

TBD 31/3a

Eigenvectors      Eigenvalues } range from 1 to n  
                             k-nearest neighbor

- Data an Open Street Map. (osmosis Open Street Map  
Функция привязки к географии на веб-сайте CSU)  
аннотация для OSM.

→ To describe how queries are applied to files or blocks

we have (32KB) → datafile { blocks → logical block size

→ edw size bytes!

2 types { datafile  
indexfile for R\* tree

blocks →  
block1  
block2  
block3  
⋮

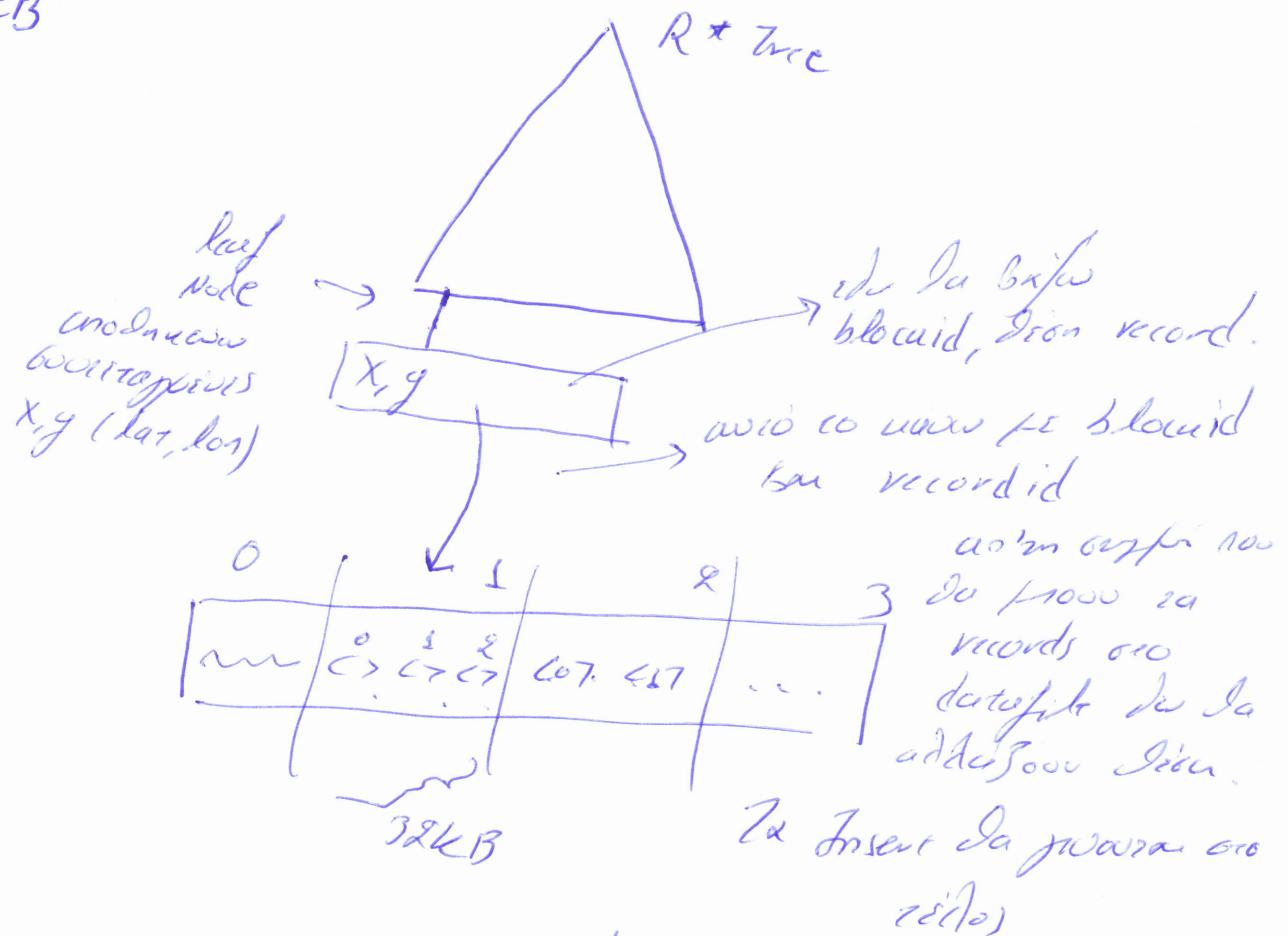
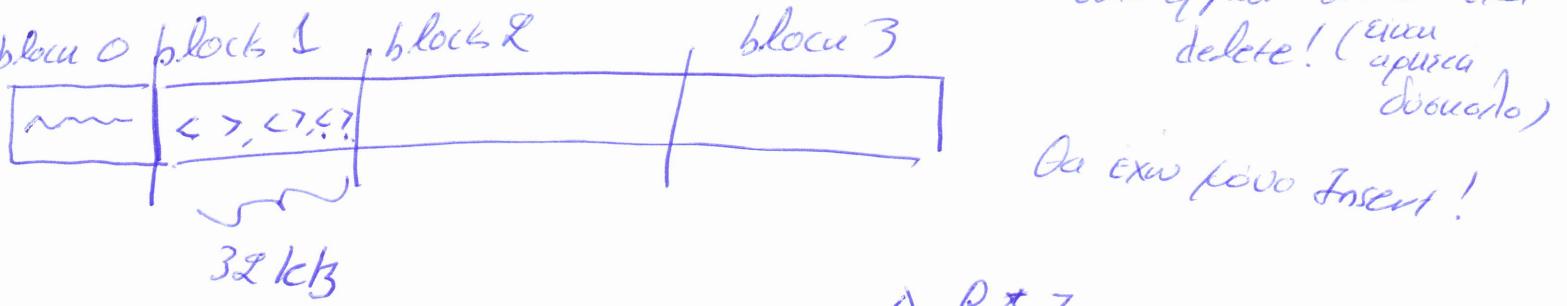
→ edit before propagates  
new query.

metadata

100A records etc.  
100B blocks etc.  
(200C rows for each)  
→ indexfile

→ Sjau projektor blokken.  
Nooit eerdaagse ova  
en projectoren ova nergens  
ova record exo's id  
lat  
don

$\Rightarrow$  O business speaker va prezenta pro <sup>top</sup> 100 de diapositives! va elibera jurnalos!

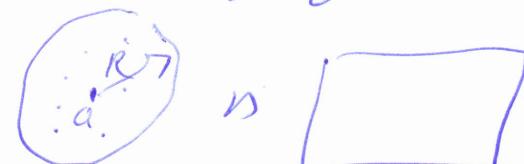


7. Da se p'lo 2 records exoun idio x,y?  $\rightarrow$  awaçipar o're 1º  
ni on leaf Node das p'los tam co id.

Ese xzifra co Index File

deku epurufaca } odnosegwan jwawa  
bx1

epurufa repikas.  $\rightarrow$  n'igre q' se auwa k



operac'ario.  
n'igre, p'los + m'los

TBD 31/36.

- 1) Luv Sequential Search Scans all the blocks
- 2) So R\* search optimizes from blocks.

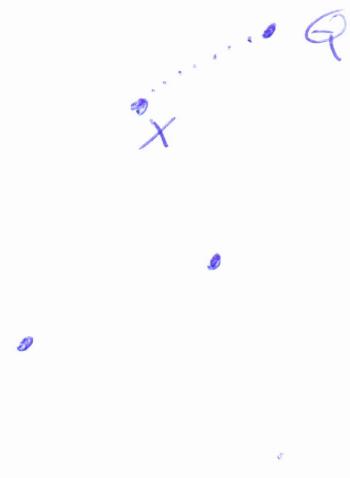
Applicability

Sequential search vs R\* tree search.  
1) work on subset in regions overlapping  
2) fitness or distance to the nearest neighbor.

R\* requires to do R tree insert, to search even if one  
is in the same group of points R tree can be R\* inserted  
but it's not always true! R\* trees gain more space  
but less performance.

---

Efficiency  $k=1$  due to Anisotropic feature.



### Sequential Scan

- Open data file
- Read all records
- Process according to 'Q' condition
- Process in groups sequentially
- no exit to own distance.

or Index 100s K, iteration 20s K  
iterations. (Max Depth Max Heaps)  
max depth distance i. Max Iterations  
below an even higher max iterations  
before, too. Max Heap (max distance  
into 20 iterations)

Kon PE Race.



$[R_1 | R_2]$

$[R_3 | R_4]$

$[R_5 | R_6]$

$[A | B]$

$[D | E | F]$

$[G | H | I]$

$[K | L]$

$X \dots R \dots X$

предыдущая модификация  
буферов не определена

$X \dots R \dots X$

Несколько буферов  
не определены  
анонимные до этого  
модификации не явны

новые конфликты на  $Q$

для двух пр/а, блоки  $R_1$ ,  $R_2$  должны быть  
одинакового размера по памяти. Но  $R_1$  имеет  
размер  $R_3$ ,  $R_4$  меньше размера  $R_3$  и одинаковы по размеру.

Блоки  $R_3$ ,  $R_4$  должны иметь одинаковый размер  
и быть A или B объектами. Но это не всегда?  $\rightarrow$  A.

Блоки не определены  
одинаково блоки  
ниже не явны.

TBD 33/3c.

Adioo eten na erogos, am uno zo Q ens zo A.

At 10 km. Naas zifw ou de spesifika zo gefu a Ado  
toffina zo deugos? Xpafian ux fum en Ru?

A Ru antea nawi en 10km OK!  
de spesifika. <sup>Itw oan</sup> <sup>Wido swanguey!</sup>

Nayu ja kivip zo Q bau anuva en anobrion uno zo A  
avios o boklos des nfuva zo Ru.

Enobrion en R<sub>1</sub>, R<sub>2</sub>, ja en ido Nayu en des nija en Ru  
des da nawi eote ozo Re.

na k=2

En R<sub>3</sub> da erogia A bau B, bau ws anuva da cira en  
anobrion zo Q uno zo B (nivo diwu zo pugantigo, o nlo  
anofaupufedos) (nivo diwu jesi nicos)

bau da erogia da kifw bau en R<sub>4</sub> (nivo diwu ozi  
to repon)

apa da des zo objets [D|E|F] ofws hawwa des ~~erogos~~  
repixim piwo enzo mibdo  
pe tonyo Q bau anuva R.

Lipgo dypos en MBR eten bau kifw, ja uno zo Jeku o-frain  
Enobrion en pifz, o boklos anuva des erogia zo R<sub>4</sub>.  
apa brafam

Tentativ na k=n bau nobw toffina zo deugos en  
erogia o-jouros da des repixim anuva, an Propri  
da repixim, Erpar anuvalopredos da zo kifw  
avios o algoridm erpar anuvalos!

≠ Για αυτούς οι, ο μετασχηματισμός είναι ο K (αριθμός των Ρεγμάτων)  
αφού ο αλγορίθμος θα σταθεί στο R1

αντιτίθεται: στο μετασχηματισμό παρέχεται η είδηση για το  
μετασχηματισμό γείσων. (δάδας)

Οι παραπάνω R1

επιλέγεται μετασχηματισμός του A

καθώς το ίδιο αναγράφεται (γίνεται στο R1)

αφού δε γίνεται στο R2! Οι παραπάνω για την άποψη του R1!

Η ημέρα R2

τα διαδικαστήρια του R2 θα διατίθενται στο K, αφού αποτελούν τους

τελικούς μετασχηματισμούς της επιλεγμένης είδησης για το K.

Επειδή τα περισσότερα...

δεν δινούνται στην αρχική σειρά αλλά στην τελική σειρά στον ίδιο ρυθμό +00  
Χρησιμοποιείται Depth-First Search (DFS)

Η έκθεση των γείσων περίπου στην ιδιαίτερη μορφή  $t=1$  } διατίθεται στην  
(γενικά 160 γείσες σε κάθε διάνυσμα) } σε πολλούς  
διατίθεται στην ιδιαίτερη μορφή  $t=1$  } διατίθεται στην ιδιαίτερη μορφή  $t=1$

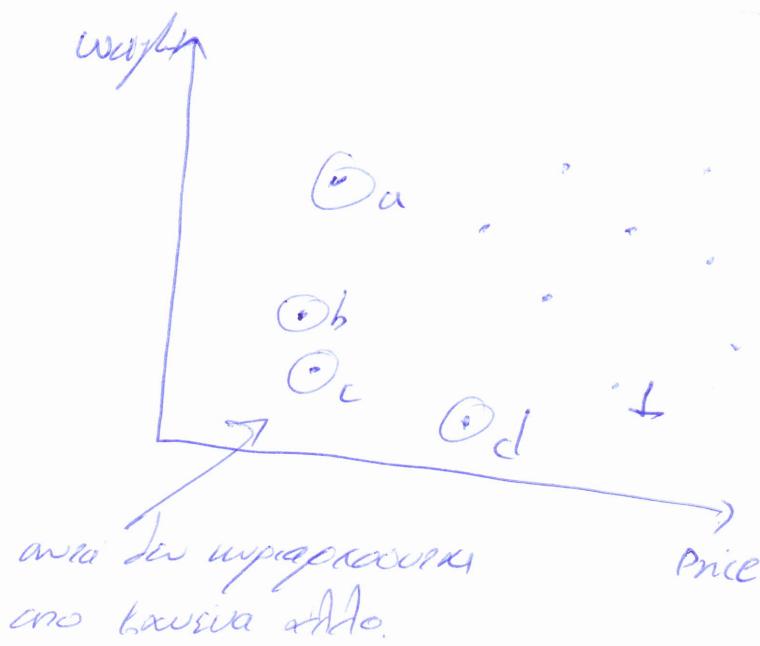
προπονείται στην ίδια μορφή  
εξαρτώντας την από την ιδιαίτερη μορφή  $t=1$   
 $getAll = true$

TBD 31/3 d.

1:07:45

aiða spærifara (ekios knn bæt area range)  
meðan <sup>nefjóðus</sup> ~~nefjóðus~~ → nlogn n logn se spærir Rtree.

## Espærifara Körumálfarðar



deitir ydina hér  
slappia (nú laptop!)  
slappið danci = aðeins laptop.  
nú eru Þóru ydina hér  
slappið slappið laptop  
 $k=4$   
min - max.

Leyfilefx min - max  
síðu slappið.

max - min, afbæt laupia  
→ nánar deit.

I espærifara að d.  
(nánar eru min - max leyfilefx)

fun weight for max price  
→ fun deit.

→ Eru ófriður körumálfarðar  
nánar sé einn ófriður

max weight for min price  
→ nánar deit.



er g spærifara nánar deit.

~~slappið~~ bæt Ø eru

slappið bæt us spær

en fyrir umhverfum bæt us

spær eru ófriður.

x a, b, c, d eru körumálfarðar  
aða kvarna, bæt kvarna  
flestu róður eru körumálfarðar.

!javikarinn er fyrir ófriðum körumálfarðum  
mínus Þáðar körumálfarðar =  
að fyrir Þáðum ófriðum.

Ma didikates



Griva arithmetiko kuriapxhi

Two categories

selection to indices

deks no sevka kuriapxa  
besas, zosha no sevka  
da napis oso eido.

Now) xroloforoia ova R-tree na va bpos eno kuriapxhi;

Na bpos na sevka noo des kuriapxhoxwxi.

→ Me brate force & Zgop. Ifo.

Ou xroloforoia vadiablos napa  
noddles yppis me secondary  
memory  
badalou avodoteti doan

narpow oda ca dovara feryapia  
bae bnos eido da napis avia noo  
des kuriapxhoxwxi.

Typ em t<sub>q</sub> des av co p kuriapxhi  
mo co q

Θoutuo bpos ~~avodoteti~~ an kuriapxhoxwxi  
mo naroio adda mida za sevka  
 $O(n^2)$

→ Xroloforoia wi index ova R-tree. Na eno eipreson  
enou kuriapxhoxwxi.

Perivo diavres  
10diodikisata data  
exw o-an index

Now) xroloforoia wi R-tree na va bpos eno  
kuriapxhoxwxi?

1...? : eno 1 kuriapxhi eno ?.

