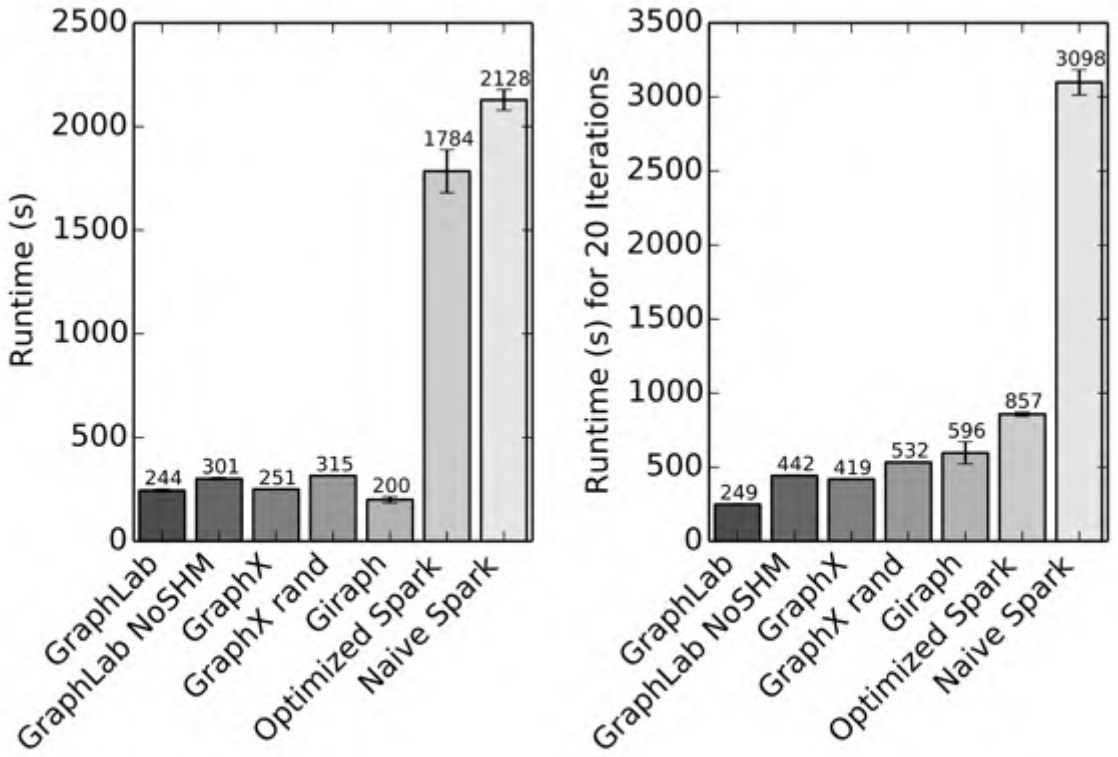
每次细粒度的数据更新，由于spark基于粗粒度RDD只读的数据对象模型，需要RDD变换，即有大量数据的复制，导致处理效率不高。



Spark的局限性-实现层面

- Spark基于Scala语言，运行在JVM上
- 内存表示冗余，占用内存大
- 内存分配与回收开销大

GraphLab在某些任务上比Spark快10倍



(a) Conn. Comp. Twitter (b) PageRank Twitter

Gonzalez, Joseph E., et al. "Graphx: Graph processing in a distributed dataflow framework." Proceedings of OSDI. 2014.

图计算 – 折衷的大数据分析平台

MPI，OpenMP
- 可读写的数据
- 容错困难
- 不支持自动负载平衡

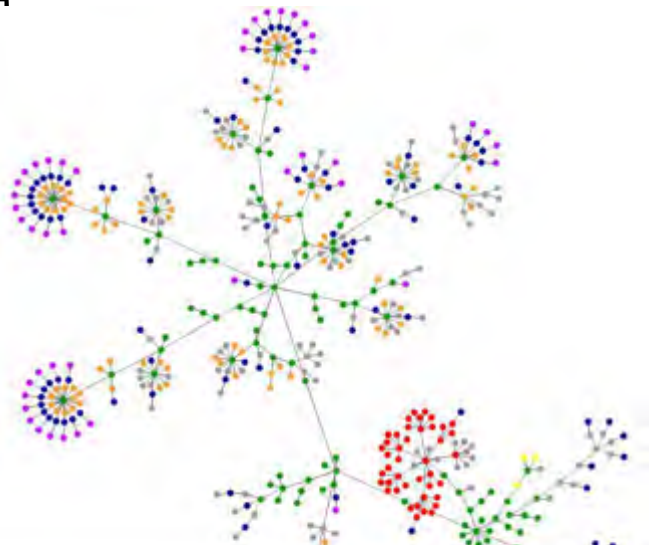
GraphLab,Gemini
- 可读写的数据
- 容错性能较好
- 一定程度的自动负载平衡

MapReduce，Spark
- 只读数据集
- 容错方便，扩展性好
- 自动负载平衡

性能　　扩展性

图数据的重要意义

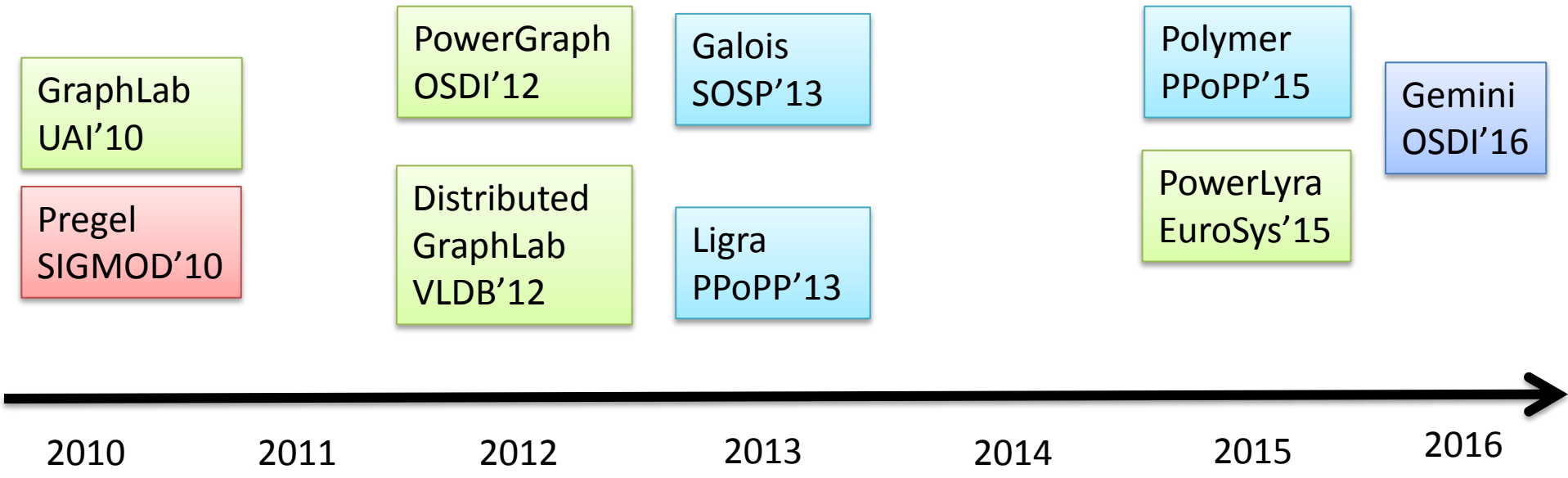
- 图能够表达丰富的数据和关系
  - 网络连接
  - 网页链接
  - 社交关系
  - 蛋白质交互
  - 人与人，人与公司，人与产品



