

Redis

Bloom filter —
set membership

HyperLogLog
cardinality

Morris counter
counting

— monotonically increasing

approximate counter

ctr. increment()
ctr. getValue()

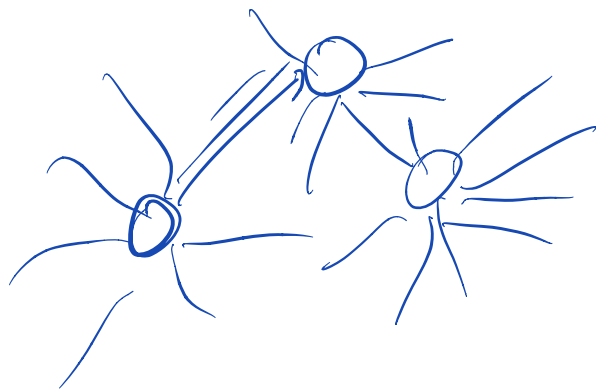
key: "counter 12"
value: 235

exact

```
increment() {  
    x++;  
}  
  
getValue() {  
    return x;  
}
```



eventual consistency



Norm's worm

ctr := 2

1, 2, 3, 4, 5, 6, 7, 8

2, ..., 4, ..., 8

increment () {

$P = 1/x;$ (1... x) $x++;$

with $P(1):$

$x = 2 \cdot x;$

}

1 2 3 4 5 6 7 8 9 10

2 $\frac{1}{2}H$ 4
~~2 $\frac{1}{2}T$ 4~~

$E[\# \text{ of successes} / 2 \text{ trials}] = 1$

4 + + + +

6 + + + +
 + + + + 16

32768

(1)



w.h.p.
with high probability

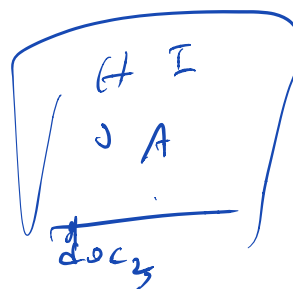
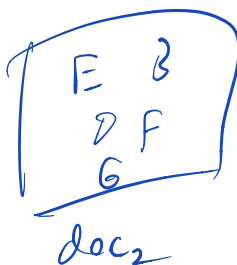
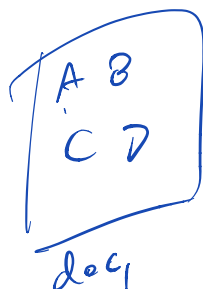
PERS

Accuracy

SPACE
EFFICIENCY

PICK TWO

SPLITTING



||


 inverted index

word	doc #
A	[1, 3]
B	[1, 2]
C	[1]
⋮	

map ([(url₁, "the sandwich is delicious"),
 (url₂, "the pizza is delicious"), ...])

↓

[("the", url₁), ("sandwich", url₁) ...]

"reduce"
 ↓

[⋮
 ("the", [url₁, url₂ ...])]

BisTable

rows
 columns
 structure
 ↓
 value

word	urls	freq
⋮		
the	[1, 2, ...]	