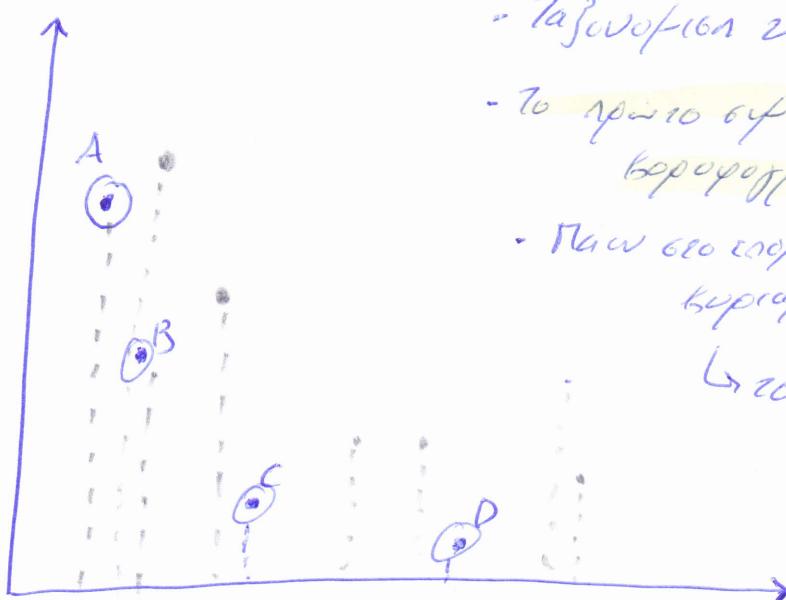
→ ido g, diagoprenio x.

na vo kuriapxhi co p losu eno q.

da zposi  $P \leq Q$  adda vo stixhi cool 1 enou kuriapxhi  
noo ro P avodotai bad-nes.

TBD 31-3e.

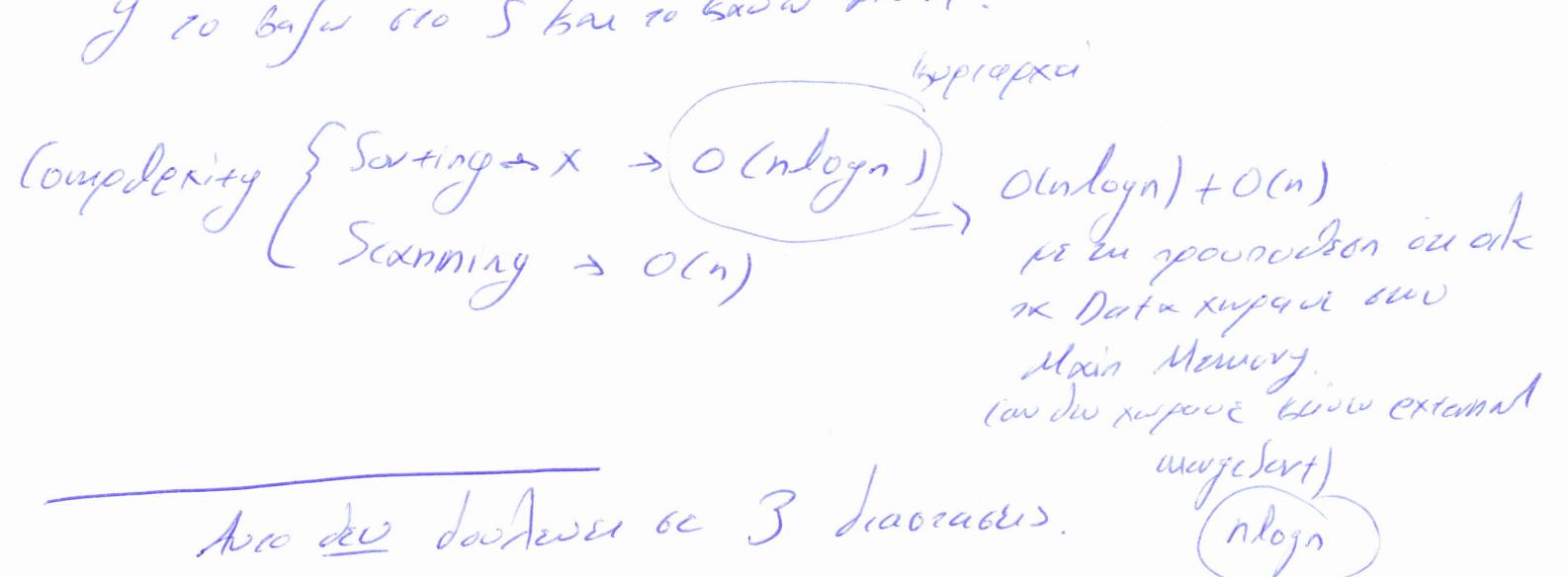
Για δύο διαράσσεις:



- Παρεις γραμμές αλλα είναι στην αυτή σήμερα X
- Τα γράμματα είναι γραμμές.
- Το γράμμα στοιχίου αντικαθίσταται στην καροτσοφόρα.
- Μάλιστα το γράμμα παραπέμπει στην Α καροτσοφόρη σε αυτό.
- Στη Β δε καροτσοφές στην Α χρησιμεύει το γράμμα στην καροτσοφόρα.
- Το γράμμα στην Β σημαίνει την καροτσοφόρα στην Α.
- Η λέξη περιβάλλεται στη Β μεταξύ των δύο καροτσοφών.
- Η λέξη περιβάλλεται στη Β μεταξύ των δύο καροτσοφών.

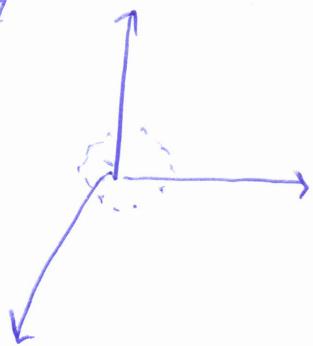
$$S = \{A, B, C, D\}$$

Στην αρχή τα δύο γράμματα στην Β αντικαθίστανται στην Α.  
Οι δύο γράμματα στην Β σημαίνουν την περιβάλλετα.



## OR Tree kai epomenika Korologymata

day 7



3 diabolos  $\rightarrow$  max-max-max

Μηρια οντο περιπτωσεις ενα-ενα-ενα  
σημαντικη παραγωγη απο την αποστολη

(Block Nested loops : loops in loops in blocks : ανταναγγει)

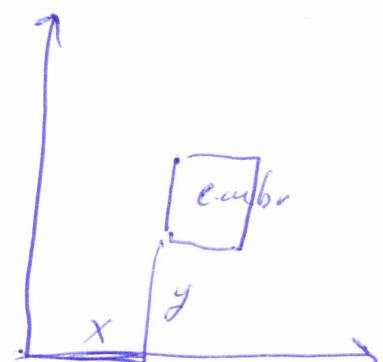
## BBS Algorithm - Branch & Bound Strategy.

Now Διαβολοι = νοικια σφραγιδων χωρου Διαβολοι

Now Διαβολοι η παραγωγη (Αδαμα) Διαβολοι.

Ο γραμματικος πει ειναι φανατικος συνδυωσης, βασιζεται σε μονοτονια,  
ειναι αποτυπωσια, αλληλεγγυη σεν αποτελεσμα.

Minimum distance = min manhattan



$$\min dist(e, \{x, y\}) = x - e + y$$

Ενα καλοπανικο λειτουργικο προγραμμα που παραχθηκε για την εφαρμοση σε  
blocks Διαβολοι.

(Δεν υπάρχει παραγωγη!)

$\begin{cases} \text{Order} \\ \text{Διαβολο 5: σφραγιδωση} \\ \text{Priority Queue: Ranking την αντικ.} \\ \text{ειναι γραμματικη μονοτονια σεν αποτελεσμα} \\ \text{ηο ειναι απλη ειναι απλη} \end{cases}$
--

TBD 31-3 f.

Appartezza diag. 26.

		(min heap)	
	action	heap contents	S
①	access root	<del>(e7, 4), (e6, 6)</del>	$\emptyset$
②	read e7	$(e_3, 5), (e_6, 6), (e_5, 8) \rightarrow (e_4, 10)$	$\emptyset$
③	read e3	$(e_1, 5), (e_6, 6), (e_5, 8) \rightarrow (e_4, 10)$	$\emptyset$
④	read e6	$(e_1, 5), (e_6, 6), (e_5, 8) \rightarrow (e_4, 10)$	$\emptyset$

отсътствие на елементи в минимална куча  
он борави със структурата на кучето  
он борави със структурата на кучето

①

пак още пъти съм съзнателен за една грешка, която да правят?!

ако човек има някои битки и тези битки са създадени от е7

$e_7 : 4$  } ако създавате е7  
 $e_6 : 6$

②

битки  $e_3, e_4, e_5$   
 $e_3 : 5$   
 $e_4 : 10$   
 $e_5 : 8$

③  $X, K, i$

единственото място, където  
авторитетът  $\Rightarrow$  е пропускан една  
битка, която е създадена от същия  
авторитет.

Однократно създаване на  $i$ ,  
което е създаване на куче.

④

бъдещите създавани  
се създават със  $i$ . Ако създавате създаване.

④ read es ~~(e<sub>1</sub>, g)~~ (e<sub>1</sub>, g) (e<sub>1</sub>, 10)

5  
εις

⑤ read es.  
(e<sub>1</sub>, g) (e<sub>1</sub>, 10) εις

⑥ read e<sub>1</sub>.  
~~(e<sub>1</sub>, g)~~ (e<sub>1</sub>, 10) εις, ας.

⑦ read e<sub>1</sub>  
(k, 10) εις, ας  
Gratiafu ton en  
repouso hypapaxios

Extract - μεν

∅  
εις, ας, βηματισμον  
hypapaxios.

↳ ουα γρα κωρδες  
Επιφανι.

⑧ e<sub>1</sub> du ειναι αλλα

ποια δια σημαντικον, απο το πλεγμα

b } b hypapaxios l ⇒ b δια hypapaxios μο λανιδα  
του S. απο την πιο σημαντικη.  
skylae.

⑨ dabbefu es

e<sub>1</sub>: { δια τα βασικα λανιδα της Μην Ηερα.

e<sub>1</sub>: { το εξ δια το οντασμα υπο hypapaxios!

εις οδοιδηματα μο εοι.  
απο την ια ειναι εφεις skylae.

⑩ dabbefu es

μ: { hypapaxios μο εοι απο την dabbefu!

η: { λανιδα εξεργασι μο υπο hypapaxios μο πιντ του S.  
δια το εξεργασι με οντασμα μο Ηερα την πιο σημαντικη.  
παιδια μο Ηερα.

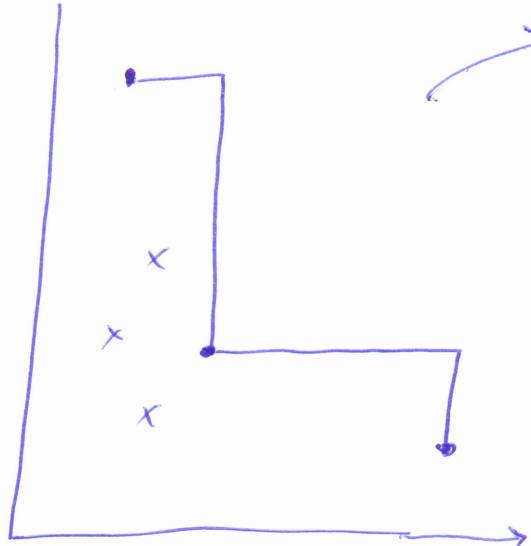
⑪ το ει δια αντικε μο δια σημαντικον, απο αντικαθισμα της  
κυρων μο την dabbefu

δ } b hypapaxios μο αγα πλεγμα πιντ μο κ  
b } c hypapaxios μο ει } δια το βασικ μο S!

TBD 31-3 g.

Σιγκ 33, διάγραμμα της "εκτάσης βυριαρχίας"

αν είναι γενούμενο έτοιμο πρότυπο της ολής της skyline.



ουν βέβαια στοιχεία δε χρειάζεται  
να ταύτισεται με την ημέρα  
εξαγωγής αλλά το γενούμενο έτοιμο  
βυριαρχία στην βυριαρχία που  
βασίζεται στην ημέρα της Σ.

αν καρπώνεται, τα γεγονότα  
της βύρωσης αντίστοιχα

→ Δια Delete

αν δε γίνεται από την Σ αύρια ήτοι

από την ημέρα  
από την ημέρα.

αν γίνεται από την Σ εξαγωγής την ημέρα

BBS Algorithm Performance.

Ενας γενικός θεώρησης για την ΒΒΣ. Εξαγωγή των εγγενέων  
χριστιανών της Αιγαίου. Ο BBS είναι πολύ λεπτός  
για την ΝΝ της ημέρας της προβολής για την Σ διαδικασία.

Baravipes

Independent: Open poppa

Correlated: as subjects n't be subjects b/w n.g.

Anticorrelated: obo subjects & fia diacritics wobbo de  
subjects n' affin (ex obo no subjects  
wobbo b/w obo nro  
subjects ja wobbo)

Gen Ja era n'Affa rep'lecoys,  
o fia am koyayayayay!

- Obo rep'lecoys sun or diacritics wobbo rep'lecoys o fia  
Ja era sun koyayayay!
- Yoyku o fia for diacritics sun sun Ja za koyku o fia n' o fia  
(mu wobbo fia o fia diacritics)
- Mac: fe affixes laconizes u m'laconizes sun o fia uk  
koyayayay sun sun sun n' pibzi

TBD 7-4 d.

Erofevo falypa Spark.

## \* Rapiddides Texikes deoxipion Bedefionu.

cluster computing / Hadoop / Map Reduce

Nodo fogales enneis Data sou Jidu va ete seppuow

## 3 V's of Data.

To xapiba sou Data exous alMafes **Volume**: perigas ojhos Data

**Variety of Data**: noikihofopgia sou  
Sicofia, erofe nypas adfara n infidexifera Data ta  
Data da speni va nraue nraue sou Database ja va rk  
onofre

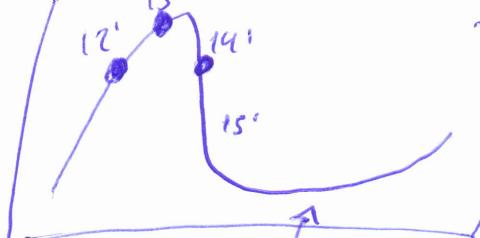
**Velocity**: ncedpsva sou alMafes { Batch update  
Streaming: online Data.

+LV  
Value  
Variety :  
+LV  
VISION  
Visualizetion

Noo ja nraue sou Data? Eisai optees or baous Bedefewu fias?  
Or sou Data alMafes souxou ta nrapabondader sou osoepa? Ao dinabioocross?

~~Hadoop~~ Reducers

drop. 8.



Eisai ta nrauegova  
H14 nraudigna

(L) Map Reduce

Arto 15 han peca sou  
nrapku o jyos Big Data  
(or exer fiai nraue!)  
Eisai bounisifo sou Data.

To Big Data avoyepeda se  
nraudignis poulepes!  
Data auan se no nraies.

- # More CPUs { run program faster  
Faster everything! } # More Disks  $\Rightarrow$  Parallel I/O  
read files at the same time.  
or else there wouldn't be any disk to store all the app's.  
500 TB per 40MB/s of the app!!! (a.8 pages)  $\rightarrow$  bottleneck  
or over 10 seconds  $\Rightarrow$  10 sequential processes  
( $\Rightarrow$  parallel I/O  $\Rightarrow$  no bottleneck)
- 

More CPUs, more disks  $\rightarrow$  easier scalability.

1) Fault Tolerance: To work correctly even if one fails  
or two fail (in case of 100 CPUs or 100 disks)

100 CPUs, 100 disks, 100 RAM. If one fails  $\rightarrow$  to work  
 $\Rightarrow$  redundant!

One redundant copy over 2000 no matter what we do it's still fail!

2) Scalability: Improvement of performance

Same performance now  $\Rightarrow$  better performance

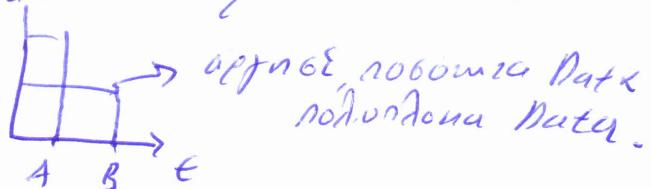
↓ Number of CPUs  $\rightarrow$  however less than 10000 executions X

↓ Number of disks  $\rightarrow$  however less than 10000 executions X  
↓ Number of volumes  $\rightarrow$  Data distribution to CPU  $\rightarrow$  to work better  
↓ Number of memory modules  $\rightarrow$  better performance

Grades of improvement for each of Scalability

3) Load balancing:

An app can handle two types of data so we  
separate bandwidth CPU A, CPU B



Separate the separate bandwidth on Data.

TBD 7-4b.

4) Efficiency : Andonborozuk.

22 3 opere, paf 100 noos. Algoceles duu ro probalit  
noo kaw au giccau nagaaliklos, awra ex 4 opere ro efficiency.

- Data Streaming processing → its xperation can now be online process of data.

Tradeoffs Accuracy / Speed.