图的计算与分析

- PageRank
- 最短路径
- 连通分支
- 极大独立集
- 最小代价生成树
- Bayesian Belief Propagation
- …

代表性图计算系统


GraphLab
UAI'10
Pregel
SIGMOD'10
PowerGraph
OSDI'12
Distributed
GraphLab
VLDB'12
Galois
SOSP'13
Ligra
PPoPP'13
Polymer
PPoPP'15
PowerLyra
EuroSys'15
Gemini
OSDI'16
2010
2011
2012
2013
2014
2015
2016

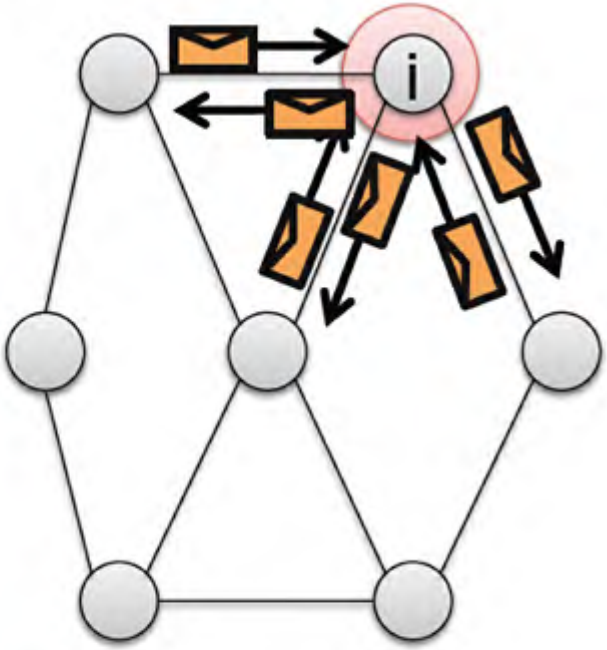
The Pregel Abstraction

Vertex-Programs interact by sending **messages**.

```
Pregel_PageRank(i, messages) :
  // Receive all the messages
  total = 0
  foreach( msg in messages) :
    total = total + msg

  // Update the rank of this vertex
  R[i] = 0.15 + total

  // Send new messages to neighbors
  foreach(j in out_neighbors[i]) :
    Send  msg(R[i] * w_ij) to vertex j
```



PowerGraph/PowerLyra的问题

- 计算性能低，处理小图时8台机器性能还不如单机系统

System	PageRank	ConnComp
Ligra [29]	44.1	7.46
Galois [21]	19.0	11.5
Polymer [34]	43.1	7.14
PowerGraph [11]	40.3	29.1
GraphX [12]	216	104
PowerLyra [9]	26.9	22.0

twitter-2010数据集上，20 轮PageRank迭代(41.7M 结点, 1.47B 边)

性能数据对比

瓶颈在计算！

结点数 系统	1 Galois	8 PowerLyra
运行时间 (s)	19.3	26.9
指令数	分布式计算开销	
内存访问数		
通信量(GB)	-	38.1
IPC	0.414	0.655
L3 缺失率	计算不够优化	
CPU 利用率		

网络带宽远远没有饱和
(100Gbps)
(38.1*8/2/26.9/8=**0.708**Gbps)

局部性差

CPU利用率低

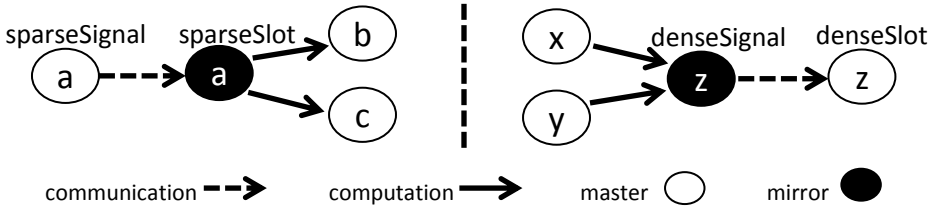
twitter-2010数据集上，
20 轮PageRank迭代
(41.7M 结点, 1.47B 边)

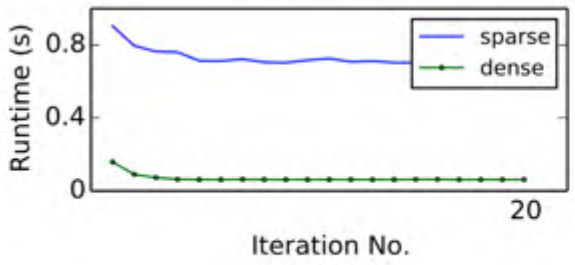
分布式图计算系统Gemini

- 在高效性的基础上支持扩展性
  - 避免没有必要的“分布式”副作用
  - 优化图的划分与计算
- 设计理念的变化
  - 以计算性能为中心的分布式系统
    - 分布式系统有快速的通信网络
    - 计算可以与通信重叠
  - 效率优化
    - 自适应push-pull转换
    - 层次化的分块划分
  - 扩展性优化
    - 局部性感知的分块
    - 基于分块的任务窃取

