NoSQL — MapReduce — tuples (key, value)

SQL — databases — "triples"

OO databases ≈ NoSQL "objects"

XML

"trees"

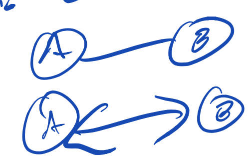
graphs

adjacency matrix

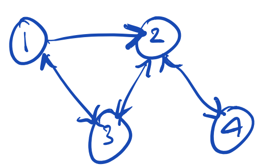
symmetric (around diag)

undirected } weighted

directed



} ≡



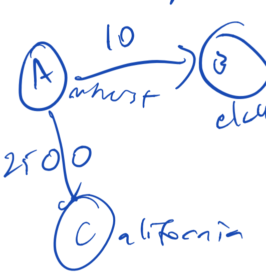
} vertex-edge relation

directed

$A \to B$

$B \not\to A$

/weighted



(cities)

undirected — symmetric

space, FB social networks ~~internet~~

directed — asymmetric

internet

Twitter

static — always unchanging

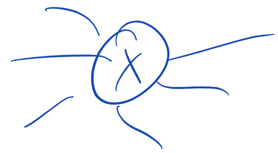
Dynamic — potentially changing

vertex vertex

(1, 2) weight
(1, 3)
(2, 3)
(2, 4)

(traffic - weight)

social network



deleted edges
new edges
new vertices (users)
deleted vertices

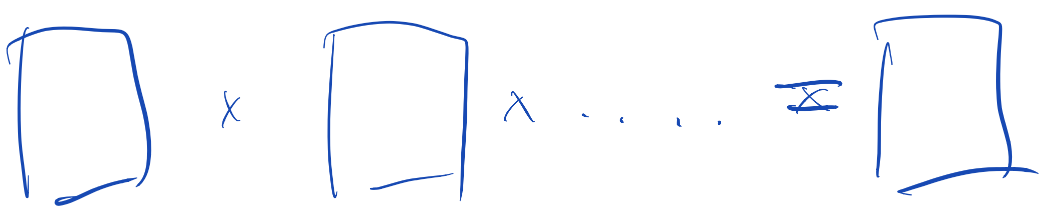
implicit graph representation
(edges)

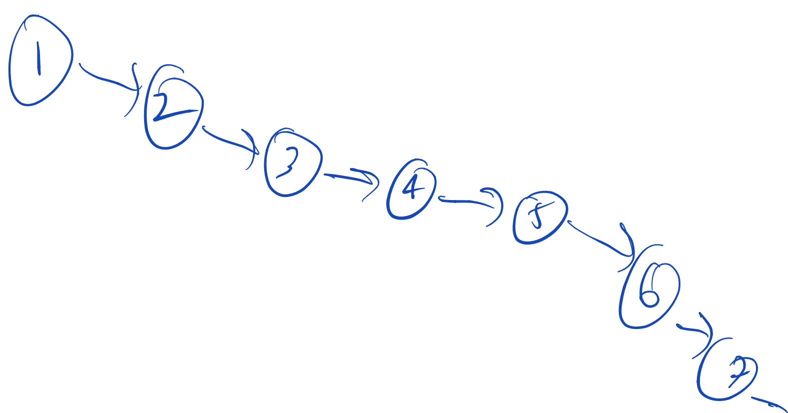
edge-centric

edge list

(2, weight)

1	[2, 3]
2	[3]
3	[1, 2]
4	[2]





1 2 3 4 5 6 7

$$
\begin{array}{c}
1\\2\\3\\4\\5\\6\\7
\end{array}
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0\\
 & 1 & 1 & 0 & 0 & 0 & 0\\
 & & 1 & 1 & 0 & 0 & 0\\
 & & & 1 & 1 & 0 & 0\\
 & & & & 1 & 1 & 0\\
 & & & & & 1 & 1\\
 & & & & & & 1
\end{bmatrix}
$$

Sparse

lots of zeros

dense matrix $(N \times N) \to N^2$

1 entry $O(1)$

sparse matrix $(N \times N)$

coord + value

(1, 1,) (1, 1, 4)

(1, 2,)

(1, 3,)

(1, 4,)

$$
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0\\
1 & 0 & 1 & 0 & 1 & 0\\
 & & \vdots & & &
\end{bmatrix}
$$