- \* proper so probabilita va fi mai mare.
- \* mai multe date sau mai multe criterii
- \* faza de apelare nu este la 100% aprobata
- \* tipa de eroare este specifica sprijinimilor algoritmilor.  
(Performance guarantees)  $\rightarrow$  algoritmii trebuie sa fie tipa care are eroare mai mica de 90% aprobata.
- \* de unde se impune performance guarantee  $\Rightarrow$  de unde se impune performance guarantee?

Scale-Up → put more resources (CPUs, RAM, Disks)

φυλακών οντος ελεύθερος ποσού είναι robust.

- nu binește doară
  - nu binește de două ori pe ziore!
  - nu iau apă? de fapt nu răpește deapă

7 auto news  
6000 Dorothy  
700  
dinner for

10 Sciaridae sicc. exo. exo. exo (exodozina)

To Scale-Up darüber s. o. ein Beispiel (Anodierung) führt zu steigender Leistung, je größer der DC-Durchfluss

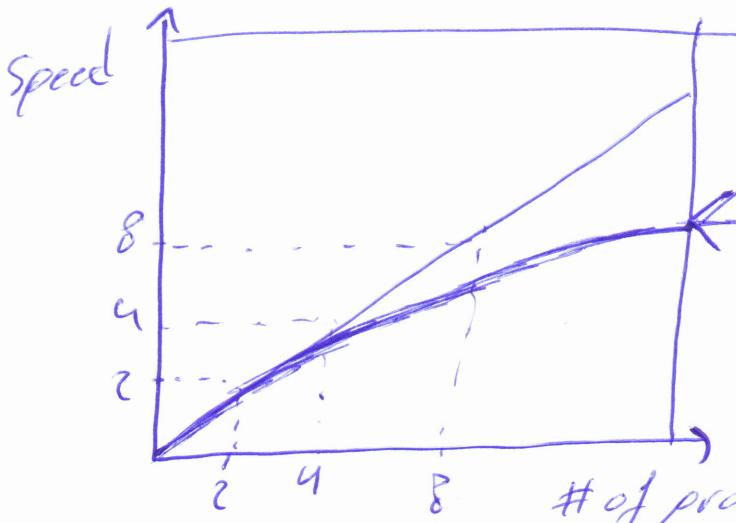
## Scale - Out

oxi eidejowres  $\rightarrow$  noiki xapenfsev huanster.

price for oikosofino hardware, but no 16x100  
over 100 hamster → require about 100 e.g. 100 x

no artigo } é interessante  
} nota sobre  
sentimentos >>  
apenas falar

glaubn wir  $\Rightarrow$  no, Max Fehler e)



awò n kòpòfin de òde  
nòde à nòde va nòde!!!

- 6 processors... jowais  
jazi? oxi kado Load Balancing
- oan ewu 10 CPUs dan  
fekilau va dòndafonu  
uswéjas, snapson bai  
coordinators
- ④ coordination overhead.
- ⑤ bainos Algorítmhos

- ③ CPU Failures & bad neighbors  
nópésben algorítmhos n' Algorítmhos  
ja enwagofora zwo overflows.

- 1) Start up cost
- 2) coordination cost
- 3) Show - Load Balancing. (o nègòs CPU bai lofipére eo xpoko euridéons)
- 4) Ensayagabia enwagoforow mochedofirou, ñeler supiano zwo le ñwan  
nawia nòpónnòdòmofò.

diag 15 : nws enwai eva Data Center aka cluster!

Rack : New hardware fe eva nòdòs mo nòfboj

Nòfbo : Gbifufa fe CPU, RAM bai ~~Disk~~ Disk

Aggregation switch : enwagofora o nòfboz zwo apòcòpos Rack  
fe bainos zwo ñefòs Rack.

Ayeuo 100 TB ñefòs Sizing.

20 xwopifòr eo dwe 62 Nòdòs bai baino nòpónnòdò Merge.

partition Data → Sort bai De partition → nòpónnòdò Merge.

nws ña pñacer owas CPU per ñer dñob } eo 80% enwagofor  
nws Apòtew ñefòs } ña eo gaw edò.

ñ ayeuo 610 ñefòs

TBD 7-9c.

## Fault Tolerance

Raparindo os bens de produc.  
↓  
ido prod. ↓  
desp. fixas + peças

I discos : da xaricou e o encontro é feito com P  
 $P = 0.0001$

ja 10.000 discos, nidanomia coodaxicos I ja exer jardine  
a 2x2 2 discos e da xaricou os 2 de ofertas ou da xaricou o  
xmas (dados)

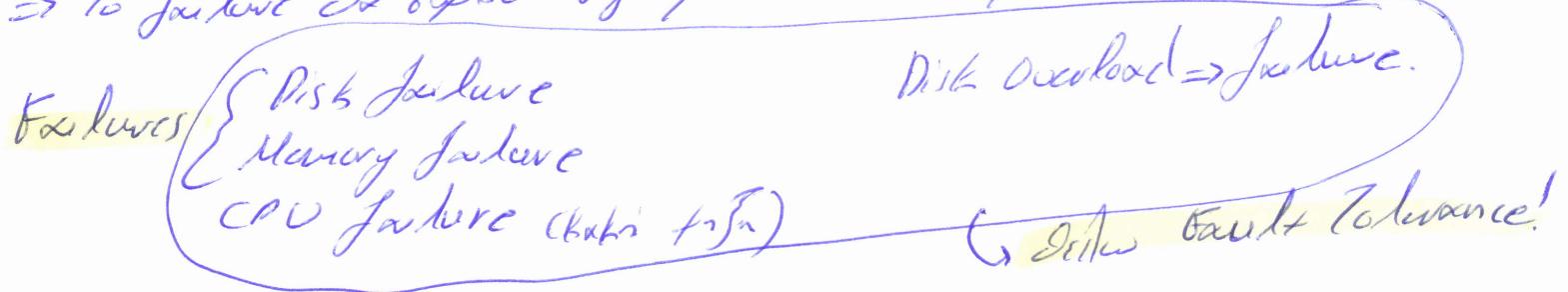
Yoodena : n nenhuns dos discos eram avaragamente Data

nidanomia do 1º xaricou unicos no nro D=10.000

$$(1-P)^D \text{ opa } 1 - (1-P)^D = \rightarrow$$

$\Rightarrow$  Prob{at least one disk failure} = 0.63 , 63% !

$\Rightarrow$  No failure da ofertas aqüo  $\rightarrow$  Aior: impossibilidade



To da know:

- 1) Simplest protocol: restart the job : se fizer spew ou em opção  
ao destruir o job pode ser preferivel excluir 4 repes. Failure?
- 2) Divide in datasets or software (split the job).

job  $\rightarrow$  tasks : propõe avaragamente datasets!

Node  $\rightarrow$  failure  $\Rightarrow$  divide in task allows {task restarting}

Hadoop: Hoxa es Data Center base Java que va  
toward saving (probablem op.) da una probabilidad  
que da importancia a la available  $\rightarrow$  MapReduce

Reduce  $\times$  el xénero secundaria

Hadoop: open Source!

MapReduce es Hadoop.

Aplicaciones en Google

- 1) Xusto Inverted Index
- 2) Cálculo PageRank algoritmo.

Inverted Index: na hache token nois achats se enregister.  
Leyendo = ordre se desg. toutes les web : register reading Leyendo!  
do date → 1 1 1 1 1 url pages  
non se enregister

Page Rank:  PageRank =  $\frac{0.5}{0.9} + \frac{0.9}{0.5}$  = 1.0  
PageRank =  $\frac{0.9}{0.5} + \frac{0.5}{0.9}$  = 1.0

Doug Cutting: "dagewis kan weergeven" → "loop"  
Een wa file system dat aanduidt welk pad de opnames van elke  
audiofile zijn te vinden.

Diag 2025: ca spătiva eisa avea noo obiecte  
"Endoop." HDFS: Distributed File System: pro modulare za  
modularizacija za spina  
Code fo endoop spe noo! foa pe ena idia autotilia u  
apna za propu za ca avansion! Propriu za za avansion

Hadoop Map-Reduce: edw ppaper co spjppata

Yet another resource Negotiator (YARN) manages the allocation and address space.

See HDFS proposal for generic nosql queries

TBD 7-4d

(2-30) (1000)  
Apiesekwontiki Hadoop : Master - Workers

ou neviosos Master ando do andher workers with  
hardware eou Workers  $\rightarrow$  jobs  $\rightarrow$  tasks.

Apiesekwontiki HDFS

Hadoop Distributed File System

O uade worker gera oce service : DataNode kai auoue.  
ihi geraspera eo diono eou.

Master koffari: NameNode  $\rightarrow$  DataNode  
ow dous pa permissions, nois eua o owner, nois kopen va  
spfer o hadias eber, nois eua eo blocksize (= 328 MB n 64 MB)  
replication level ?  
a) uapto apxelou ee koffaria  
b) replication re kai nois server

Ex: pugack Volume  $\rightarrow$  10 cras ee koffaria } aw io uade koffaria eua  
} modifkatio poso fia  
yapa, mupkes dous

Replicate: anafkewre noth i fia

default replication level = 3 (anafkewre  
3 diafoprenous workers)

NameSpace

Eper oti mu ierapxia eou apxelou, se noba voffaria eua  
broofrivo eo uade apxelou.  $\rightarrow$  awio idou boso Master.

Block Mapping

Eper now broofrivo modifkatio eo uade koffari eou apxelou.

Propriu ia exw 2 uadios Master, ya va proovimopar no failures!

In bache eva konwoprio DataNode grotworfka o DataNode ja  
tagfer us spes zos NameNode : öxi co anafkayo naus o  
NameNode eua uaplio).

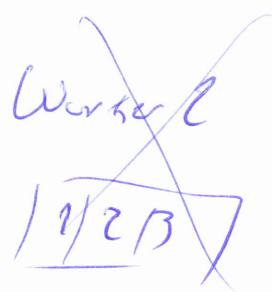
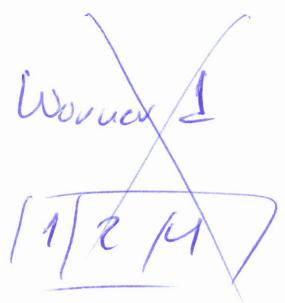
Ta bradawo avva no ex Data Node ou NameNode jra avva zo  
ago. O NameNode onda ex Data Node ooo duxawo bigingoa!  
badungpaga ou Name Node ooo bigingoa propose.

Napadejfa oft 35.

1	2	3	4
---	---	---	---

block size = 128 Mb.

node Worker  $\rightarrow$  Node =  $\frac{\text{avviofia}}{\text{osoftx}}$



Worker 3

1	3	4
---	---	---

2	3	4
---	---	---

Ekooyan apxios ou HDFS { 1) onaifia apxios ou blocus litanus  
4 blocus  $\times$  32 MB = 4 blocus  
zav 128 MB

replication level: 3: odu za block eivaw  
anakkefeda 3 yopeis.

A avva duxos neou za block ta vava replicate  
oddu.

Ciaw i rodatwaw  
baffar vo vava  
propaga, vada da  
nivei 128 MB !)

Sio napadejfa Mnogu ja duxawo E failovers { 70%  
workers !)  
 $\times$  va xadina ske  
Raum switch!

Napadejfa oft 36.

block A  $\rightarrow$  saved { Data Node 1  
Data Node 2  
Data Node 4

- o vana xponua duxawo
- o Data Node onda **beat**,  
na vo duxawo ou eivaw ok.
- o Data Node 2 den **corrupte**!  
 $\Rightarrow$  error HDFS

ape zo block A exi nidev replication level 2  
polis o Data Node awal yopeis ou o Data Node 2 de duxawo, tigun ou  
o Data Node 2 exi mol. co block A.

TBD 7-4e.

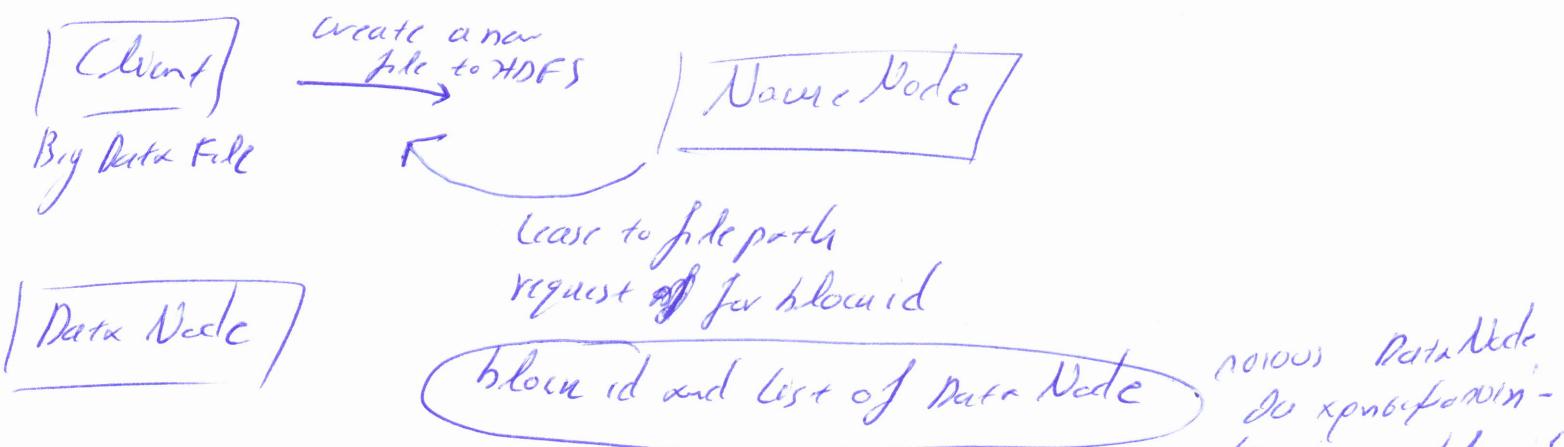
Ten o NameNode carrega pedaços de dados em todos os nodos  
excluindo referidos ao bloco A, no caso DataNode A.  
Todas as réplicas do bloco A são criadas no DataNode B  
(replication process for block A)

↳ replication factor : 3

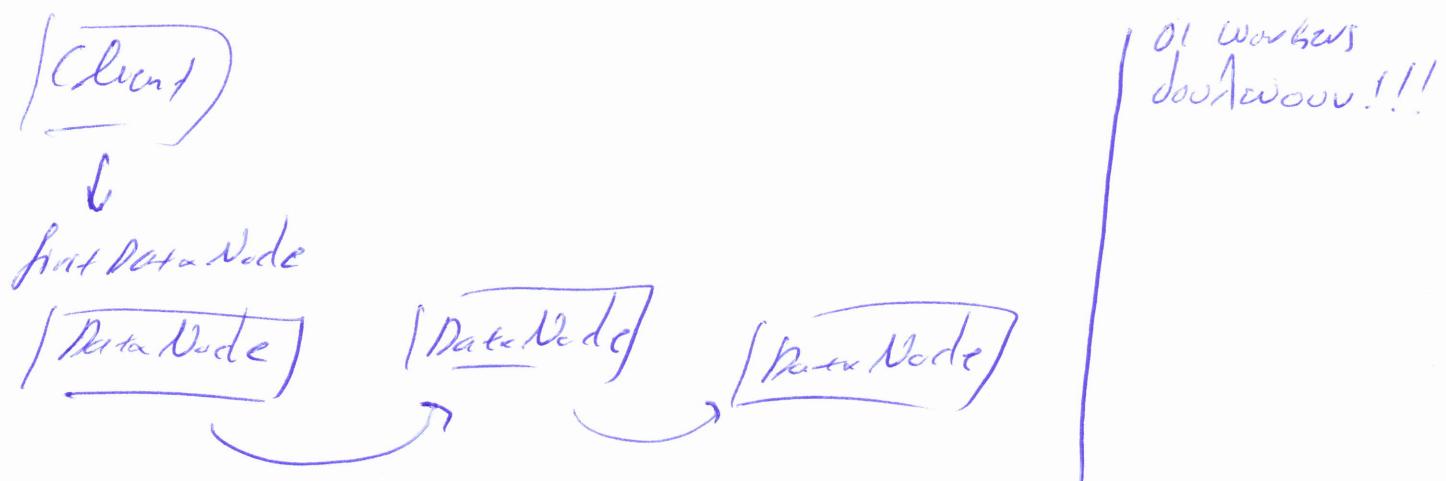
Olhar o 2º ponto online, Depois ou no Datacenter de mapreduce!

Replicação 37

avaliação feita no exemplo visto em HDFS



→ daí a ~~do~~ o DataNode e o NameNode da arquitetura são feitas  
por uns mesmos, o DataNode e o Client.



→ Nella offerta di Hadoop quale framework è più adatto?

Worker { Node Manager ←  
Data Node

YARN

6x) 4S → 4 Workers, 3 Masters { Resource Manager  
① each has 100% Node Manager  
↳ scheduling tasks  
② execute and coordinate their tasks  
③ no 200% Work Manager

② Evenando per  
to security.  
access control.