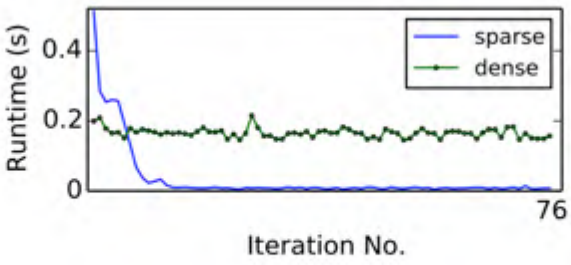
稠密-稀疏双模式的计算模型

- 图计算中的活跃结点数在不同迭代步骤时不同
  - 活跃结点多，适合稠密模式
  - 活跃结点少，适合稀疏模式

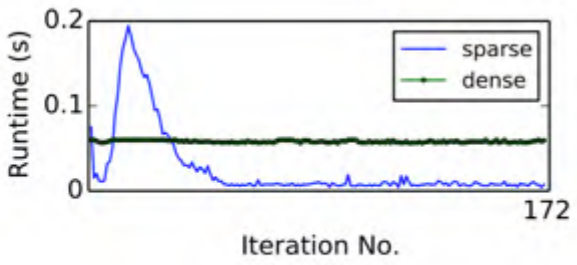




(a) PR



(b) CC

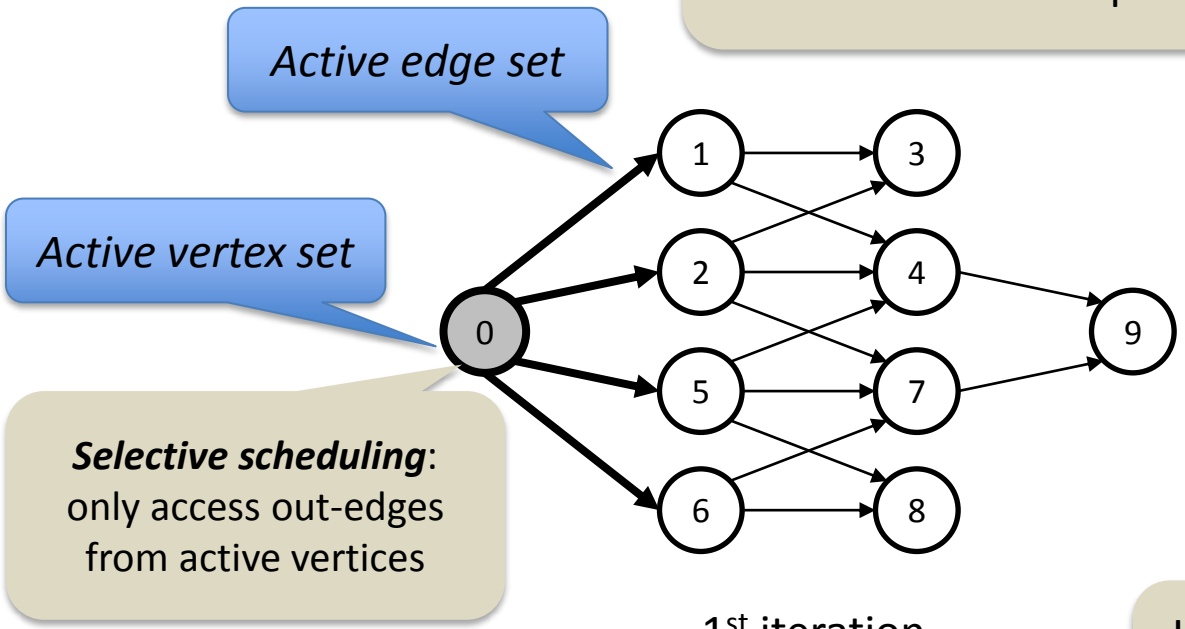


(c) SSSP

双模式: 以BFS 为例 (1)

Dual mode updates proposed in shared-memory systems (Ligra[PPoPP '13])

$|Active\ edge\ set| / |E| < threshold$
$\Rightarrow$ ***Sparse*** *mode*
$\Rightarrow$ ***Push*** operations

