Lecture 22:

Data extraction

abstraction:

extract - combine

extractcombine :

```
    ('k * 'k → order)
  * ('a → ('k * 'v) Seq.seq)
  * ('v * 'v → 'v)
  * 'a Seq.seq
→ ('k, 'v) Dict.dict
```

Google MapReduce

Yahoo Hadoop

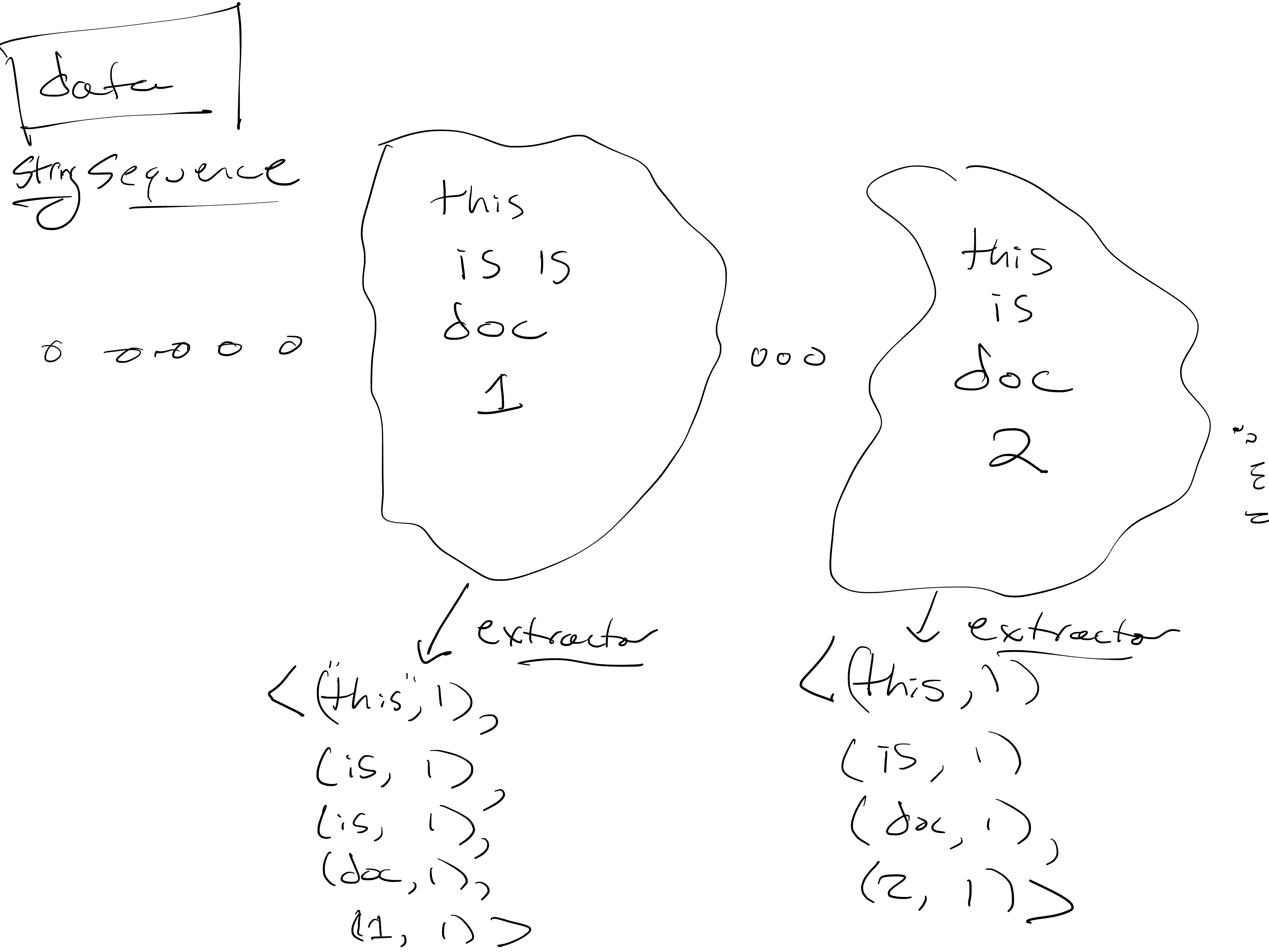
comparison

eXtractor

combiner

'a is a doc

(documents)


data
StringSequence
this
is is
doc
1
this
is
doc
2
extractor
extractor
<("this", 1),
(is, 1),
(is, 1),
(doc, 1),
(1, 1)>
<(this, 1)
(is, 1)
(doc, 1),
(2, 1)>

Dictionary where:

{ "this" ~ 2,
"is" ~ 3,
"doc" ~ 2,
"I" ~ 1,
"2" ~ 1 }

key value

```
fun   wordcount(docs: string Seq.seq): (string,int)
                                          Dict.dict =
   extractcombine
      (String.compare,
extractor  fn doc: string => Seq.map (fn w => (w,1),
                                           words doc),
combiner   fn (x,y) => x+y,
        docs)
```