③ Garante disponibilità. Node Management  
informa di heart beat 200 Node Manager  
ovunque per disponibilità dei nodi  
(no timeout effect)

Master 1      Master 2  
NameNode      Resource Manager

Worker 1      Worker 2      Worker 3      Worker 4  
Node Manager      Node      Node Manager      Node  
Data      Data      Data Manager      Data

Aprotezione dei Node Manager  
Nella formazione di 600 Worker

Diagramma { Data Node  
Node Manager - Distribuzione CPU e RAM 200  
Workers

spesso 5 diag. tasks per sei containers.  
Java Virtual Machine  
self-contained JVM → run a task there!

TBD 7-4f.

εντ λεξικόρ: ανδ. σπάσου  
Fault Tolerance.

#### 48. Writing a Map Reduce Program.

Είναι αυτός οι δύο προγράμματα που θα χρησιμεύσουν για την λειτουργία της λογιστικής στο Hadoop. Την πρώτη μέρα θα διαχειριστούμε την λογιστική λέξης.

MapReduce 3 γιατίς είναι ελεγχόμενο 2 φάσεις  
3 γιατί shuffle-and-sort

Map() {  
reduce() } {  
δραστηριότητας και αναπαραγωγής.}



mapreduce: WordCount σεντεχ, δίδω word count!  
(word, # occurrences)

input: can you see the real me? can you? can you?

output (can, 3) (you, 3) (see, 1) (the, 1) (real, 1) (me, 1)

→ χρησιμοποιώντας λύση Map.

insert { first → add, → value = 1  
else → increment count. }

δύο σε  
βεβαίως δεν  
μονομερή σε API!

Dear Kofo του HDFS

3 splits

Hello there

Hello again

Hello to the world

This is small

example to explain

how MR works

task1

task2

task3

είτε μεταφέρεται σε buffer σεριες  
notes γιατίς αρχίζει με τον πρώτο word.

M<sub>1</sub>: split, εντάση task, split to 000000  
1<sup>st</sup> block

M<sub>1</sub>, M<sub>2</sub>, M<sub>3</sub> tasks, split to 000000  
ειδικότερη data.

output budekaa Map

M1  
 (Hello, 1)  
 (there, 1)  
 (Hello, 1)  
 (again, 1)

M2  
 (Hello, 1)      (this, 1)  
 (to, 1)      (is, 1)  
 (the, 1)      (x, 1)  
 (world, 1)      (small, 1)

M3  
 (example, 1)  
 (to, 1)  
 (explain, 1)  
 (how, 1)  
 (MR, 1)  
 (works, 1)

Applying crossReducers  
produces local file  
systems (local FS)

O Under workers applies Unix, has advantage also to apply comba  
before the final reduce task.

Shuffling phase

now do you understand  
our work now  
do have been done  
Reducers?  
 ↓  
 Shuffling!  
 per mod 2!

(again, 1)  
 (Hello, 1)  
 (Hello, 1)  
 (is, 1)  
 (small, 1)  
 (the, 1)  
 (true, 1)  
 (world, 1)

Reduce

R1

our reducers operates on idios  
they own the raw documents  
reducers!

Output  
 (again, 1)  
 (Hello, 3)  
 (is, 1)  
 (small, 1)

:

(x, 1)  
 (example, 1)

:

(to, 2)  
 (works, 1)

R2

$0 \rightarrow R_1$   
 $1 \rightarrow R_2$   
 $l(x) = (dx + b)$   
 MOD... MOD2

(example, 1)  
 (explain, 1)  
 (how, 1)  
 (MR, 1)  
 (the, 1)  
 (to, 1)  
 (to, 1)  
 (works, 1)

TBD 28-42

<sup>68d 59</sup>  
Flexxua Map Reduce Model: Oda pwoia (Key, Value) pair

(Key, Value): key must be comparable  
value bafis one deka, adda ooo reduce spesia va  
exa vonta eo aggregation pwoia uno value  
rapadejfa:  
pattern duz  
word count +

MapReduce { Map: uade worker duzkae ooo dizi zo Map  
tau rapadej output ooo dizi zou mitfaz  
eo modukazej zonua  
Ode ooo Mappers gerekko ooo Map  
Reduce: gerekko ooo idios keduas (onws ta ooo Map)}

Oda za 'Hello' pene ooo navi fusi → xpeno besti funzioni  
ta idio stand. da maa ooo idio reducer... exw en devatuzura da idio  
ooooo reducers deka, o YARN da kponi av da pwo eow dedesi!

Kividuas

diag 63 "Java ewas nglawoxa zo hadoop (corallis jiduas da ekono overhead)

diag 64 map() da doekya ooo pia vogelka ewes block  
ewa kawo zo tewe preprocessing

Ola jpayw yoffatta ce Map kponi ja kawo copy - paste aveo  
kaw miti ooo zo Map kaw nifodos modukazej ooo local FS zo Mapper

shuffling - sort : awapura den doekya egen!

Fault Tolerance

reduce(): "ewan oow oo kaw group by seg"  
zo output modukazej ooo HDFS

Fault Tolerance: driver - Master faults → bivoi - upbawu mania

worker faults (disk failure): unapxovo ulles 2 replicates.  
to diawgwo oow 3 kaw ewigyanow  
tanivou allav.

↳ kaw restant co task

benjau kawas

Failure can happen reduce() -> en error happens in Mapper class  
Siadischa: dataparallel auf reduce or reduce  
zu ergebnisse der Mapper sind bei HDFS fehlt.

Mehr nützliche Funktionen da es einfache ist zu implementieren.

## Auf die xeroriparator Java, xeroriparator Python

Mapper.py, Reducer.py scripts, {Standard Input  $\rightarrow$  native spezifische API  
Standard Output  $\rightarrow$  Protokoll}

da müssen wir alle die Funktionen  
da müssen wir den streaming Java  $\rightarrow$  Coordinator / translator  
zu MapReduce aufrufen in Java!

day 25 no programming  $\rightarrow$  Other Frameworks  
(Apache co MapReduce standard)

## Spark

häufig für analysis oder Data. OLAP (online analytical processing)

day 4  $\rightarrow$  2.4.5 version

$\rightarrow$  Unified cluster computing

Programming Java, Scala

Python, R

Use for: Test designing distributed Algorithms

$\rightarrow$  YARN an den Ideen der xeroriparator, nur zu HDFS!

$\rightarrow$  Zu MapReduce kann direkt sorting (no shuffling & sort)

day 8  $\rightarrow$  Spark nutzt xeroriparator in Main Memory } für große Anzahl von  
Mapper und Reducer zu MapReduce } bedarf an co-native MapReduce.  
oder nicht optimierung How?

## Spark core: basics

libs: libraries

Applikation: Input ~~Processor~~ resources  $\rightarrow$  Output  
und ausw. Datenbank.

Cassandra, Hbase  $\rightarrow$  nosQL unterstützung  
spezielle einfache API Read/Write

Cluster Manager: Tonua spogowis de xerencia opus ja va eou  
xerencia (nx ja hukum) co spark xerenciadores  
(cluster managers (a tawu - uau))

Standalone } euan o default Cluster Manager  
Scheduler } eou xerenciador oupe ja testing

~~SQL API~~

**Libs**: official Spark libraries

**SQL and DataFrames** } opixu SQL querries oc doku noo!  
euan nx oc CSV, jeniuoc euanifyfica data  
to spark better optimization via querries bai spikovo rengopa

DataFrames: eas moduus ee Database

**MLlib and ML** } Machine Learning Libraries  
classification, learning, Neural networks  
avudon eua data (nx sensitive analysis bai tweet)  
(tweets → quaxas Model → dixiu veg tweet  
bai bauu avudon)

**Streaming** } data noo addjuu real buku, bai detai va ea avudon  
Spark → bai bai input oc koppina → ea avudon  
discuodos hananuus prograxtunatos

**GraphX** } data noo euan graxifaza  
graxoso graxifa → bai bai bai cluster } exi bai bai graph  
mapu auxil, sortbas  
gricos bai...

**APIs** } to spark querencies ee Scala  
java: euan nidoorpuen uno en Scala } Scala & Java euan awia  
unocepsa python & R pe neipropofu euan bibliodinuss

bauu n Scala enu functional nuffern, euan diajoperium!

diag. 12 Napadeza

Hive Context: analytical queries

diag 13. Napadeza μελε & ML λογισμοί για Machine  
logistic regression → χρησιμότερη από baseline για classification

diag 14 Graph napadeza για τον PageRank: είναι general purpose.

GraphLab: γράφω προγραμματισμό C++

GraphX: γράφω σε Hadoop Map-Reduce

## ② RDD API - Resilient Distributed Datasets

τελευτής προ οποίο νοεί βρέθηκε σε cluster  
μητρική στοιχείων Metastore

Είναι Read Only διά ποσού update αλλά το παραγόμενο σε γενιά ή άλλα.  
(καταχρέωντας update, ήτη Insert, ήτη Delete!)

RDD: που δίνει γραφεία σε όλη την αρχιτεκτονική παραγωγή (πολλοί γεγονότα)  
Creating RDD } { Transformations  
transforming RDD } { Actions  
actions in RDD

### Creating RDD

HDFS αρχείο 100GB ↪  
γενικών RDD για να τις χρησιμοποιήσουμε

γενικά για ως  
πρωτότυπος σε όλη  
μέθοδο.

Vul hdfsRDD: γενικών RDD του datav ου προσδιορίζεται σε διάφορα γεγονότα.

local RDD: local γεγονότα

π. οι γεγονότα σε μία προγράμμα.

list RDD: n θέσεις σε κανονική πορεία.

TBD 28-4-0

"To Spark war es nur ein feind!"

## RDD operations

{ transform  
actions

$X \Rightarrow X.contains("apple")$   
Scala ist funktional. Dazu zu X die entsprechende App.

reduziert zu einem Wert, nützlich für die Vergleichsoperationen.

zu filtern führt zu Transformation.

→ Es ist wichtig zu den Methoden zu wissen, die Spark verwendet um sie nicht zu langsam zu machen!

Day 24

→ val numbers = unionRDD.(count()).  
eine action oder transformation  
zurück da sie zu Spark

transformation: mehrere input RDD → output RDD  
action: mehrere RDD → output kann -> ein RDD. Fazit?

Auch ein Goodwill  
es kann nur 11 Zeilen von  
hieraus prozeß zu einer bestimmten Anzahl

- unionRDD.(take(5)).foreach println) Nur einige wenige Array

action  
entwickelt 14  
Zeilen 5.

To Spark fügt die first 5 Elemente  
Folgen zu UnionRDD zusammen 5 Elemente  
da Gruppierung (gruppen von den 1000 rows)  
verwendet werden.

→ nur eine kleine teil → benötigt weniger Ressourcen zu xpozen

Lazy Evaluation } → To Spark benötigt mehr  
Ressourcen zu einer action.

## Day 27: MapReduce Transformation

"Scale your workers per  
row instead of per Data"

### Filter & Map

Input: MLE oso mapreduce + 2

distinct: powadina orokoa con RDD

Sample: Jaffer con 1x oxi 100GB  
nape puro 1GB

FlatMap: eduk nape oso x bin eureko

Table: va piz gara bako uox nou  
etwu oda dantza oso deska  
0.2 → 20%

Nicue nou da nspire oso x, xis  
bin x-1.

to flat mapper zuo dibus en mapreduce  
puro piz (ya oso de xnpotxu  
enzo map)

3<sup>rd</sup> MapReduce → seed propri de datus  
Va enmaper nraia eo ido deska  
(std random seed 3.42 emapre de  
ceodondio mo deska)

to flat mapper zuo dibus en nape  
estament.

testing algorithm baso ido testing  
data

textFile: transformation

transformation fe 2 RDD (union) intersection) subtract(), cartesian()  
(join)

### Actions

count(), take(), top() / za fgarazipa orokoa con RDD

countByKey() / enmaper key, value pair → an word count.

ra nraia enmaperpoiora a Array (fgetedo?)

collect() enmaper o Array oso de RDD, nro da nra oso libra memory  
propria oso mapreduce oso  
fgetedo, aplo ncollas oso  
eva pizno. Diket  
nraia!

### Diagrama 42.

o uniduo eus unan da spesia 600 Master

(Workers)

600 Wdoo oso Mappers han Reducers, 600 Spout etwu executors

3 mapReduce partaxipara => 3 executors

textFile } transformation | count → action! → ports oso binnurra o driver  
filter } transformation | count → action! → All binnurra workers el da unan

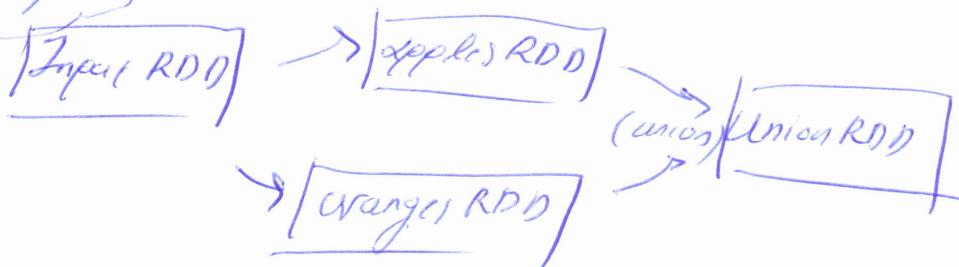
o worker gorixoa filtering o auxiliarra partition  
ta opredo. Eta enmaperpoiora ta count oso Master.

TBD 5-5a

Spark API { RDD  
Data Frame

Java code can be metastable  
avoids overheads and encodings of the RDD  
as no partitions are needed.  
May be safe when reconstruct to RDD you  
will be leveraging some of the code.  
→ now can avoid RDD and Metastable partitions.

Dataframe



good behavior!!!

lazy: takes many arguments in job → optimizations!!!  
and does not estimate the action → Spark job.

Diag 43): Reasons for map bugs in code:  
many to propagate across workers.

- Spark execute executors (like Map Reduce) Map is to persist partitions and not the Worker.
- Executor = ~~Java~~ JVM (represents one worker for one process)  
one worker has one instance of local FS and holds one partition (partitions ... 3)
- Maximizes the reuse of Workers → Workers.
- Order of the Data Locality = first no shuffle and then, second no network traffic

Diag 41) executors have one task (multiple threads or separate job)  
Worker has 8 cores → 2 executors or more on Worker  
so a single executor has 4 cores (maximum 4 tasks!)

- Flat executors  $\Rightarrow$  executor / fixe for a specific task over repous
- Flm executors  $\Rightarrow$  multiple executors  $\Rightarrow$  independent executors
  - as fixe for ta anopejw, manu apocifura, post, fadu  $\rightarrow$  bad components

Driver bawas : one task of 3-5 repous jne uade executors  
 Rule of thumb :

One driver enau zo Sport Session (zo gateway opis rov obiecte) zo  
 xperforasi. Driver jne va uade kartuifka exzeguraria.

**Scala Functional Programming:** Tu nisce enau functions  
 si je endappon 100 enau zo Data

reduce()  
 $(x, y) \Rightarrow x + y$

↓  
 anideka reduction  
 enyoppon puro anideka  
 kisa ox aggregation.

$\{1, 2, 3, 4\}$   
 $\{3, 3, 3, 3\}$   
 $\{6, 6\}$   
 $\{7\} \rightarrow 7$

Map ( $x \mapsto x + 2$ )  $\xrightarrow{\text{overset}}$   $\xrightarrow{\text{overset}}$

propi va opis pra wadepens (nodelets) han va en uadek poko  
 bto alap. (exw no evolujuwo kudina) un opisw no roun 600 code.  
 exi pagatiget efseidua.

Map  $\mapsto$  1  
 flat Map 1 pos Nodelet  
 flat: wadijases Node (x)  
 anopejwida.

REVIEW

TBD 5-56.

### 30) key-value RDD

(key, value) pair

- implement Map-Reduce

- same concepts as before

- key must be comparable (upfremd und adaptiert)

#### - join

join from 2 rdd

n join join of 2 rdd.

either both values now available or  
one of them

left, right, full - outer join.

~~DEN even action -> own key~~  
odd own key.

transformations day 30  
development of RDD now done at day  
- groupByKey: possible to do key  
by us first in basic or  
via dict.

return (k, Iterable(v))  
allowing graph  
in scala

return to user after  
first option! does  
workers!

#### - reduceByKey

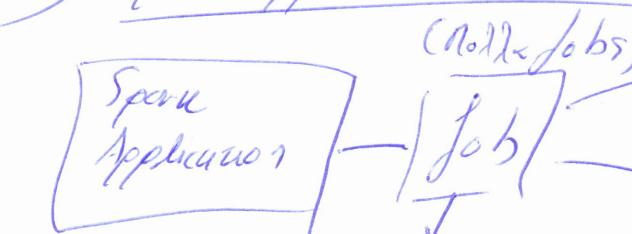
Map the key to the values.  
join two nodes together.

map (k, v) → f(...)