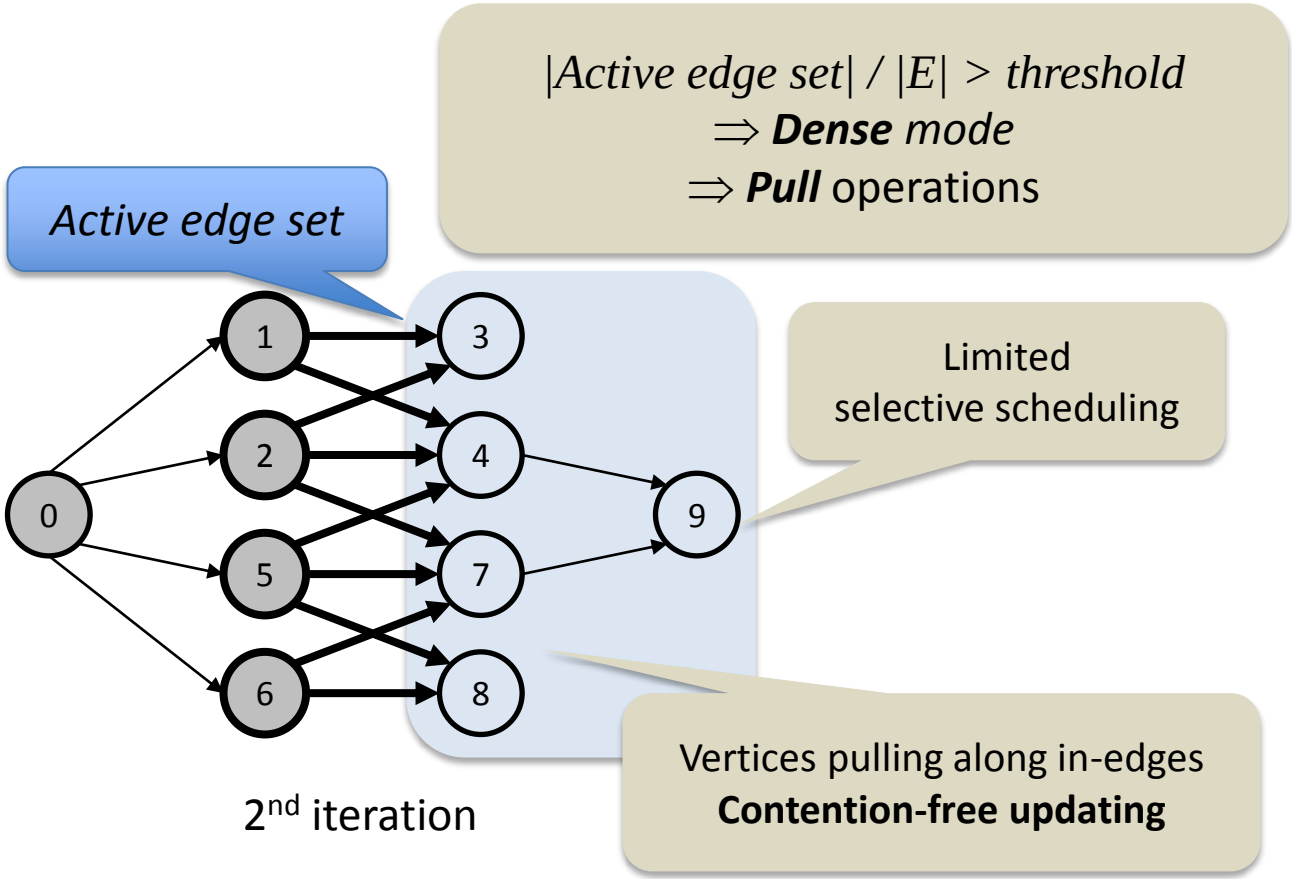


Selective scheduling: only access out-edges from active vertices

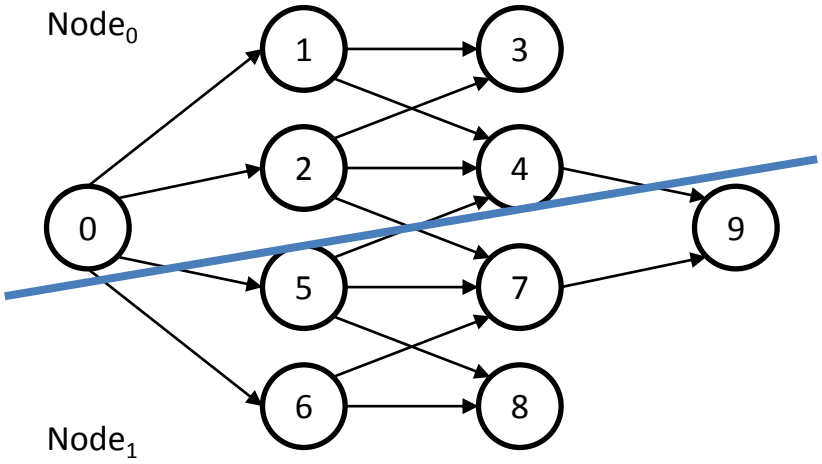
1st iteration

Locks/atomic operations required for correctness of concurrent updates

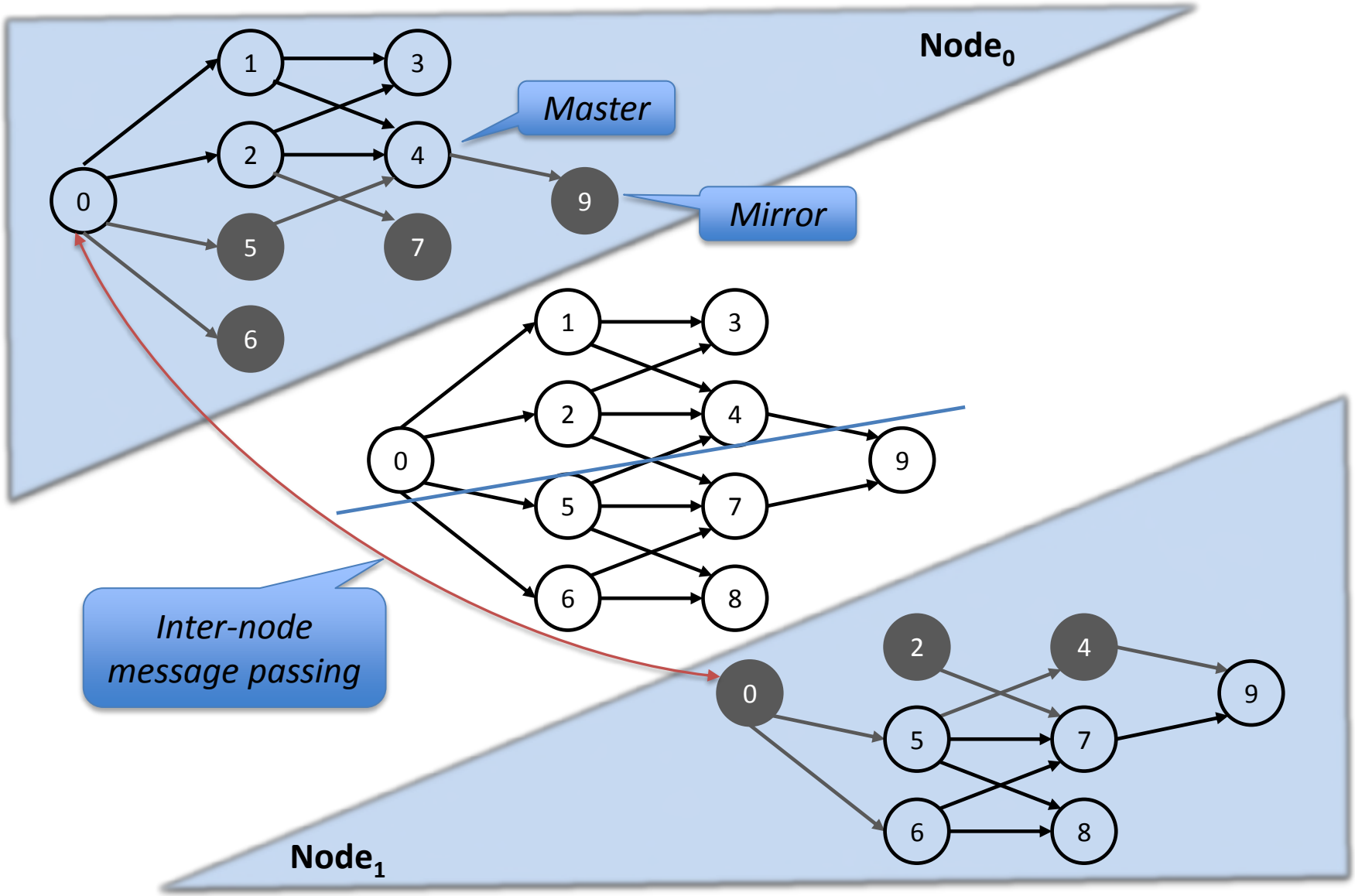
双模式: 以BFS 为例 (2)


|Active edge set| / |E| > threshold
⇒ Dense mode
⇒ Pull operations
Active edge set
1
2
3
4
5
6
7
8
0
9
Limited
selective scheduling
Vertices pulling along in-edges
Contention-free updating
2nd iteration