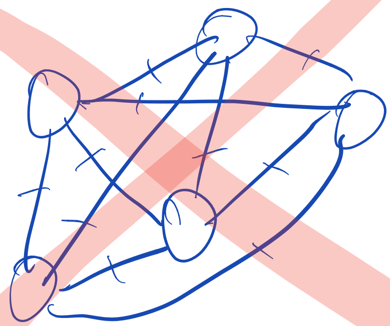
(1, 1, 1)

(1, 3, 1)

(1, 5, 1)

social net:

N vertices $\ll N^2$ edges



5 vertices

$\frac{N(N-1)}{2} = 10$

$O(N^2)$

Complete graph

⟹ sparse representation

- run-length encoding (suboptimal for graphs)
  sparse matrix
- (vertex, vertex) $O(E)$ ~ $2E$
- edge lists $O(V+E)$ ~ $V+E$

V vertices
E edges

if $V \ll E$
$\approx E$

edge list

weights

1 [edges]

2 [edges]

⋮ ⋮

key , [values]

(v, [e]^{n}) $\Rightarrow$

weight(e)

(1, [2 : 3, 3 : 10, 4 : 100])

$\Downarrow$

([1, 2, 3]

[3, 10]

[4, 100])

(2, . . .)

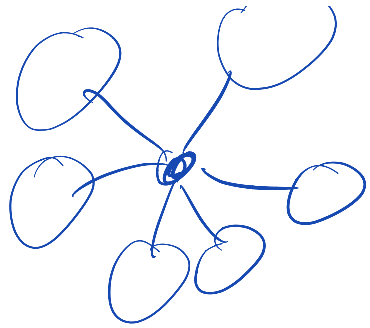
— mismatch between problem & interface abstraction — low utilization

— performance

iteratively executing MapReduce $\Rightarrow$ BAD (I/O)

— locality insensitive

framework oblivious to fact these are graphs



Friend of a friend

graph databases

input
vertices edge lists

"Combinatorial academics"
write only
LPU
salami slicing

locality
which vertices go where
vertex-centric
"Think like a vertex"
all edges are with each vertex

problems: Justin Bieber
Ellen DeGeneres
selfie

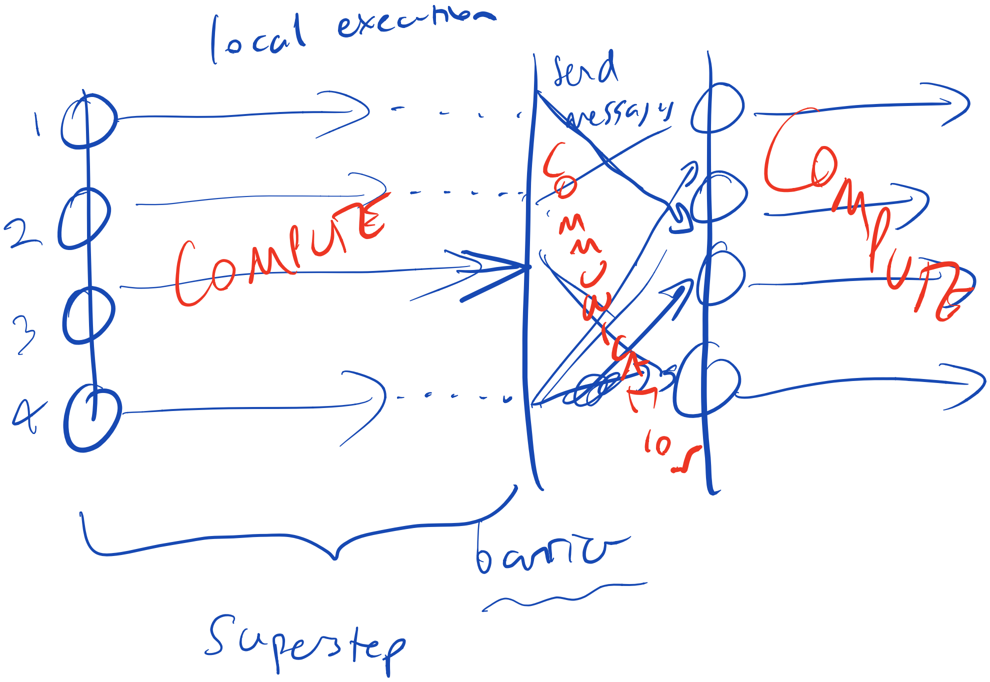
avoid messages
=> local
(fault tolerance)
too many!