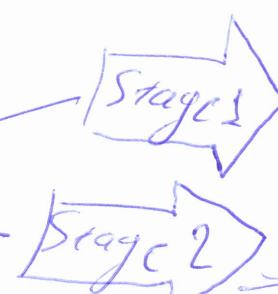
or add to each other for reduce

den xmap function group by, map to  
reduceByKey via Network traffic.

### 37) Spark Application Tree



multiple jobs  
additions and  
transformations now  
available per every action



avvia trigger da  
operazioni da applicazioni  
Mappe stage da destra to  
output con stage 1

edit due  
rapporto fino  
Workers  
cavalcando  
due due  
tasas cuando se

→ 1 partition → da due separate jobs

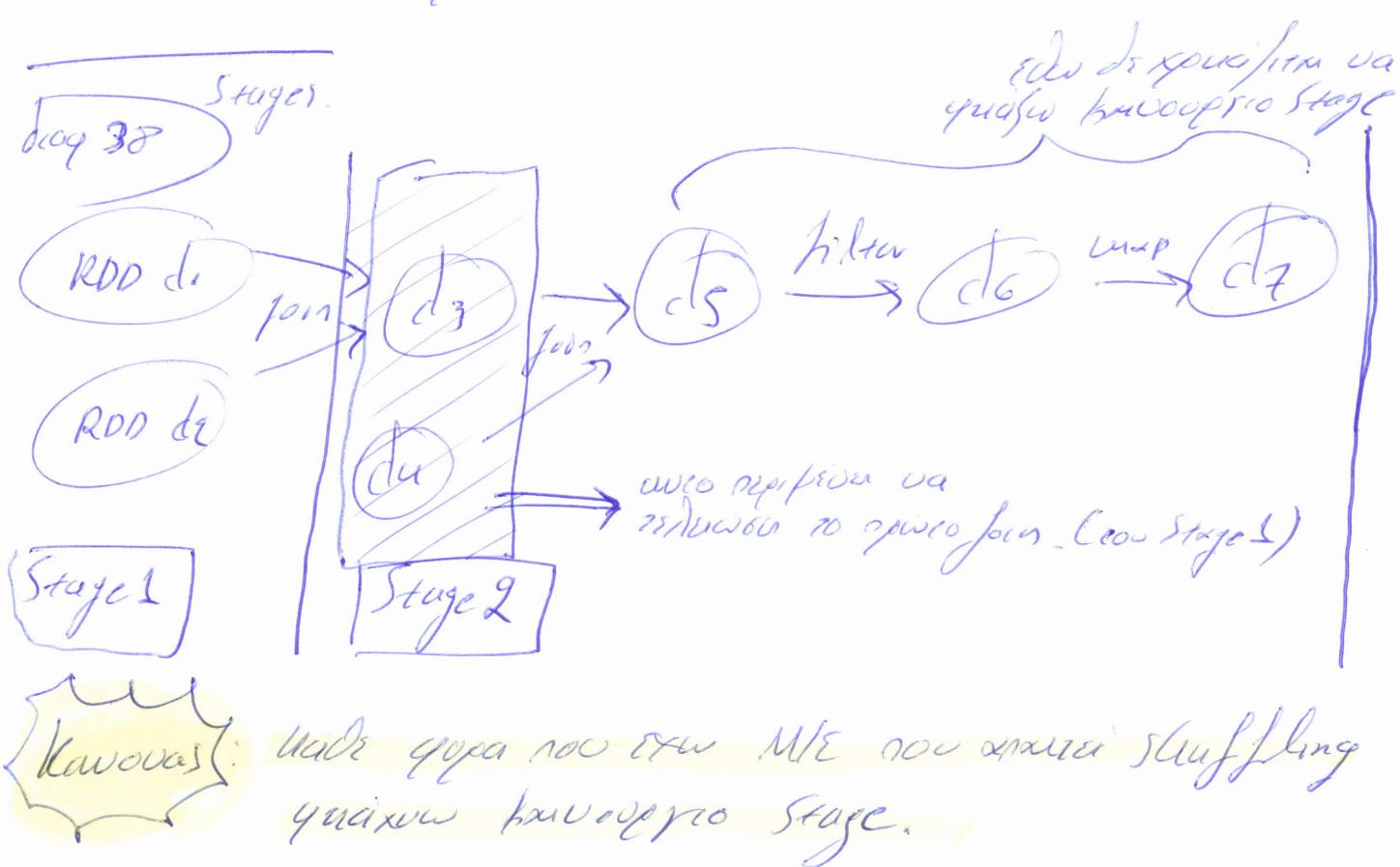
→ 1000 partitions, 10 workers ✓  
o una de ejecutar la aplicacion

replicación en 1 partition

avanza n partitions! Basan Marca en este. Spark = Partition

en 1000 partition exec  
avanza 1000 workers, han 3  
particiones da dividir en 01 3  
workers!

- ⇒ De nova partitions da xaréus ro apxaro?
- ⇒ ro file vivu coo HDFS ro file ema nida xaréus ro  
partitions. Apa fórm nida writers da xaréos.
- ⇒ H Text file proprio da desidera roso rrofego ro nida  
xaréus partition nida desidera.
- ⇒ → SO, anfórm nida bloca ema xaréus ro file coo HDFS.  
(read Partition Merger Function)
- ⇒ dea ligas berduanomas amais co nida em partition.  
nuxuas hantageas runtime.
- ⇒ Ben agnouope ooo sporu OME ns berduanomas!



TBD 5-5c.

Diag 36

## Narrow or Wide Transformations

(Narrow) Dependence

child : RDD nov quaisquer output)

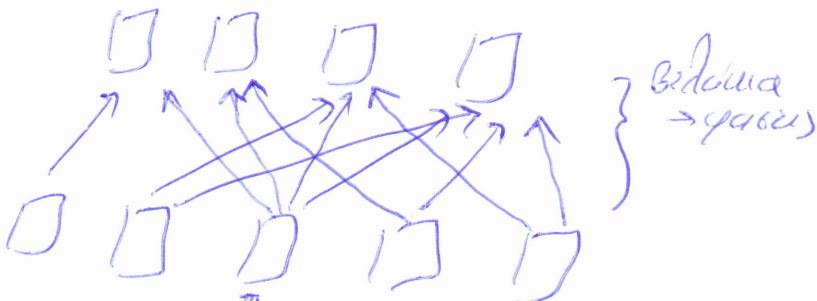
Parent : Input RDD



→ wide partition RDD parent stages are now in one parent partition.

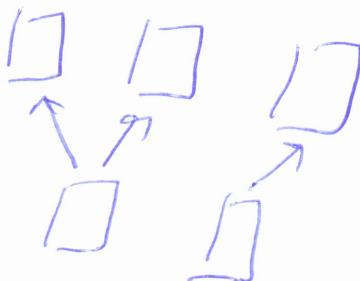
(Wide) Dependence

- no spurious redundant info



avio espaco no ORA!  
data em ods stages!

na repartiza da proposta  
va unica tlp. (infiltros)  
o Executor não de atribui co  
PI da quaisquer C3 apa  
ara o 2 fom os os ido  
executor. n.o.4 p4 C8, C3



bi avio downliv...

nx join(): Transformations

(1 espaciam poco no PI (rewire!))

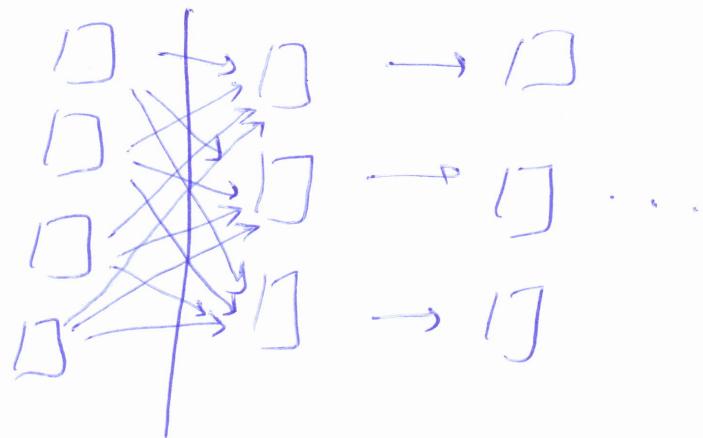
nx SortByKey()

ta repartiza se C3 de duas fases de  
esta fose no PI! To Sort deles Data  
em uma partição...

→ em wide dependence, ix (shuffling) → processa deixa  
para o stage operation  
executar.

→ ta wide dependencies hadoop fara ta stages.

Day 39) Repartition: New groups to stages ...



Day 45) (diagramm veranschaulichen)

Stage 1: Derkt feste 1 Stage!

val result = rdd

\* filter( $x \Rightarrow x > 100$ ) // Narrow

\* map( $x \Rightarrow (x, 1)$ ) // Narrow

\* reduceByKey(- + -) // wide! Stage 1 Stage 2

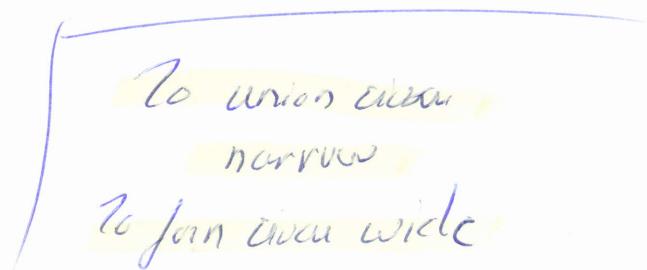
\* filter( $((x, y) \Rightarrow y > 1000)$ ) // narrow

\* sortByKey() // wide! Stage 3.

Muss bei jedem Stage in fester Anzahl Phasen!

→ Day 46) galoren Repartition!

Mappe von narrow visualisieren zu stages, mappe von repartition.



TBD 5-5d.

2) Η αιροφόρα καινών φερά στην επικέλευση  
της αερού

## Persistence

sisteme  
"diarofw" co ROD un raportare, a New Jo magazine

```
var result = rdd.map(x => x + 5)
```

Println(Count().Count()) **Action** now va quejar el resultado  
now va quejar el resultado ... u.o.u.

```
println(result.collect().mkString(", "))) // yoda da glosada noo RDD
```

(Plano da  
experiência)  
em profa!)

W van en op de roest

now vaqueta co red ... k.o.k.

most evil acts

to result in 10 ips. Oct 10

red for the new version

*persist!*

result.persist(StorageLevel.DISK\_ONLY)

Ask Sparu  
to keep (persist) the Data.

report Iyoryoo so → MEMORY\_ONLY (exception error)

Second Storage 110 case MEMORY AND DISK (includes software n perfon 1021 600 Disk)

10 val face pacifer

to result.

→ Aho de <sup>9º</sup> acción completa, la acción de yeso no se aplica.

$\Rightarrow$  result.unpersist() [Händler für xref  
esu zu "agnose"]

## Broadcast Variables | Boosting Efficiency.

# Broadcasting

Nous co-Opéra en tant qu'en Rata les meilleures exécuteurs?

43) Create a function to do whatever  
or modify the do whatever.

| *Богов да  
бездна у ада...*

Order va fonda av  $x \in S$

Tx RDD etotal oem 11.11.2019  
executors

$S$  [ 100MB ]  $\left\{ \begin{array}{l} \text{funcion:} \\ x \Rightarrow x \in S \end{array} \right\}$  rdd.filter( $x \Rightarrow x \in S$ ).count()

useMap.

Driver  $\rightarrow$  Worker.  
rdd  
filter function  
Count  
 $S$

# Localiza pro goppa ja  
neste partition  
100 partition  $\rightarrow$  100 goppa  
1000 partitions  $\rightarrow$  1000 goppa  
 $S = 100MB$ . Myndo  
return traffic

$\rightarrow$  Okt zo  $S$  va dividir pro goppa ja neste executor.

$\Rightarrow$  Da zo worker broadcasting wort va nise \$ goppa ce neste executor

SQL

val bvector = sc.broadcast(vector $S$ )

rdd.map( $x \Rightarrow$  bvector  $\circlearrowleft$  value  $\circlearrowleft$  contains(x))

Xplicativa aveo ja vo enfeitor 100  
executor va xproponimosa ro broadcast, sao  
zou exar ordena no spiv.

$\rightarrow$  A new grupo ro vector rows va pro spisa va zo uave broadcast  
adda jenuna onda basta spacioi

Accumulators  
Shared values

$\rightsquigarrow$  Write Only : Executors  
Read : Also o Driver

Oi executors proponeva univeral add owo accumulato.

$\rightarrow$  Des proponeva univeral read.

raci?  $\rightarrow$  7pessoas 10 executors sara cluster kia exovo store pro fechar  
uma povo ou 60000 add. On read. Read de jenuna raci eliva depa  
bujxpoingos!

TBD 5-5e. Accumulator X [ ] → queued spouts

54) val counter = sc.longAccumulator("counter")

o executor bower add 6100 counter.

da proposeer da new data toe va houw appende om Africa.

Daayopaa pr thounds: za thounds itwoon nooit prijs  
na executors dat exoor!

## DataFrames

Eva DataFrame exce RDD, addtixu extra spoforce  
awaan ew distributed table : pe rows, columns kwe erithods.

→ DataFrames are Structured & query SQL queries  
analytical queries.

58) val df = sparkSession.read.json("....people.json")

→ gerakwoope df are json, csv ... wa nalla & Ma.

→ Anqur va opies ea oofaca ewo berikus, n va za new vendue.

df.show() // show just few lines of DataFrame.

df.printSchema() // shows headers iwu, a duxwa n hale oefdu.

59) <sup>105 zoonos</sup>  
df.select(\$"name", \$"age" + s)  
↓  
Eva MIE →lazy  
gos zoonos

sparkSession.sql("and SQL query") → iwu 'people' awo den oofaces  
nauw. Ewwa co people we co  
dataFrame!

1<sup>st</sup> option

Sql query = `df1 = spark.read.  
sql("SELECT * FROM source  
ADD AS new_table_name SOURCE")`

Adds source code  
before compile

→ Error due to ~~compile time~~

are addressed by using `fixed`,  
now due to sparsity is the  
advantage.

2<sup>nd</sup> option

`SQL("string")` due to  
query parser is in database  
not in application!

adds to application due to  
source code.

It has to read more lines...

due to `addSource`  
application due to application.

→ Error due to runtime.

disadvantages are  
due to fixed no more can  
be done after due to

## 64) Word count in Scala

`val local[2] : array[map[2, rdd]]`

`val txtFile = sc.textFile("inputfile") // RDD`

(Map) Map due to rdd. Now due to application of reduce.

Each reduce, so split into many reduces!

.map (word => (word, 1))

.reduceByKey (- + -) → reduces sparsity of values now it has to take key.

.save ...

advantages wordcount!!!

TBD 12-5x.

10 hdfs exec Web Interface!  
Hadoop Spine Hands On! See apx 10 hadooplab.pdf

- Start-dfs.sh // grixer o HDFS, e om ifuncionar cluster.

- Hadoop fs -ls / // root directory to hdfs

hadoop file co  
HDFS.

user: unique identifier or users  
hbase permissions, ownerships...

- Hadoop fs -ls // home directory. http

- Hadoop fs -put local file, co and file to hdfs

- /leonardo.txt llll.txt

- get gracioso (da naa co base directory.  
super!)

- cat // correr comando cat, piso da expo dopo principio...

- tail // numpa digo no se redos con o apxios.

- Loo hdfs sw ~~on~~ en permanencia co ~~overwrit~~  
overwrite bca qoxera.

replication level  
=3

- Hadoop fs -rmr output // quitar directory no co output

- rm llll.txt // desaparece co llll.

ano hdfs : DataNode  
NameNode

ano garn: NodeManager  
Resource Manager



TBD 19-5a

## Καρδινάλια Σταθερότητα Διαδικασίας

### ① Επιδιόρθωση Transaction

Ένα σύγχρονο εργαστήριο / λογογραφία που αναρτεί  
πρέξει | αναρτώνται  
| απελευθερώνονται  
| διαγράφονται  
| αντικαταστάθηκαν.

→ γενικός ουαλδάγιος επεξόρυξη

### ② Concurrency control

πρωτικός επεξόρυξης "μετατίτλων" ουαλδάγιον

### Agradis (αρχαίος):

πετα και βανδιά failure σεντ στοιβάσε (disk, power failure)

### ↳ δεδυτικό recovery:

### Replication

database σε διαδικασίες αι 2 πολεικούντων

αι 3 γρεκούντων σε μέτρη τη PEYMA.