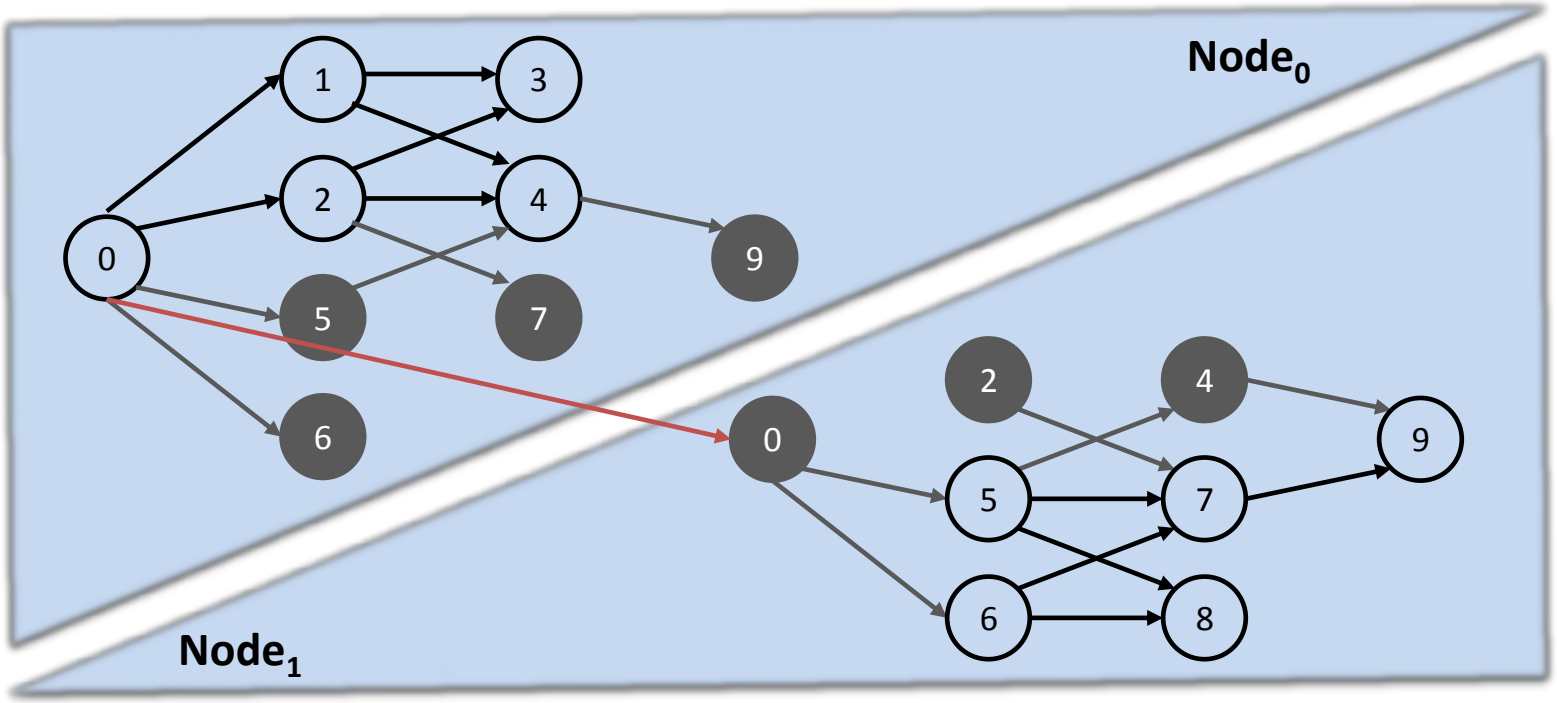
分布式双模式计算



分布到两个节点

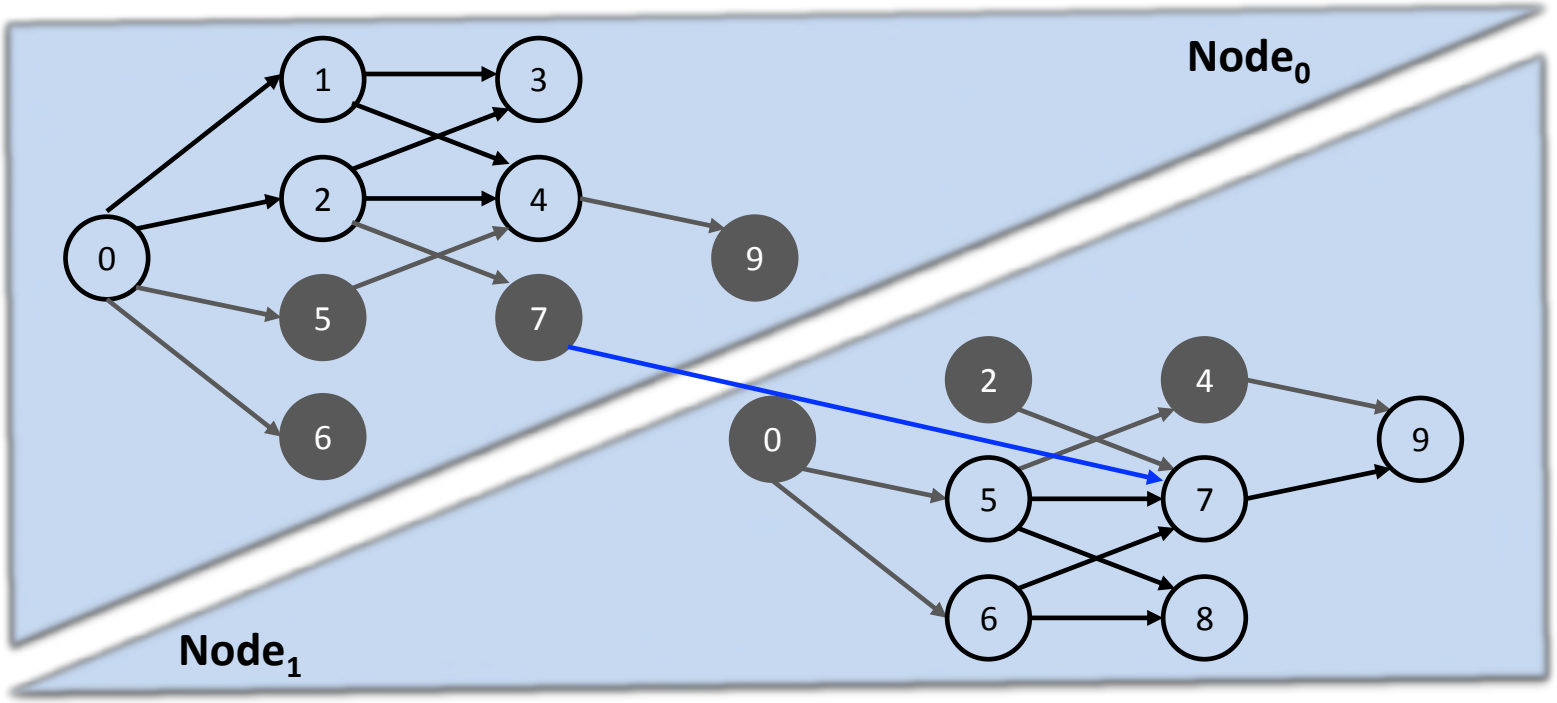

Node0
Master
Mirror
Inter-node
message passing
Node1
0
1
2
3
4
5
6
7
8
9

Gemini的分布式push



Masters message mirrors, who update their local neighbors

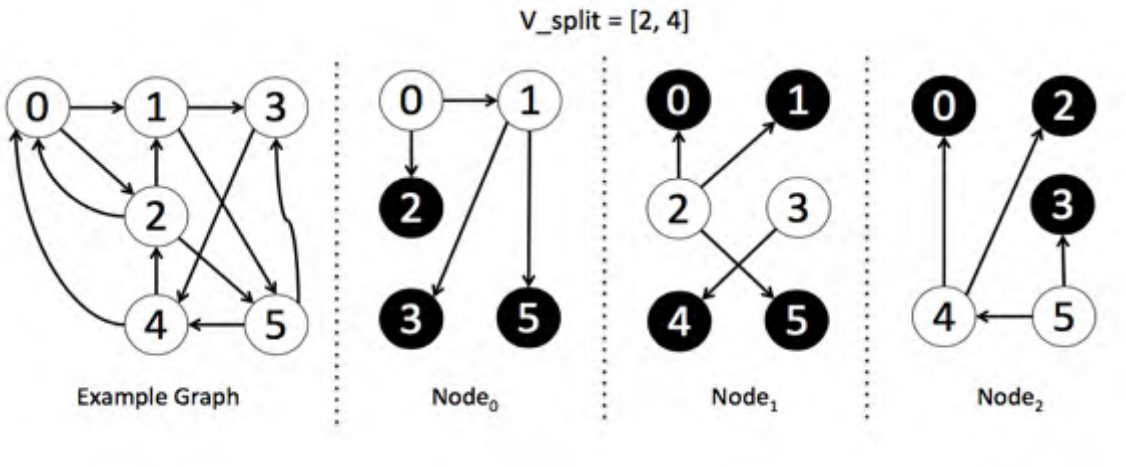
Gemini的分布式Pull



Mirrors pull updates from neighbors, then message masters

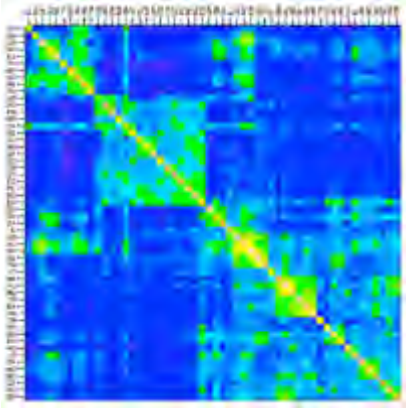
基于chunk的图划分方法

- 传统图划分方法
  - 代价高：metis
  - 划分效果差：hash
- 基于chunk的划分
  - 利用数据集中的局部性

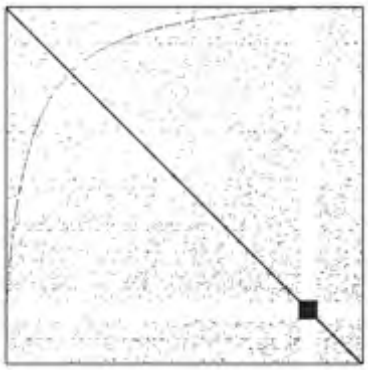


为什么做chunk划分?

- chunk划分保留了局部性!
  - **很多实际图中都存在局部性**
    - E.g., WebGraph[WWW '04], BLP[WSDM '13]
    - 图结点按“语义”排序



Facebook Country Adjacency Matrix[1]



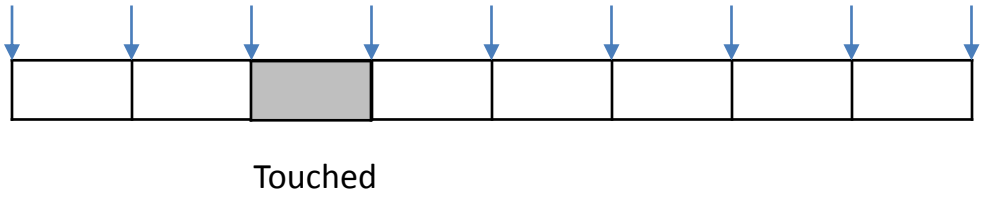
UK Web (2005) Adjacency Matrix

  - 结点没有排序时存在可接受的预处理方法
    - E.g., BFS[Algorithms 09], LLP[WWW '11]

[1] The Anatomy of the Facebook Social Graph

Chunk的其它好处

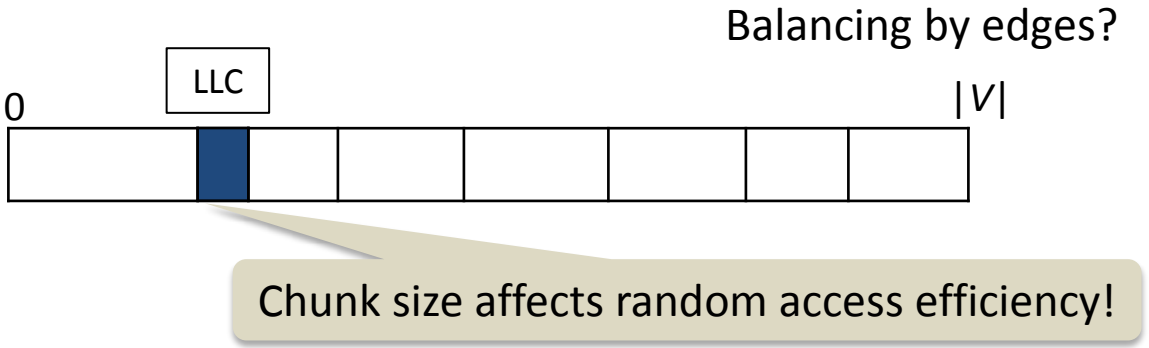
- 不需要转换vertex IDs (global ⇔ local)
  - 大大缩小了partition信息的维护开销
    - O(*p*) chunk 边界
  - 结点数据更容易管理
    - 在共享内存中分配连续的数据



- 可以在多个层次递归地使用

局部性感知的Chunking

- 如何分块?