IN RAM

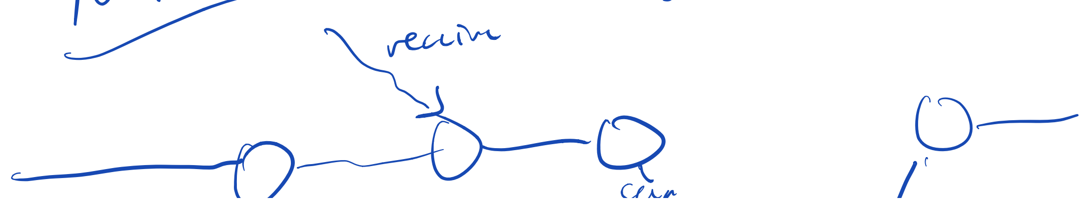
BSP bulk-synchronous parallelism
Valiant 1980

programming model
perf guarantees / cost model

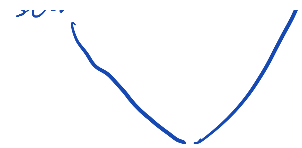
~~PRAM model~~ academic



MPI Message passing interface



async



synchronous

GENERAL
unconstrained
parallel
execution

MT, MP

broadcast (all)
group communication (some)
point-to-point

mpi_receive (2,
(MPI_ANY

async
synch

problems
deadlock

SPMD
single program
multiple data

no shared memory
(distributed)

```
if (mpi-self == 0) {
  mpi-recv(1, b)
~~if (mpi-s~~
else {
  mpi-recv(0, b)
}
```

```
if (mpi_self() == 0) {
   // master
} else
   // worker
}
```

```
if (self == 0)
    mpi_recv(Any)
    print result
else
    mpi_send(self, 0)
```

"race" (message passing)

asynchrony

determinism 

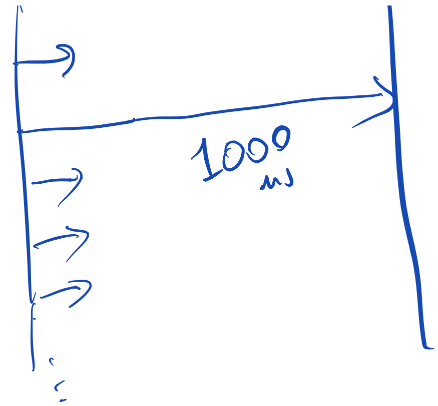
barriers ≈ 

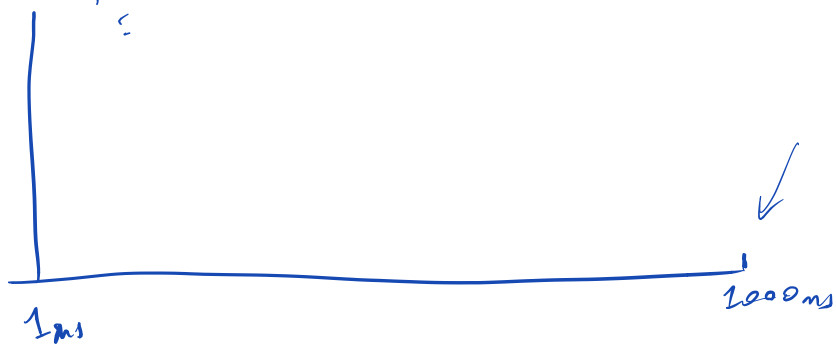
synchronous  as long as no deadlock

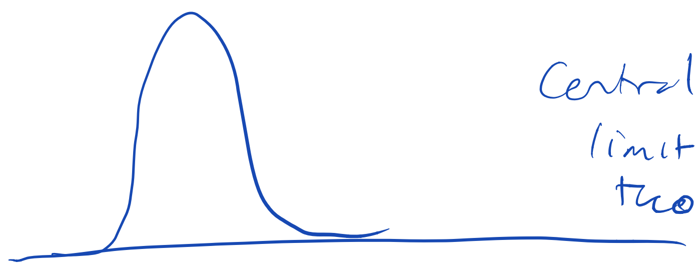
---

BSP – no comm cycles ⇒ no deadlocks

1 ms







Central limit theorem

sum of ~~add~~ independent identical distributed variables

=> approx. normally distributed

random → load balance