σαρά με επεκτείνεται σε αρχεία σε local file να είναι

notes στην αρχεία  
αι 8 λεπτά

τι να κάνει;

↳ διαρροή σε  
επεκτείνεται

διαρροή σε γνωστόν  
κύριο

κύριο

σε υποχρεωτικόν

### 5) οργανός ουαλδάγιος

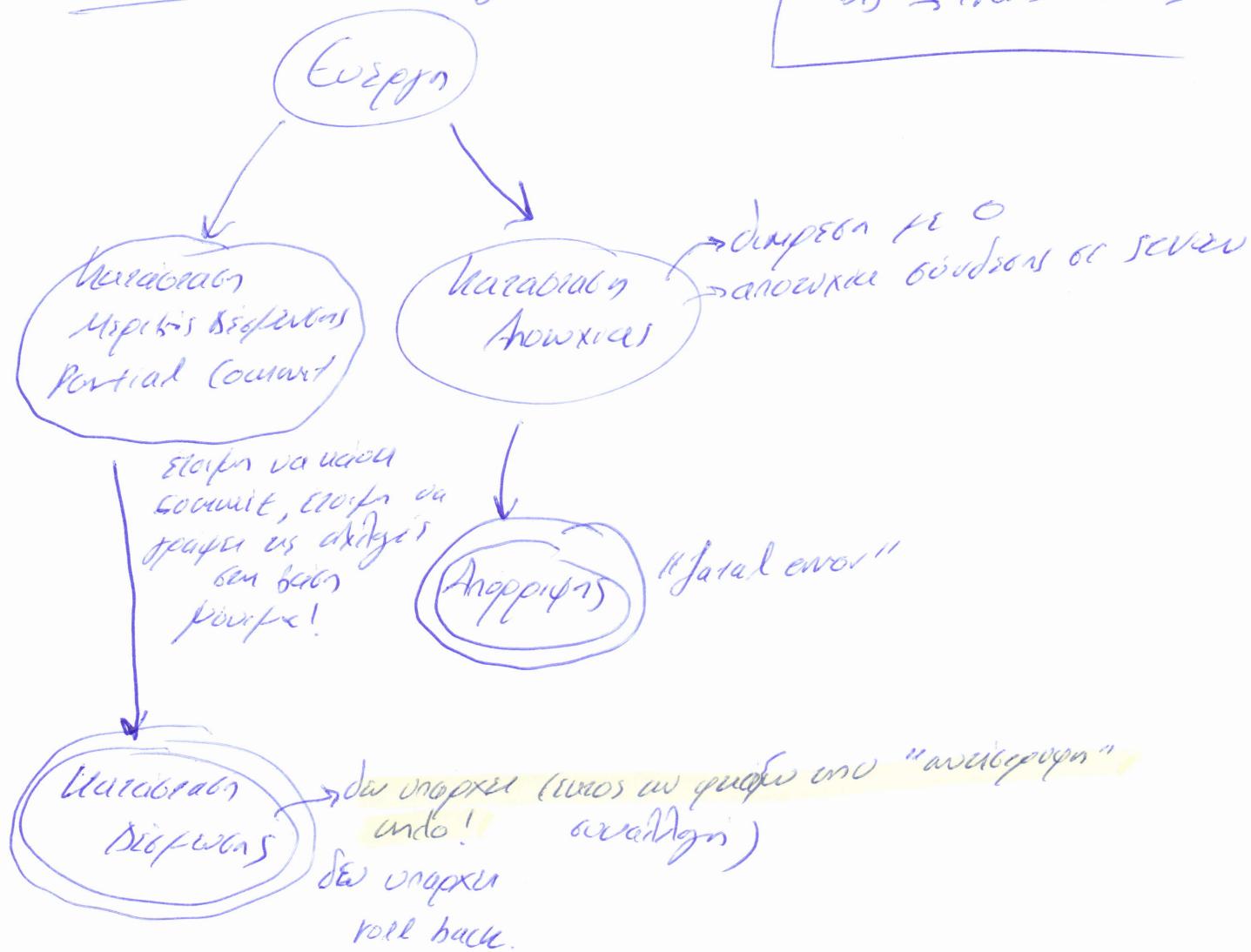
πολεικούρα επεκτείνεται γραφικά

που είναι γραφικό σε high level language

## • Karaciobes lvaAddgw

OS → processes

BB → transactions



## • Idiomas lvaAddgw

- Acoplamiento: un par de entidades anexas da nombre al IDN
  - Anotaciones: Guadagni anotaciones que alteran el tipo (de tipo no es alterar tipo)
  - Movimientos: ordenadas cumplir => posibles cambios  
(durabilidad)
  - Coherencia: las anotaciones se cumplen en la misma forma  
(consistencia)
- Cada ente tiene sus propias normas!
- No existe una norma universal!
- Excepción aparte

TBD 19-56

OS → resources  
DB → data

#Eureileen Luca Moga

- Lepidaria

Εναρκτήσαμε την παραγωγή της από την Ελλάδα στην Ευρώπη! Διαβάστε την παραγωγή μας στην Ευρώπη!

депиаці  
хроподжажаффа

- Taxonomia

ευθέαν μήτε περιπλέγμα επιστρέφοντα

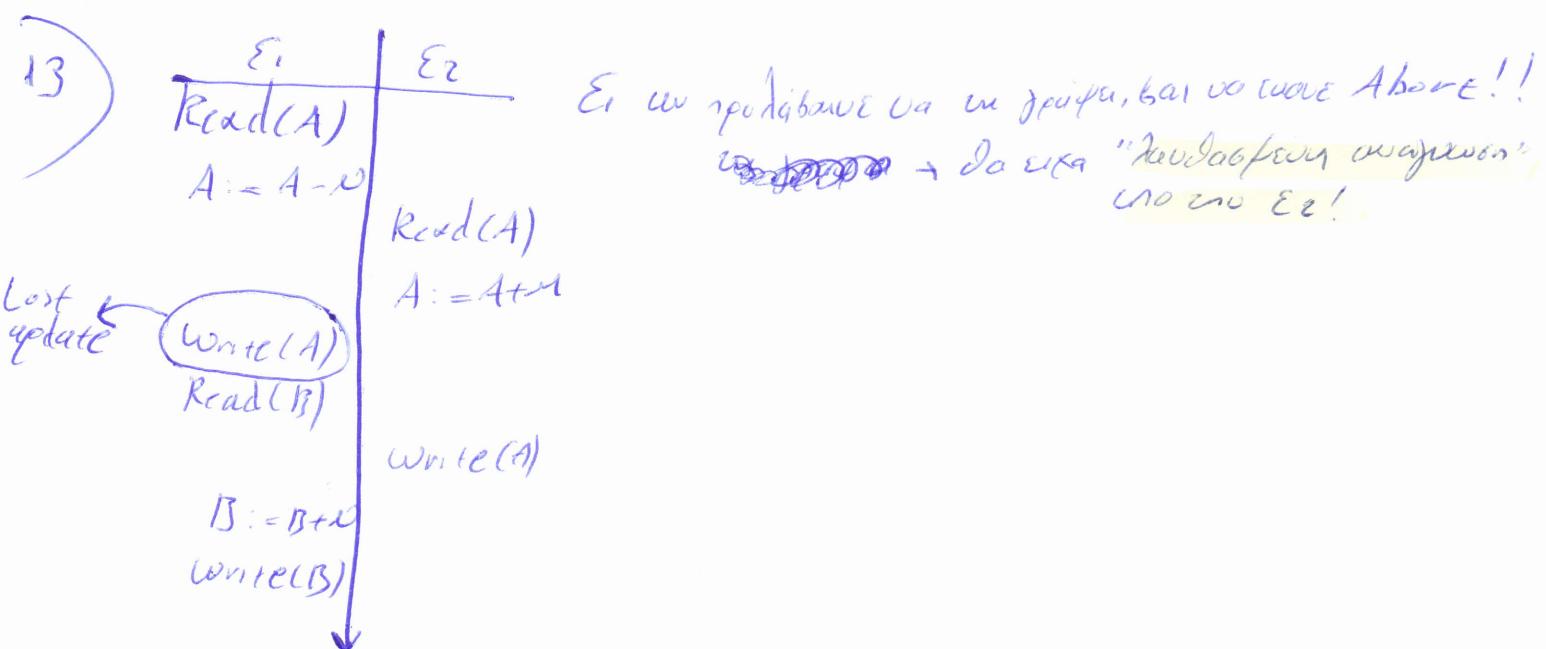
++ waiting for

- ελεγχος κωνοκρουνησ

## *# Elegchos Tavropoulosoi*

2 Xpoxes eozoxpoxo Write below idia paffi!

(spores produced on old NB → new spores)



## Ο Μηχανισμός Ηλεκτρονικής Λογισμικής Locking.

Ηλεκτρονική Λογισμική που περιβάλλει την αρχιτεκτονική ενός προγράμματος σε αρχεία δεδομένων.

Επινόηση } Βιδεύεται όλη η λογισμική...

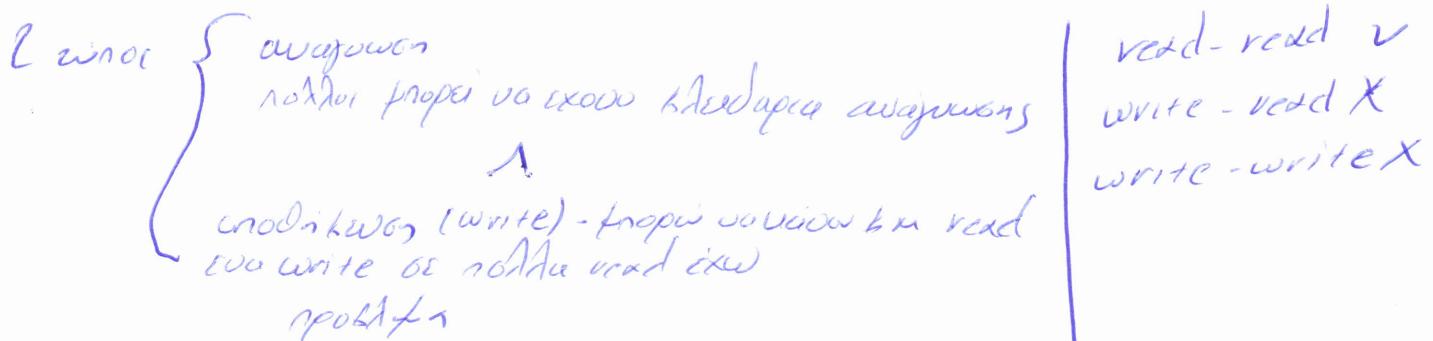
Ηλεκτρονική Επινόηση!

Ηλεκτρονική Σύγχρονη!

Ηλεκτρονική Κατανομή!

} προστασία

→ Επινόηση λογισμικού ταξιδιού.



## Μηχανισμός Διεύθυνσης

- αναγνώστης → Ηλεκτρονική προστασία (όποιας λογισμικού εργάζεται)

- επαναλειτουργίας → Ηλεκτρονική προστασία

{ οντωτική προστασία  
    παραγόμενη από ηλεκτρονική προστασία  
    οντωτική προστασία που προστατεύεται από  
    οντωτική προστασία που προστατεύεται από ηλεκτρονική προστασία }

→ Διαχείριση αναγνώστης και ηλεκτρονικής προστασίας σε αρχεία

→ Μονίμη αλλαγή προστασίας σε αρχεία

- upgrade, downgrade.

TBD 19 - Sc

Negacyclophic Locus

Mo zo Lock  
moga jor banana  
prob. Impara. OX!  
OAA!!!

	$E_1$	$E_2$
Read(x)		
$x := x + 100$		
Write(x)		
Read(y)	Read(x)	
$x := x * 2.5$		
Write(x)		
Read(y)	Write(x)	
$y := y * 5.5$		
Write(y)		
Read(y)		Lock - WR(x)
$y := y - 100$		Read(x)
Write(y)		$x := x + 1.5$
(X)		Write(x)

$\rightarrow E_1 \rightarrow$  Abort  
Delete Undo aus  $E_1$   
Kann  $E_2$  die Basis  
Abort sein!!

	$E_1$	$E_2$
Lock - WR(x)		
Read(x)		
$x := x + 1.5$		
Write(x)		
Unlock - WR(x)		
Lock - WR(y)		
Read(y)		
$y := y * 5.5$		
Write(y)		
Unlock - WR(y)		
(X)		

$E_1 \rightarrow$  Abort } To loca der sperrwerte  
Delete Undo } GE reprimieren short!

1) Maximales Blockausfall 2 process  $\rightarrow$  auto erneuerung  
bei Ausfall!

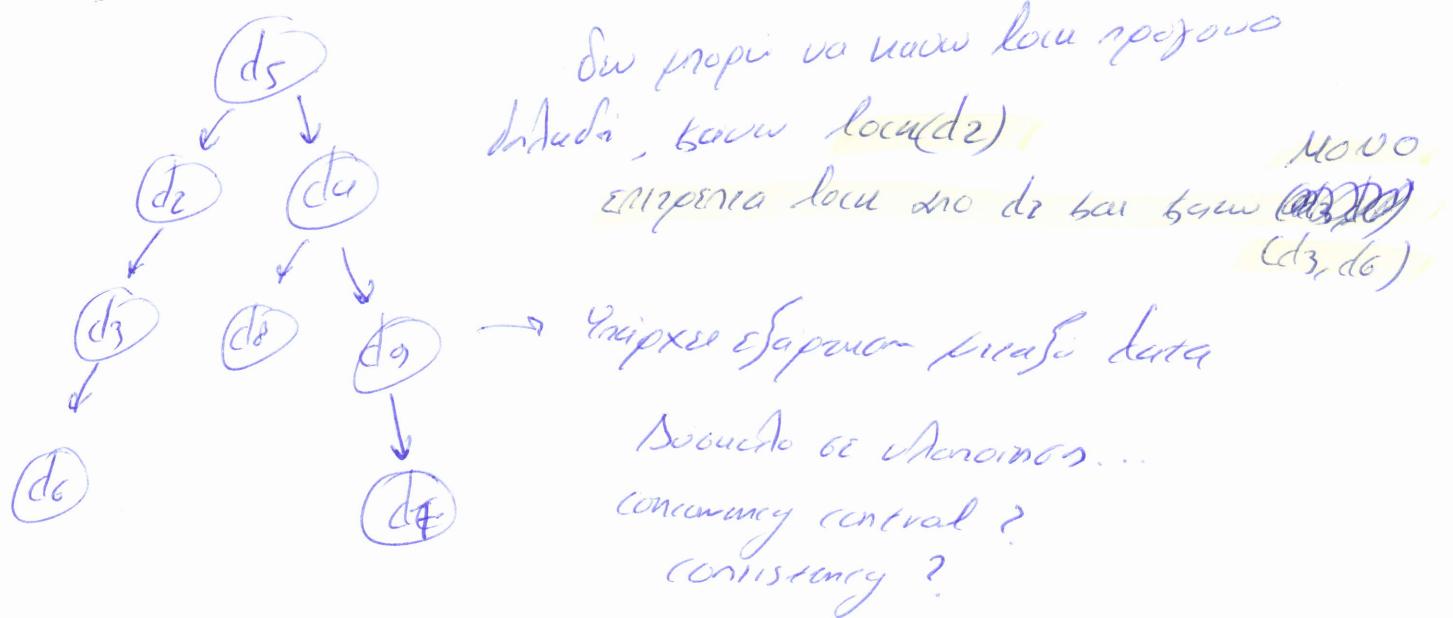
-gute Auswirkung: nur während des Fehlers Blocksperrung

-gute Auswirkung: nur während des Fehlers Blocksperrung

$\sim$  Problem explizit einfließen - Occurrite.

	$E_1$	$E_2$
Locu(A)		
Locu(B)	Locu(A)	X
		X
unlock(A)	X	
unlock(B)	V	

2) new constraint about race (graph).



### Maximizing Adhesions Deadlock

Even if we want to avoid deadlocks, we do  
not have to implement one after another to avoid them!

Two types of adhesions → barriers (deadlock detection & recovery)  
Implementation → barrier avoidance! (deadlock avoidance)  
Implementation → barrier implementation locates in such a way to prevent  
deadlock.

### Minimizing Adhesions

Xprocess deadlock can be avoided by reordering operations in schedule.  
switching order of operations → can affect schedule (in uniprocessor, can never restore)

### Avoiding Adhesions

coordinating the process execution (maximize)

$E_1 \rightarrow E_2$  : if  $E_1$  starts  $E_2$ , and no pre-reqs  $\Rightarrow$  deadlock!  
( $E_1$  needs to locate higher priority)

Resource ordering: maximizes mutual exclusion

Shared DAG graph

New DFS

pre-order ✓  
post-order ✓

- Mnope na zapis DFS ozav. davanjuvuv da vrednost očekujete,  
nau des zapiszavajućim!
- Endju MIA, kai su zapisujući. NOIA? } { Sledjaju po oktakas (nako...)  
proper barao vaspade luge!  
zapisujući barao nesjekunac  
npo luge!  
zapisujući auži nau posne redoslijeda  
čemu da je snage nado davanje  
zapisujući ovaj nu se raspoređuju  
(dešenju važi naveden restant!)