- Gemini 同时考虑了结点和边
  - 边数: 处理的工作量
  - 结点数局部性
  - 混合度量: $\alpha \cdot |V_i| + |E_i|$
    - α 现在设为 8(p-1)

多层次分块划分和任务窃取

Per-node partition

cluster

Per-socket partition w. NUMA-aware placement

node

Data partitioning till socket-level

Per-core work chunk

socket

core

Mini-chunk for work stealing

Work partitioning within socket

Locality-aware

Fine grain (64-vertex)

性能评估

- 平台: 8-结点集群
  -  Intel Xeon E5-2670 v3 (12-core CPU), 30MB L3 cache
  -  2 sockets sharing 128 GB RAM (DDR4 2133MHz)
  -  Network: Mellanox Infiniband EDR 100Gbps

- 测试程序
  - PageRank (PR) (20 iterations)
  - Connected Components (CC)
  - Single-Source Shortest Paths (SSSP)
  - Breadth-First Search (BFS)
  - Betweenness Centrality (BC)

- 输入图

Graph	|V|	|E|
enwiki-2013	4,206,785	101,355,853
twitter-2010	41,652,330	1,468,365,182
uk-2007-05	105,896,555	3,738,733,648
weibo-2013	72,393,453	6,431,150,494
clueweb-12	978,048,098	42,574,107,469

单结点效率

Application	Ligra	Galois	Gemini
PR	21.2	19.3	**12.7**
CC	6.51	**3.59***	4.93
SSSP	**2.81**	3.33	3.29
BFS	**0.347**	0.528	0.468
BC	2.45	3.94*	**1.88**

Runtime in seconds (twitter-2010)