MAP-reducable type

lists

map

reduce

trees

map

reduce

seq

map

reduce

files

map+

reduce?

networked

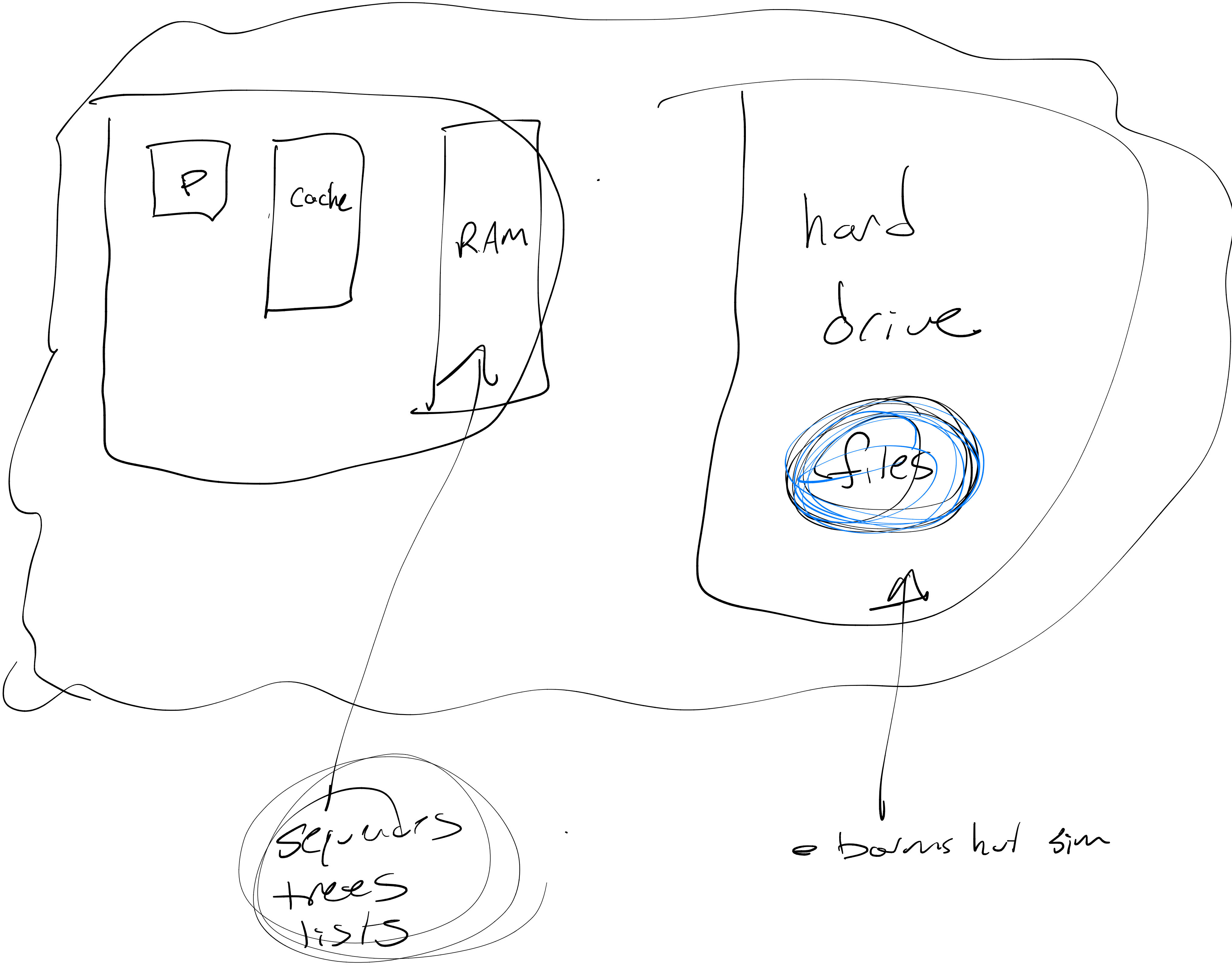
cluster

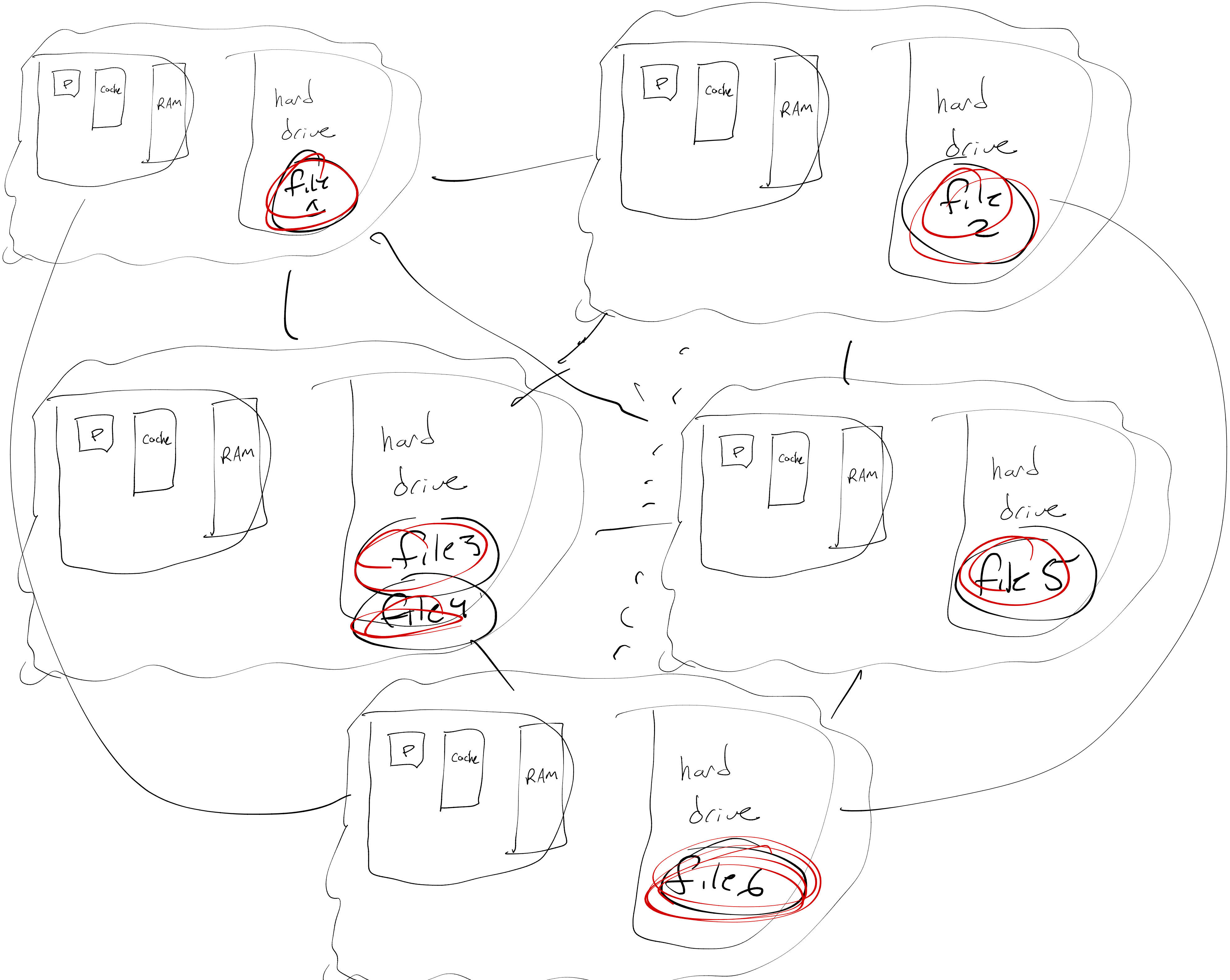
of

computers

map+

reduce


P
Cache
RAM
hard drive
files
sequences
trees
lists
e bounds hut sim


P
cache
RAM
hard drive
file 1
P
cache
RAM
hard drive
file 2
P
cache
RAM
hard drive
file3
file4
P
cache
RAM
hard drive
file 5
P
cache
RAM
hard drive
file6

```
Signature MAP_REDUCE =
 Sig
    type 'a mapreducable

    val map: ('a -> 'b) *
             ('a mapreducable)
          -> 'b mapreducable

    val reduce: ('a * 'a -> 'a)
              * 'a
              * 'a mapreducable
end
           -> 'a
```

Short summary of lots of data

reduce (+,

0

map (fn x => x+1, <1,4,8...>))

$\mapsto$ reduce(+, 0, <2,5,9 — — — — >))

$\mapsto$ 2+5+9+ —

```
signature MAP_REDUCE =
sig
  type 'a mapreducable

  val mapreduce : ('a -> 'b)
                * 'b
                * ('b * 'b -> 'b)
                * ('a mapreducable)
              -> 'b

end
```

mapreduce(fn x => x + 1,
0,
+,
<1, 4, 8, - - - - -))

=

2 + 5 + 9 + - - - - -

```
structure SeqMR :> MAP_REDUCE =
struct
   type 'a mapreducable =
        'a Seq.seq

   fun mapreduce(f, e, n, s) =
   Seq.reduce(n, e,
              Seq.map(f, s))
end
```

Structure FileMR : MAP_REDUCE

data is stored in a file

+ mapreduce over it!

training

+ test]