url	doc	freq



map-reduce

fine abstraction
easy

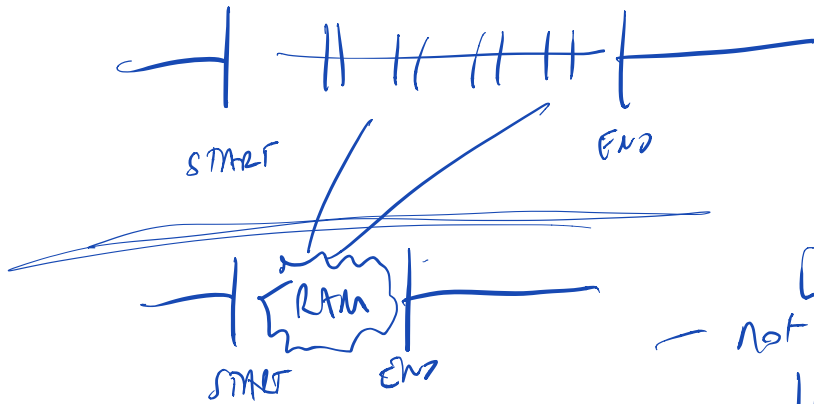
abstracts away fault tolerance

put

~~~~~ X  
/ read

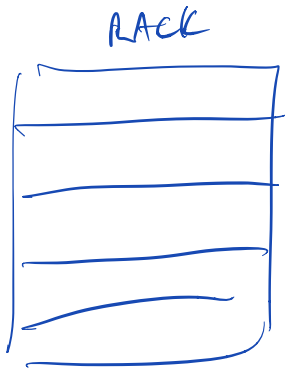
map  
reduce

~ write

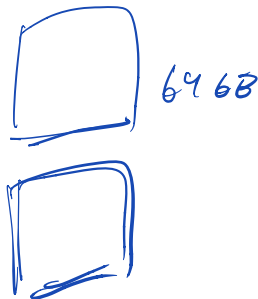


[capacity limited]  
 — not enough RAM?  
 $|RAM| \ll |DISK|$

— RAM is volatile  
 (not FT) power outage  
 computer failure  
 $\Rightarrow$  data loss



— Cost  
 $(\geq 40 \times \text{more } \$ \text{ than disk})$



NAS

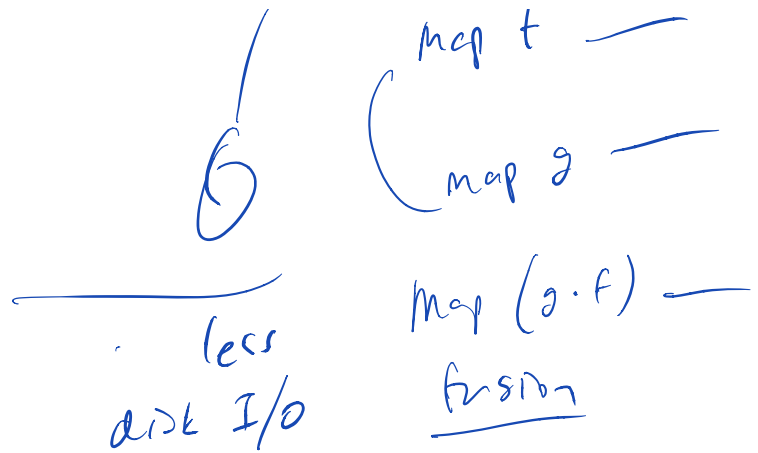


map reduce  
 map reduce  
 map reduce

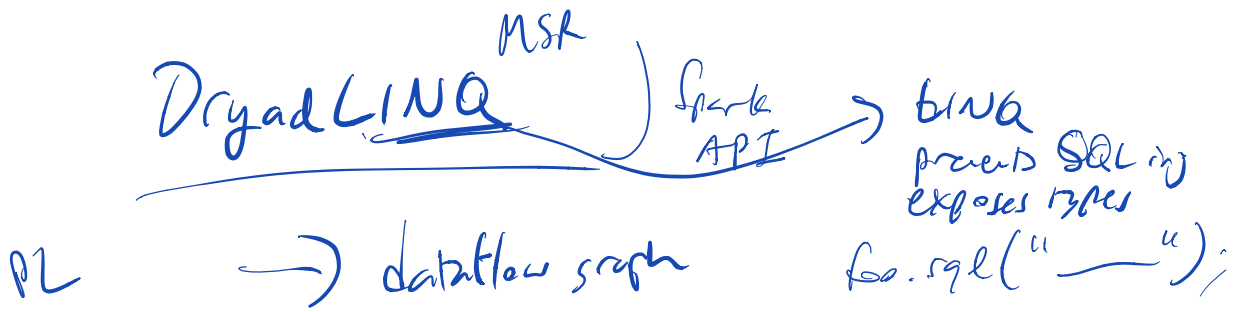
map  
 reduce


(lazily built  
 data flow graph)





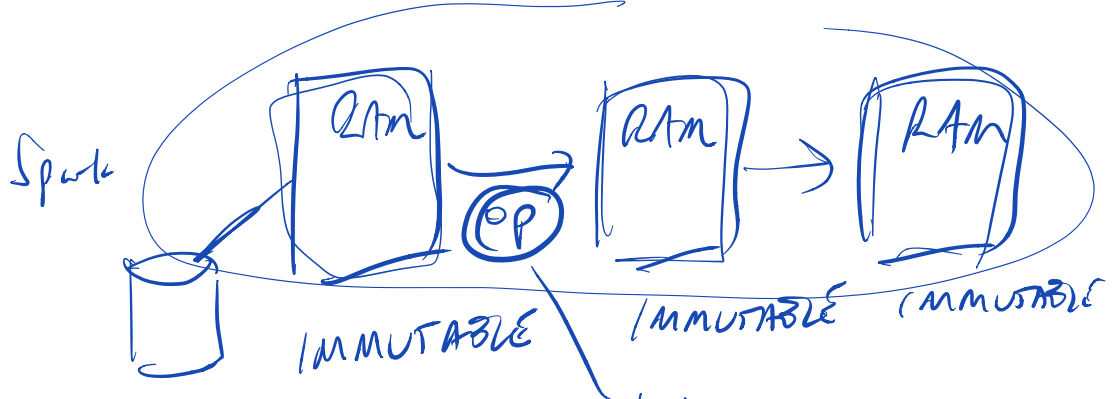
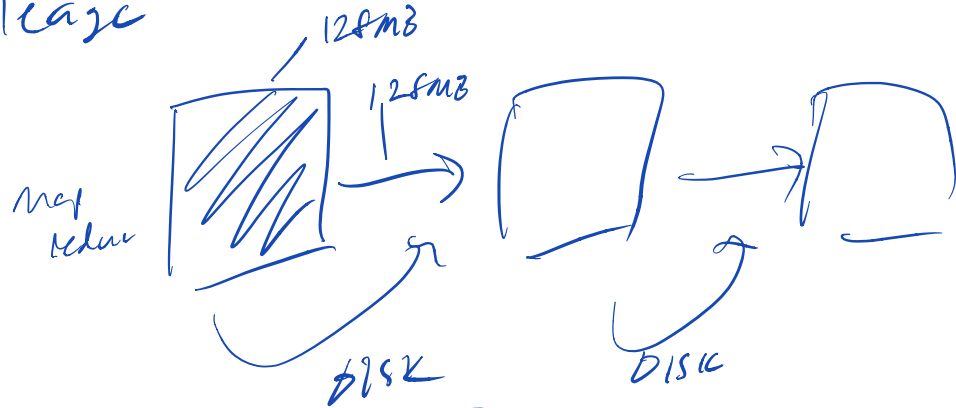
RDDs  $\Rightarrow$  Spark  