### Monotonijos xpočivih doppajidu (čuvaljuju se loci) (dva pretpostavka blizu radnog)

timestamps.

- Luka Maja prvič oči očekuje → davanje xpočiva doppajida TS(E)
- kai taj naređenje znao eukleon zna!
- ⇒ Ti spomenutim emisijama da n redoslijedu evntiput ujve ono  
radnog npr. očekujem.
- Address abort kai restant ons E ⇒ uči timestamp ja E

ja za Data : xpočivis doppajides } timestamp-W na uči blok/četvrtice  
timestamp - R ws metadata

Luka n xpoč. doppajida on  
uvodljivim nuc database  
redoslijed u bloku.

Napomene:

### E. Detalji Read na Leđdu Data D

1. k  $TS(E_i) < timestamp - W(D)$   $\Rightarrow$  za Data očekujuću  
nau većeg npr. očekujem  
 $\Rightarrow$  Abort(Ei)
2. k  $TS(E_i) \geq timestamp - W(D) \Rightarrow$  za nraspeć npr. nračna očekujem  
evntiput timestamp - R(D) =  $\max \{ timestamp - R(D), TS(E_i) \}$

(B) Ei Jaka write oso D.

1. Av  $TS(E_i) < timestamp\_w(D)$  : probnik va now write saa  
be Data nov sioxu nra address do diktis ceoqis  
suoqloji.

$\Rightarrow$  Rollback (Ei)

2. Av  $TS(E_i) < timestamp\_r(D)$  : nra va address do blok D  
nov dabocevi estavaria do xoypon suoqloji

$\Rightarrow$  Rollback

3. OK, nra write  $\Rightarrow timestamp\_w(D) = TS(E_i)$  update!

+ tbaobis pravilofos

oni locus  $\Rightarrow$  oni deadlocus.

To protocol uvedenfaco 2 qarcev  
ava nra vediavo, exi bahanypo,  
coxkponiffo

-- Elva endenuo, spoczana za deta,  
nacivoces abot do suoqloji  
upo fegadnego overhead.

713D 19-5e

Negalmo: Napamikos basus dedofosos

exw nalla data, de seipw ws, do za exesproso

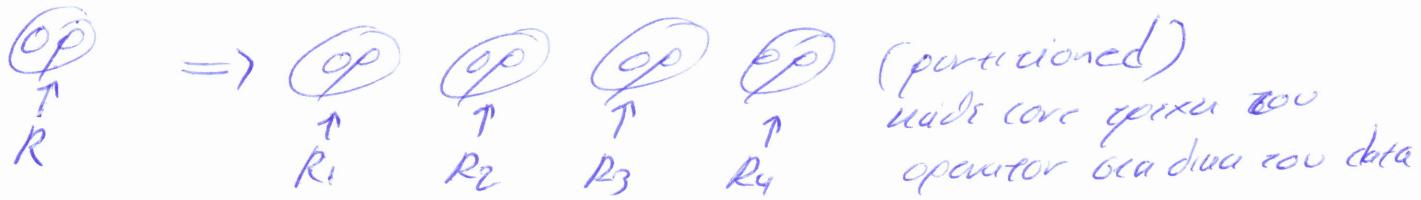
5) Napamikos {or emido exesproso}  
{or emido 1/0

- napamikos exesproso 9
- napamikos nalla ipwesfaewu. 1

Mopges Napamikos

1. Inter query parallelism  
aufaww zo polko xponos zou seiva.
2. Intra query parallelism  
enmaxoun exektonz bostimofos epwifato)

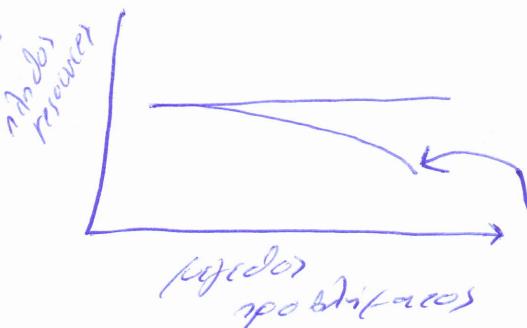
• Napamikos exektonz



• Speed-up

Aufaww nopoos (dilatwifos)  $\Rightarrow$  x2 speed up factor... de bspwren  
exw sublinear speedup. exwts aveypoi naci... [NOTE]!

• Scaling



ausfaww pojdos ipobifatos  
aufaww nopoos (avidaqa)  
an pna bwra do exw linea OX!  
de oifbawt nore  
exw sublinear.

### 13) Efrodia 8000 προσων παραλλαγή

- Ηρεμησης
- Παραβολες : κανικλιδα γεωβασης σε νεανικης γης (σταση)
- Συνη, Αυτομοτοροποιητη λειτουργη, η οπη ενος εγκινησης ειναι CPU

### 14) Αρχιευρωδεις

#### 15) Αρχιευρωδης διαχειρισης μοντελος

μοντελο τυπων για τους επεξεργαστες

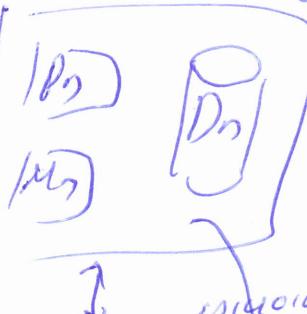
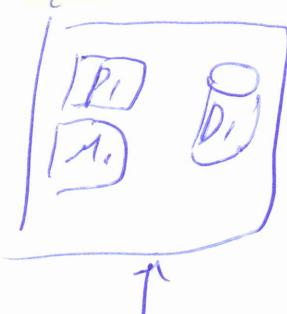
αλληλου στην μοντελο (την secondary storage)

επιμονωντας επεξεργαστη προσων λογισμικου

#### 16) Shared bus, μεταξικη μοντελο (την αρχιευρωδη)

secondary storage μοντελο.

#### 17) Αρχιευρωδη Shared Nothing!



- + Διαφορετικη παρασταση!
- + κάθε κατηγορία λογισμικού λαμβανει χαρακτηριστικα για data center
- + Εγκατασταθηκε λογισμικού
- + κοινωνικη hardware (good PC!)
- κάθε κατηγορία λογισμικού

(P.S.: προσων στην την 4η σημειος)

μοντελο συνη

αντιστοιχια σε λειτουργη clusters  
την την περιγραφη Datacenter

+ κάθε κατηγορία λογισμικού  
- περιγραφη μοντελο συνη  
- διανομη load balancing  
- αρχιευρωδη προσηγοριση συνη  
To uathixi anoi arvo prosp  
va enai slaved memory.

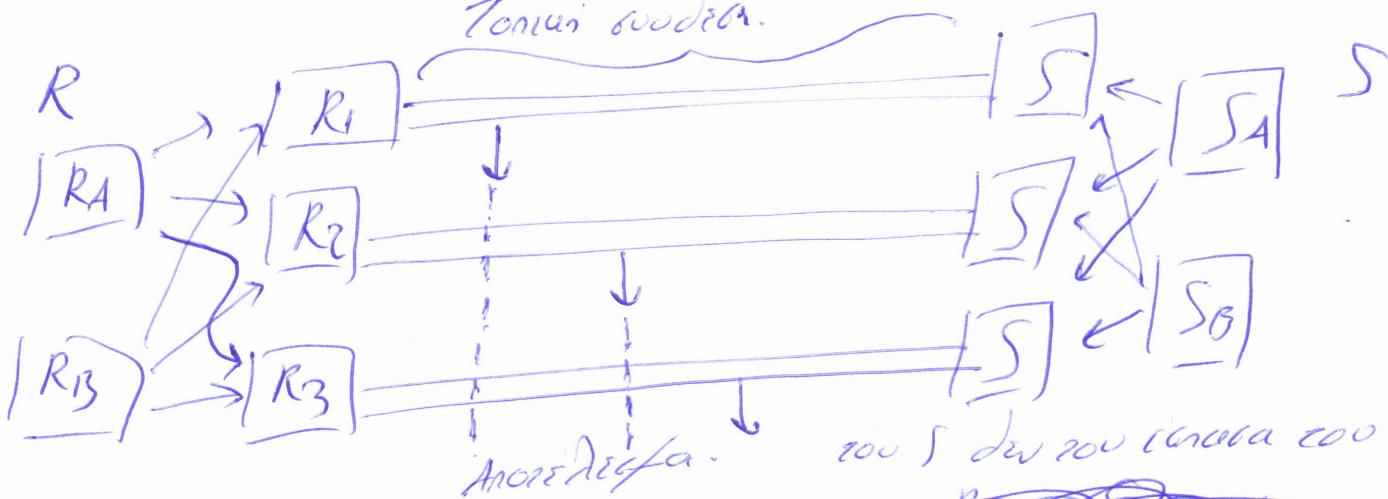
TBD 19-5f. MapReduce Baotos Bedofewu B'

# MapReduce Baotos Baotos

Dedu fan ferroku em  
Enwauuu

(exw slared Netwug) 2 nbaous { R S

Tonui wudca.



MapReduce 62

3 partition

~~MapReduce 62~~  
62th Ethau copy. h 5 fungs

no uude tiffou 200 R da exel odo

200 S => da uaua n fom n10 jppuope

HDFS: kauw broadcast em S! buu n daskia da jissi emma  
spoonodeen propos nivaud S!

→ Ar ethau pyarts o S, da emm brak bi awéou ja va uauw  
parallelism ro join. ne bondaci zo hash function  
ja va uauw partitioning em R hae S  
hash oco uideu. noo dedu va uauw ro join.

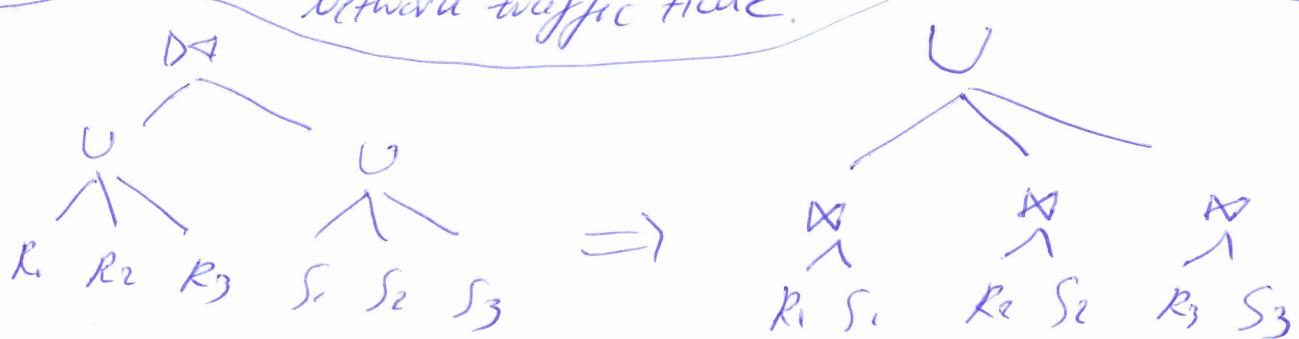
U(4) = ... mod 3. idic hash function em R hae S apa  
za bladda pe idic hash code da nboor fusi (MapReduce)  
optimise

→ buu ro Select mapReduce nre wudda.

Nodes queuing : CPU time  
I/O time

Network traffic time.

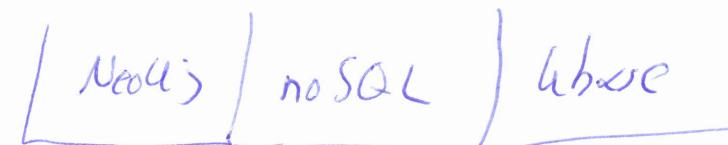
ii)



Execution workload

TBS

26-Sa.



Graafico  
Robust  
Unknown usage.

noSQL : no = not only SQL

3) • Key-Value Store : Hashmap

Eina node scalable  
endo, doar un node  
unor serv pe un singur

co value fixe sau va  
co valori va fi mai mult  
potrivit ca oca  
ora foa ifr

• Wide Column Store : organiza data cu coloane  
under organization of  
numerous columns (ex: powers to  
different departments)

+ ecranare data sau, sau  
aggregations, count  
**Libase** (cassandra)  
Ex: rezultatul DBMS modulare sau record, apoi pe  
pe multe aggregations pentru ca sa facem asta

• Document Store : mongoDB → JSON  
main object = document

+ baza de date NoSQL ca interface  
⇒ mapReduce  
⇒ Map Reduce  
+ fejlörök elvégzés (JSON)  
- deoarece ecranare multi join!

• Graph Store : Neo4j : anodnueu graphis

: Neo4j spexe reteaua  
multe web interface

5) 4 xapta conciliu

Aeriformata  
Anoploidea  
Moniliformata  
**Divergentia**

{ To (consistency) Eina co no xpoobipo  
de la prioritatis edexxo. (locking, xpoob cprazik)

# Brewer's CAP Theorem

Eina univertificata din proprietatea retelelor retezelor

① Consistency : toti ai uvertiblanoa za idia raspuma unde se scrie  
Availability : every request { succeed } promptly in less than a millisecond  
Partition tolerance : rezista la eroare amprenta sau failure

1. Serializable { durability user → data is consistent, even if no updates are issued. } durability user X
2. Sequential
3. Causal
4. Eventual: order of operations update is  
update, order of operations propagate, and all replicas will be consistent.
5. Weak: less strict consistency  
order of operations propagate, or  
operations do not propagate, or  
operations do not propagate out of order

→ A less strict consistency has less partition tolerance X <sup>less co-dependency</sup>

## ④ Not Only SQL.

TBD now the approach uses copies now instead of Relational DBS  
 Now dragging: data model, storage model, failure handling, consistency  
 model... now!

## ⑤ HBase

→ a storage system

→ BigTable: can handle sparse edits, even sparse models or null values  
 → multi-dimensional system. der modellierung  
 (unstructured data, timestamped (Tx or unstructured binary data!))

→ n HBase instances to cooperate

TBD

26 - 5b.

	HBase	RDBMS
Layout	column-wise	row-wise (rowular)
Transactions	single-row	yes
Query language	get/put/scan	SQL
Index	row-key (a region contains row-key, proper sa exec secondary indexing diff for scanner bc performance)	by column
Scalability	Peter Byted	Tara Byted
R/W	Million queries/sec	10 queries/sec
Schema	Schemawless untyped opns for geo-replication approach diff from traditional Structure over RDBMS	well defined

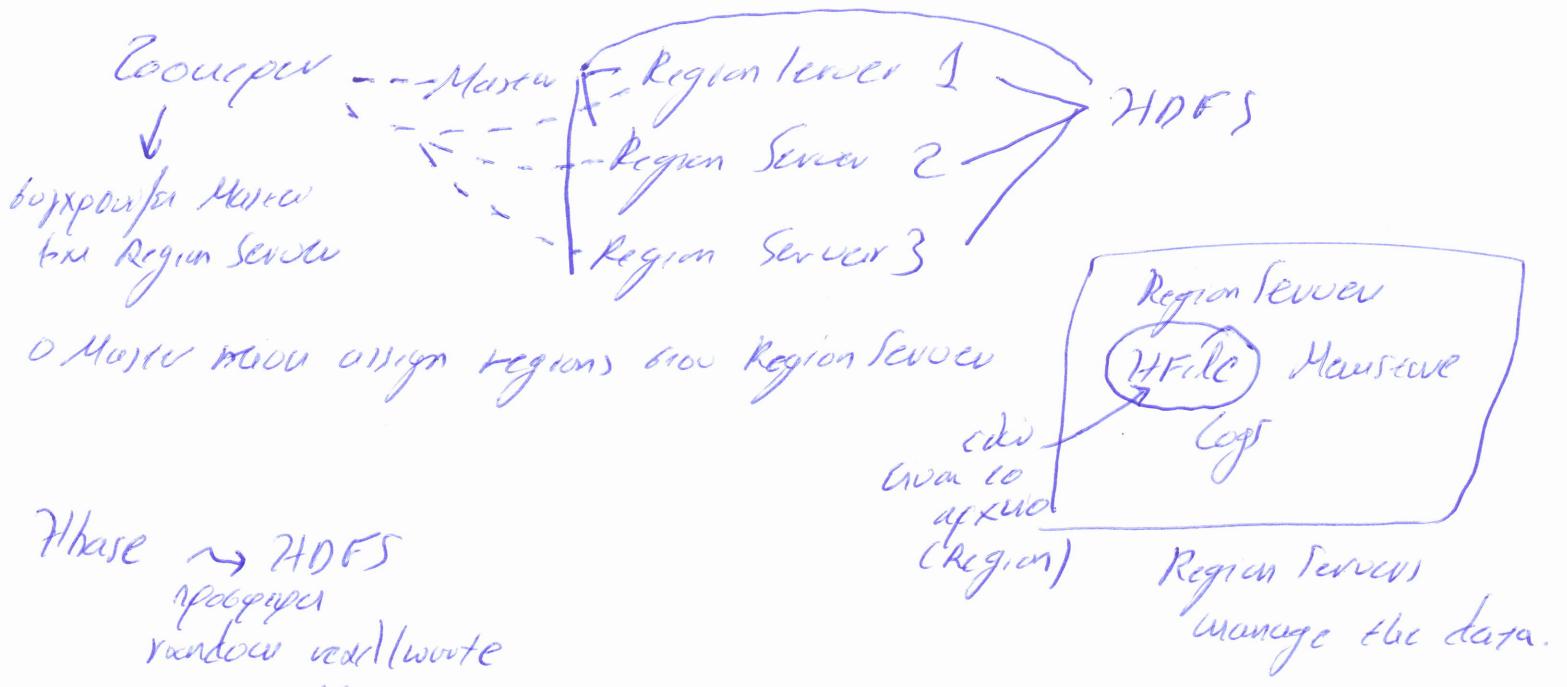
Hbase optimizes Bigtable  
to Bigtable Service on Google  
No crawling on web.  
Diff in: web page, oral news before  
: need be consistent  
Hbase = open source version of this.