More iterations

More instructions

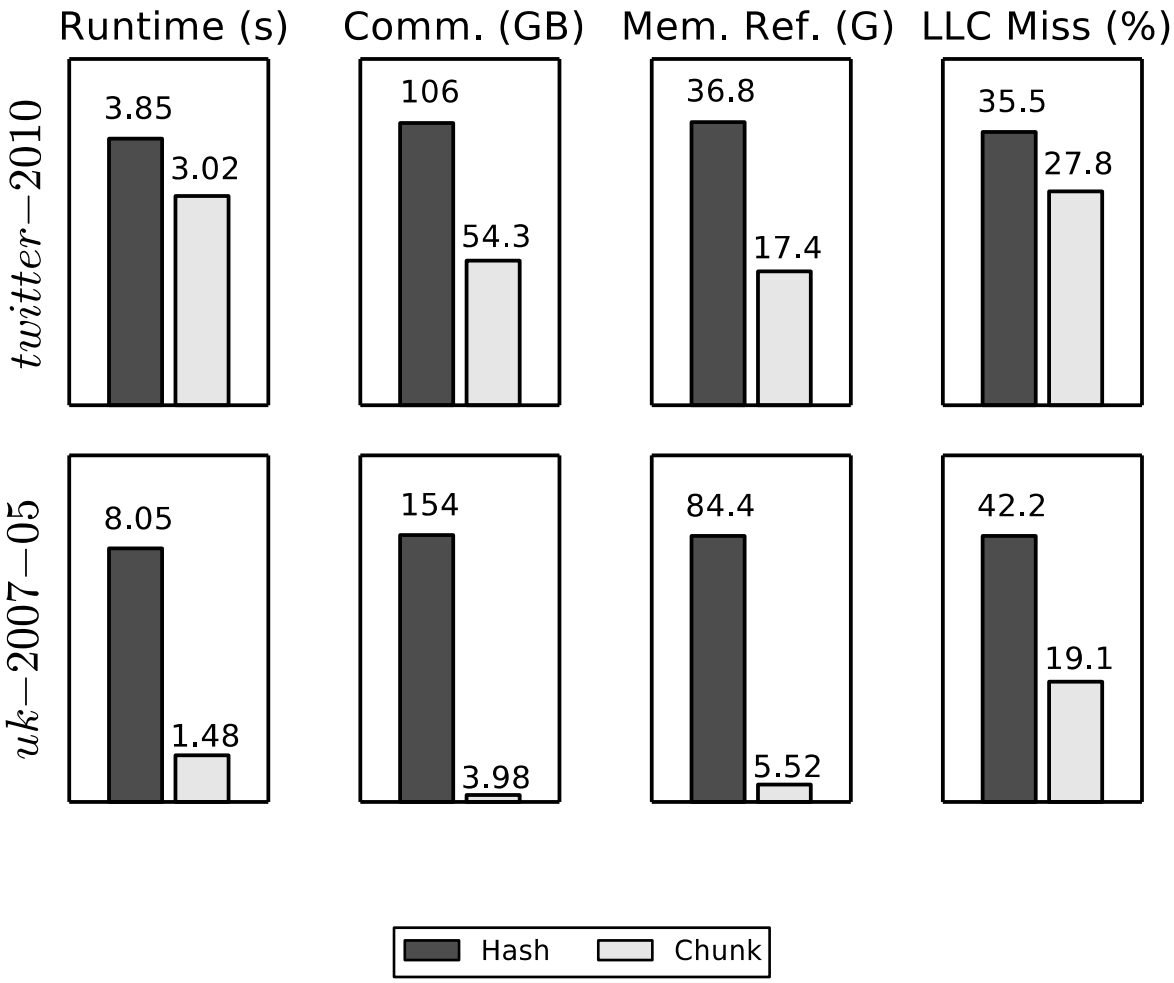
NUMA-aware memory accesses

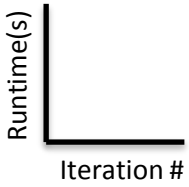
System	Ligra	Gemini
Remote access ratio	50.1%	9.10%
L3 cache miss rate	52.6%	40.1%
Average access latency	183ns	125ns

Memory reference profiling results

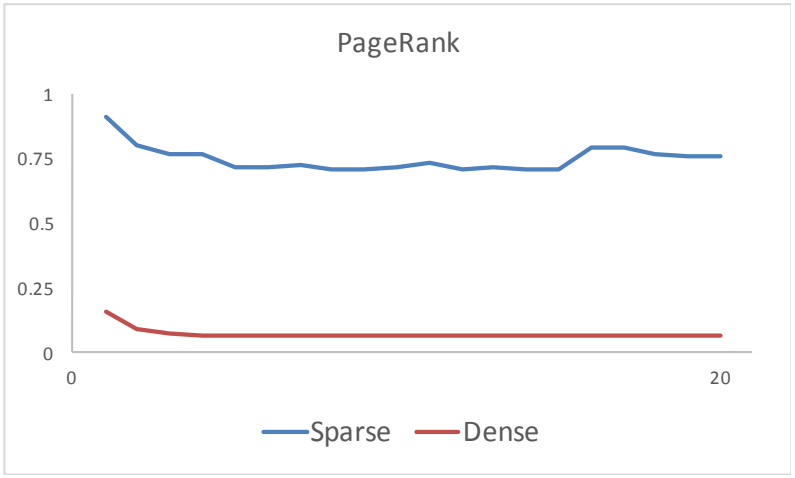
"*" uses different algorithms.