resilient distributed datasets



SQL injection  
" "   
Bobby ; DROP TABLES

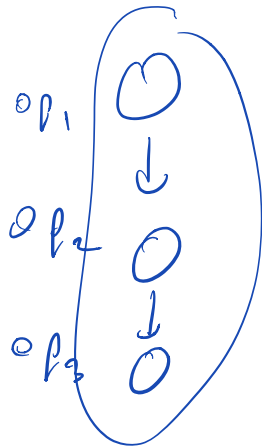
SELECT \* FROM FOO WHERE NAME = "John Doe"  
(OR NAME = "X")

lineage

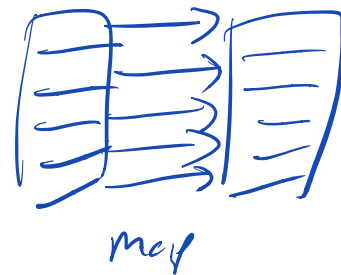


deterministic  
side-effect free

$f(x) \rightarrow \text{output}$



narrow — one-to-one  
wide — one-to-many



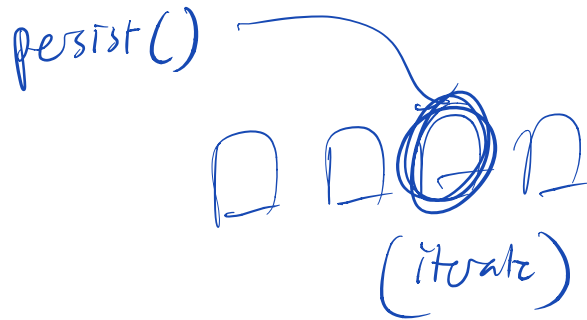
aggregate  
ops

shuffle  
hash based join

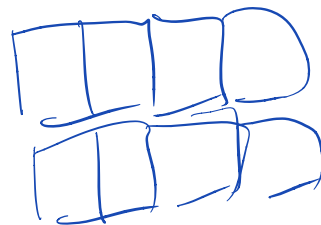
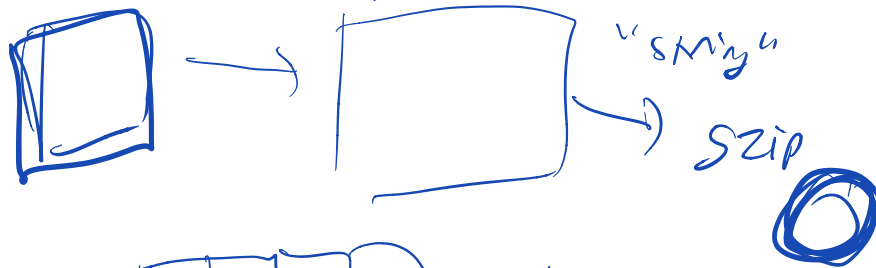




Spark APZ  
over RDDs



Serialization

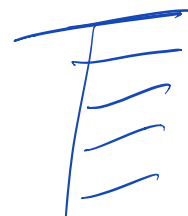


@ Endianness



just like  
map reduce

BATCH processing

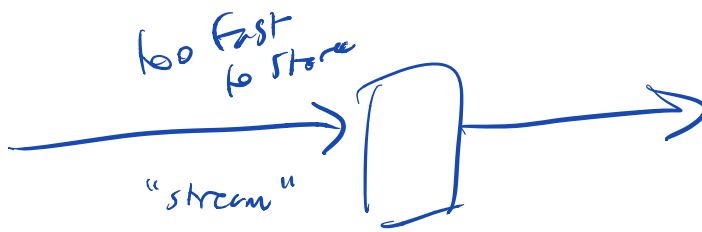


lazy  
graph  
optimize  
re-plan

Interactive

Spark Streaming

Two Sigma



one-pass