基于chunk的分块方法和基于Hash的划分方法

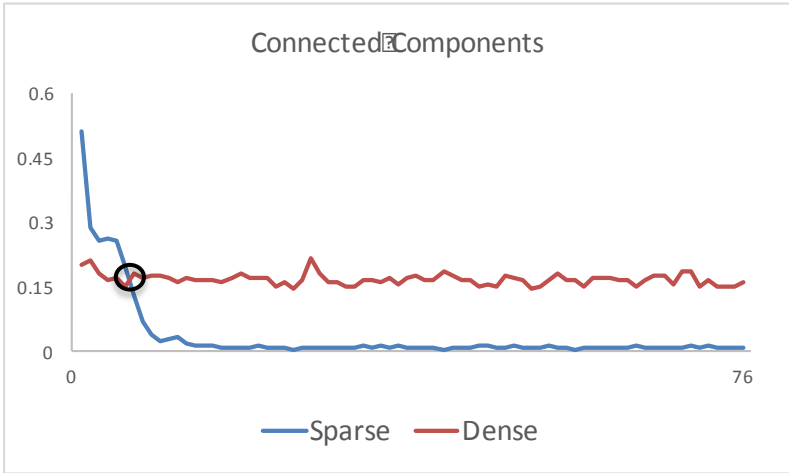


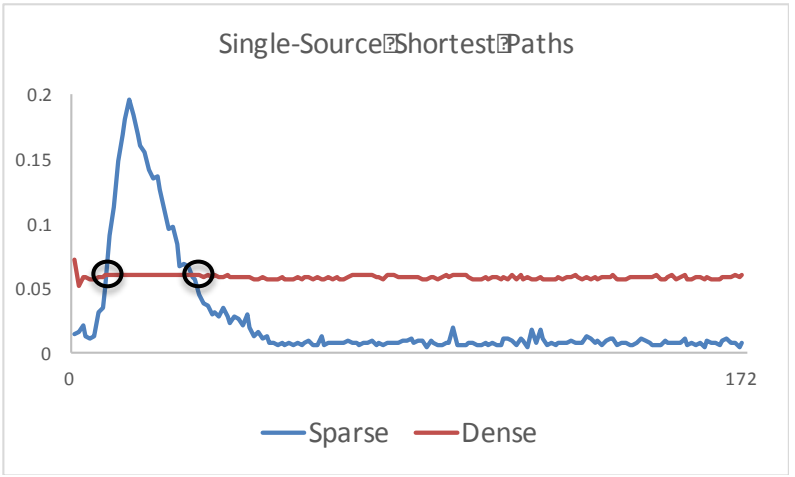


分布式push/pull效果



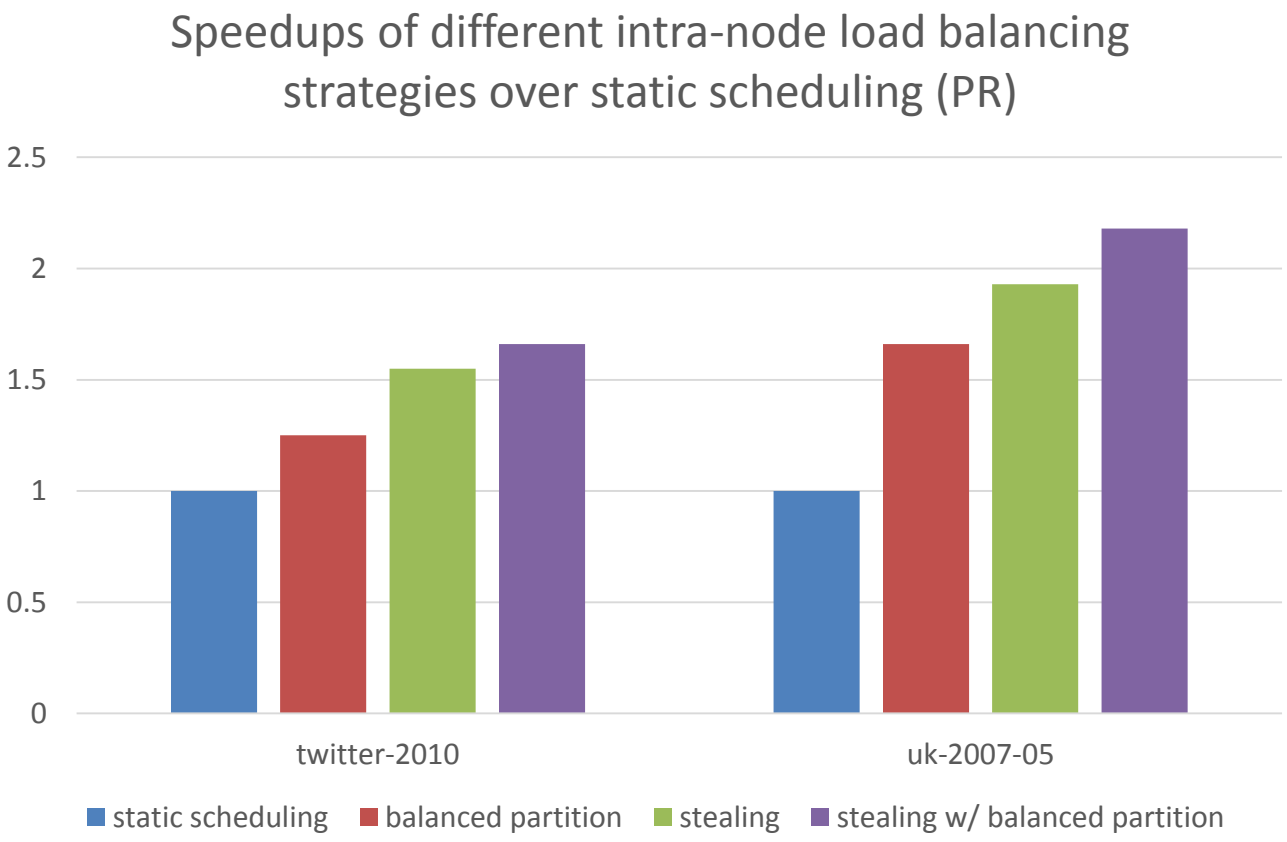
Runtime of each iteration using sparse / dense mode (uk-2007-05)





“mis-predictions”: 2 out of 76 for CC and 5 out of 172 for SSSP

结点内负载平衡


Speedups of different intra-node load balancing strategies over static scheduling (PR)
2.5
2
1.5
1
0.5
0
twitter-2010
uk-2007-05
static scheduling
balanced partition
stealing
stealing w/ balanced partition

分布式系统Gemeni - 性能

- 测试环境
  - 8台双路Xeon E5-2670V3 4 (hypert.) vCPU cores
  - 128 GB memory
  - Infiniband EDR(100Gbps)

- 与PowerLyra相比，加速比约为18.7X
- 与GraphX相比，加速比约为80-300X

Graph	PowerG.	GraphX	PowerL.	Gemini	Speedup (×times)
PR					
enwiki-2013	9.05	30.4	7.27	0.484	15.0
twitter-2010	40.3	216	26.9	3.76	7.15
uk-2007-05	64.9	416	58.9	1.48	39.8
weibo-2013	117	-	100	8.36	12.0
clueweb-12	-	-	-	36.9	n/a
CC					
enwiki-2013	4.61	16.5	5.02	0.237	19.5
twitter-2010	29.1	104	22.0	1.13	19.5
uk-2007-05	72.1	-	63.4	2.08	30.5
weibo-2013	56.5	-	58.6	2.62	21.6
clueweb-12	-	-	-	24.5	n/a
SSSP					
enwiki-2013	16.5	151	17.1	0.514	32.1
twitter-2010	12.5	108	10.8	1.15	9.39
uk-2007-05	117	-	143	3.45	33.9
weibo-2013	63.2	-	60.6	4.24	14.3
clueweb-12	-	-	-	44.3	n/a
GEOMEAN					18.7

内存占用情况

Graph	Raw	Gemini	PowerGraph
EnWiki	0.755	4.02	13.1
Twitter	10.9	25.1	138
UK	27.8	57.2	322
Weibo	47.9	73.3	561
ClueWeb	318	575	-

- Gemini的内存占用约为PowerGraph的六分之一
- 意味着可以用更少的机器获得更快的分析速度，降低用户大数据分析的成本

性能优先的大数据系统

数据模型：区分只读数据和可读写数据

数据结构：基于混洗的数据结构

编程抽象：基于点和边的集合，编译与运行时优化

执行平台：单机内存 → Out of core → 分布式 → GPU/APU/FPGA

编程系统	数据模型	容错能力	性能	自动负载平衡
MPI	可读写数据集	弱	高	无
MapReduce	只读数据集	强	很低	有
Spark	只读数据集	强	低	有
GraphLab	可读写数据集	弱	较高	有
Gemini	部分只读，部分可读写	较强	高	有

感谢聆听！

BDTC 2016中国大数据技术大会
Big Data Technology Conference 2016