- 16) 3 approaches (indexing w.r.t.)  
to design time stamp.
- 17) Physical Layout  
Hbase: minmax, max by regions can coexist  
in server, one region could be OLAP source  
One Master, Slave approximate, LE Replicates  
coordinating TD

### How Hbase:

(Table, Rowkey, Family, Column, timestamp)  $\rightarrow$  Vector

- 18) upon update Sorted Map now sorted first on Rowkey then on Column family  
then on timestamp (2nd, timestamp)  
and then timestamp (2nd, timestamp)
- 19) upon update Sorted Map (column family)  
then on timestamp (2nd, timestamp)
- 20) Map for user-id : rowkey finds entries: column family }  
} update replicated  
Map via (row  
family) via  
entries



Hbase ~ HDFS

replicates

random read/write

nx microservice and execution on opeaq  
skudo to block (random)

microservices { get  
put  
scan }

SEN new join, to user flow  
Java + MapReduce...

### Applications

web crawling

Managing certificates

Time series

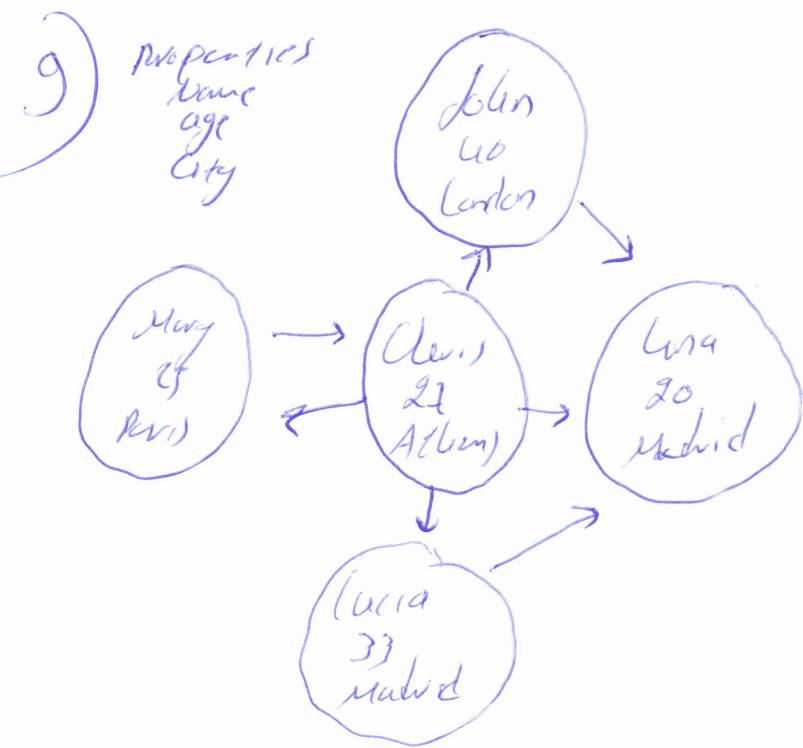
WOW = times

nx predictions and cpa

- 29) Example 30) User profile Mary Name to timestamp private photo  
Anonymouse, rel, age  
31) delete Data watermark..., purges n log in each file.

## ④ Neo4j

- Sintax es Akvoa Cypher, implementación de graph base queries que no son SQL. Existe un dialecto, no es el que se usa para la query de Neo4j. ...
- Data = data grapha, relational grapha, data no es separada no es en forma de tablas y propiedades
- Heterogeneous Network: las propiedades varían entre los nodos  
Objetos tienen → ides o nombres únicos con id, así es que el nombre es el id.
- Idios omnis nomen est, dagepnus est nomen, nomen est nomen.



10) La query se ejecuta  
en modo secuencial.

→ Creating Nodes

→ Creating Follow Relationships  
o antes create da igual  
nomenclatura, cada uno define su propia forma.

11) Loxodonta March  
con gráficu us antes.

March (a:Person), (b:Person)

Where a.name = 'Mary' AND  
b.name = 'Chris'

Create (a)-[r:Follows]->(b)

Return r;

"visual" language...

14) Simple queries

Match (n:Name: "Mary")

Propri de los nodos son flexibles.  
no no redundancia constraints

by default cada nodo tiene una unique id  
propri de cada attribute es id.

- 16) Paths
- ↳ öjdes aus  
graeferenzen, oder eg  
relationships
- $E^{\star J}$
- Match  $p = (a) - E:FOllows \rightarrow (b)$   
where a.name = 'Mary' and b.name = 'Chris'
- ↳ Neo4j and semantic data
- 17) Order By
- 18) Moonuna pinus 2. Moonuna os Rel. DBMS da ordetu unianos  
unno join. dro Neo4j eku nolo nio anto  
nau der fe endagsper n uaradunos Der Befehl besteht  $p = (x) \rightarrow (b) \rightarrow (c)$
- 21) Updates. 22) Delete Node, ~~pe~~ detach delete, Delete relationships  
(obwohl spezies auf)
- 24) Adding & Removing Properties.
- 26) Count, topkos (n) km (c) over free variable, oriodinare wofkos
- 28) Erzeugtes Biwo, Apotheor Node drap. unno, apoteor drap relationship  
Create (:Car ...);      relationships : nodes os  
serial number ciba draygerekko
- 32) Shortest Path.  $E:FOllows \star J$  oriodinare ligos.
- 33) bpes coos qidaoz unu qidaoz ens Mary noo dees unu qidaoz ens
- 
- Anwendung osu anwag: 5, 6, 7, 8  
 $\Rightarrow$  nolo eufrognis o unduwas  
 ens cypher  
 (zo ordet by der eina  
 ergebnis zo.)
- No SQL = bigData  
 after 60000!

→ Spesifikasi: imparsit dan mendukung operasi bukaan

Nugadifis: Sia awis di bantuan perifis dan node sokarum, dan perifis dia diextendable (extensible), dia dicuci, kira jaraknya spesifikasi dia diawis "zade", dia naro blok dia naro dia zo yepis (1, 2, 3, 4)

①

• Aksiorevoum DBMS: dia sepis zo node component 71 buah..

• Node zo concurrency control

• Node zo query optimizer

} perintah data exchange  
analog pada zo components  
tm dia properti ke u  
abxakiru zo node buffer

• Storage Management: records  
 Babuan duxypten data

cedidos  
 frane)

noda blok napaire atau Mass Manag  
 Import/Export  
 Read/Write Bloks

Records (nudipis pnuu)

(probabilitas pnuus (dapat))

• External Merge Sort  
 + Nugadifis.

② Dendriques Dopes Apobenfias

B, B+ Trees (ox B\*)

Sia dia zo xaiw } 1. Insert dia nro

dia ola za element),

2. (Move dia exw

ola za croixara

dadeifa)

ubow (mpvoces) pe ordening

tm xaiw bottom-up + blok...)

Split 1 → 2 min util 50%

2 → 3 min

3 → 4 min

B\* napaifidn Brree

awa dia min Split 1 Node

6E 2, napaifew dia

sepis 2 dia vo uhan

in 2 63

Kaduken duxypten xaiw

Xaiw opis di xaiw dia

da uhan lpb t

2 → 3 nro diuindha

allo xaiw di xaiw xaiw  
 ~50% (B\* = 70%)

~~Διαδίκτυο οι δόπες~~ <sup>(2nd step)</sup> ~~secondary memory ή αρχικά μέσα~~  
~~per πλήρη σύγχρονη~~

### ③ Extendible hashing

(όχι γραμμικός λειτουργός)

Απλοποιεί τη φύση των θυλών των προσεταιρέων (μονοθύλη)  
(δίκια για την πόση η θύλη να φέρει το block του πάταξη)

Στην υπολογητική επιστήμη η φύση των θυλών των προσεταιρέων  
είναι η απόσταση της θύλης από την άκρη της θυλών.

Στο Hashing αριθμώντων της ιαπωνίας η θύλη είναι σε σύντομο  
την επιφάνεια μεταξύ  $78, 23 \leq x \leq 80$   
της κάθως.

### Extendible hashing xap 160:

Έκπληκτη για την ανάπτυξη της θύλης ή (αυτοεπέκταση)  
την διαδικασία να αυτο-extend την θύλη

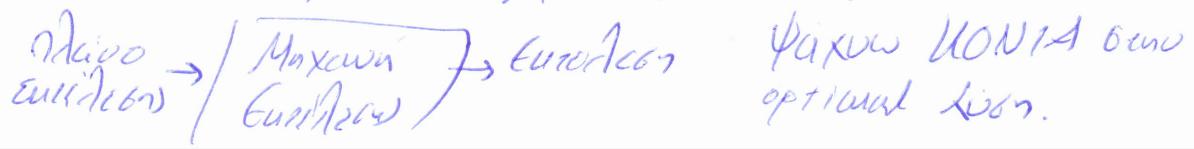
global update hit (drag) = 2, εμμέσω της local extend  
την ίδια 2

### • Βελτιστοποίηση της επεξεργασίας επενδύσεων

Τρόπος επενδύσεων για να λαμβάνουν διαφορετικούς αποδοτικούς

Αποδοτικοί μ/α, ή αυτο-επενδύσεις: output: εγκαίρως διαθέσιμο

το ανατρέπει ανενεγγύει την ποσητική αποδοτικότητα αλλά  
την πάντα πάντα! (χρησιμοποιείται, πολλοί από τους επενδύσεις  
γίνονται).



TBD 2-6 b

⑤ R-tree

ou efecte nu nu se grada.

⑥ KNN, skyline

⑦ Cluster computing - Cadoop

grado de disponibilitate  
ML, knowledge algorithms  
eficiență și performanță  
Data Management.

⑧ Spark

⑨ ... ⑩ hands-on course

⑪ Databases, locking, schema redefinition based

⑫ NoSQL basics Elastic  
shard handling  
store memory

Definiție

Na rafinarei se apără 512 4|96 47|139 2|8

56 2|99 4 3|2 1

un record se se divide bloc proprie va fi redus la 4  
apăratori sau se o buffer extre 3 directe frames. De unde  
se obține se apără maxim?

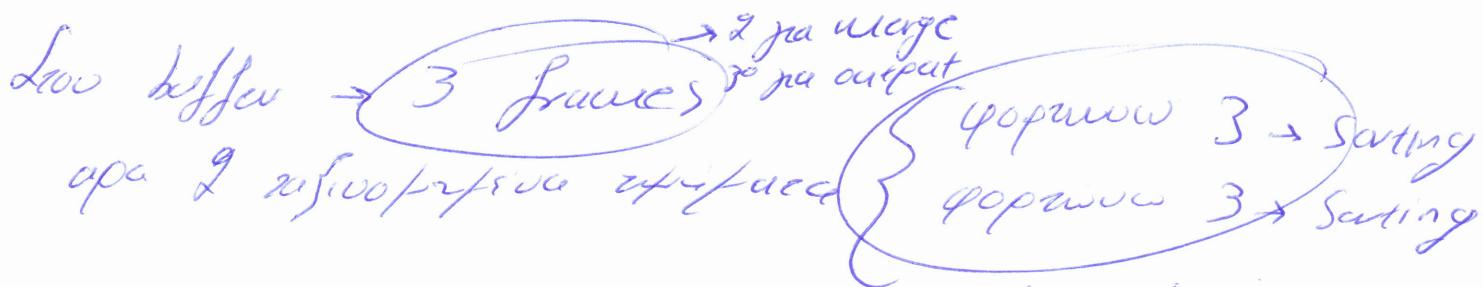
pentru 3/0, se apără na un rafinare?

W6-7

lucruri exite 22 apăratori (records)

de tact bloc se apără 4

apă se apără un bloc cu cel  $\frac{22}{4} = 6$



και σε αναχώρια τα δεγχωτές  $\xrightarrow{2 \text{ sorted segments}}$

πε 1 ιμποστα opa 10 operations =  $2 * 2 * 6 = 24$   
 (2 reads from read/write)

6 block  $\rightarrow 3 \rightarrow \text{Sorting} \left\{ \begin{array}{l} 2 \text{ reads to expand} \\ \text{Combine...} \end{array} \right.$

Opfa 2

Ανάδειξis R(a,b,c) επωφελή Select R.a, S.d From R,S  
 S(d,e,f) where  $R.c = S.f$  and  $b > 10$

$\rightarrow$  2 διαφορετικά πλάνα εργασίας, διαδικτυος ενδιάμεσης

Τηρηση λαμβάνοντας προσδιορισμονας B+tree σεν R.b και Ηάδην συνδέουντας οντος προσετίσματα

Λύση

Ηύπαρχη υπόσχιση για join

Μετα σηματόδογμα  $b > 10$

Ηύπαρχη σηματόδογμα  $b > 10$   
 σε κάποιον index

Μετα υπόσχιση για join

η  $b > 10$  καρτών index

Μετα υπόσχιση για join

ανάλυση για την πλήθη των records  
 στην οποία πάντας βρίσκεται το 10<sup>th</sup>  
 ηδυση είναι διαδικτυος

Υπόδειξη για επαρχιακούς πλάνους  $b > 10$

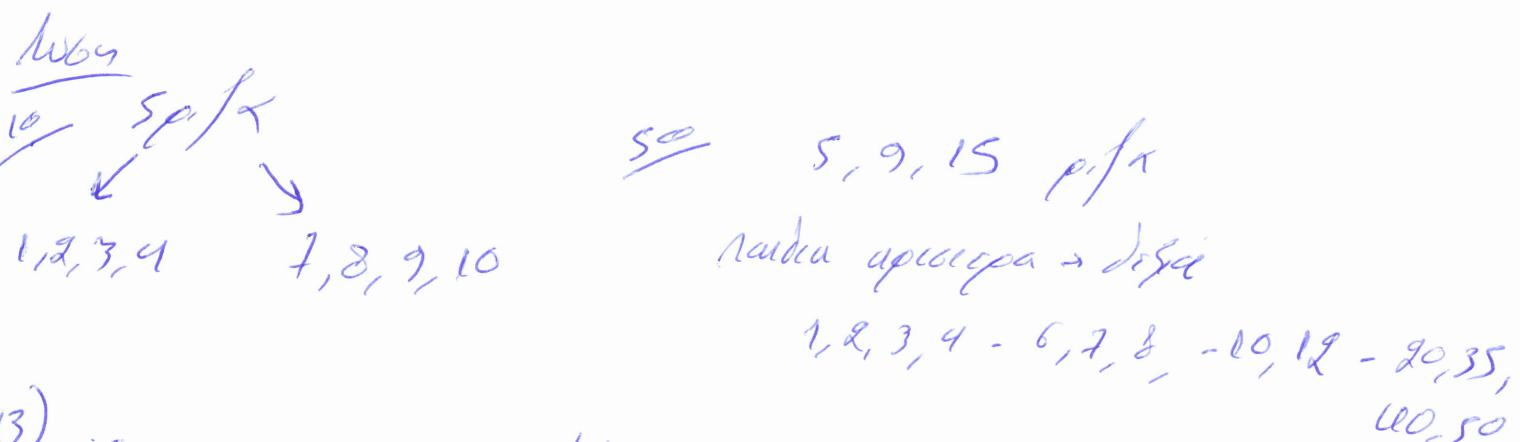
TBD 2-6c

### Opção 3

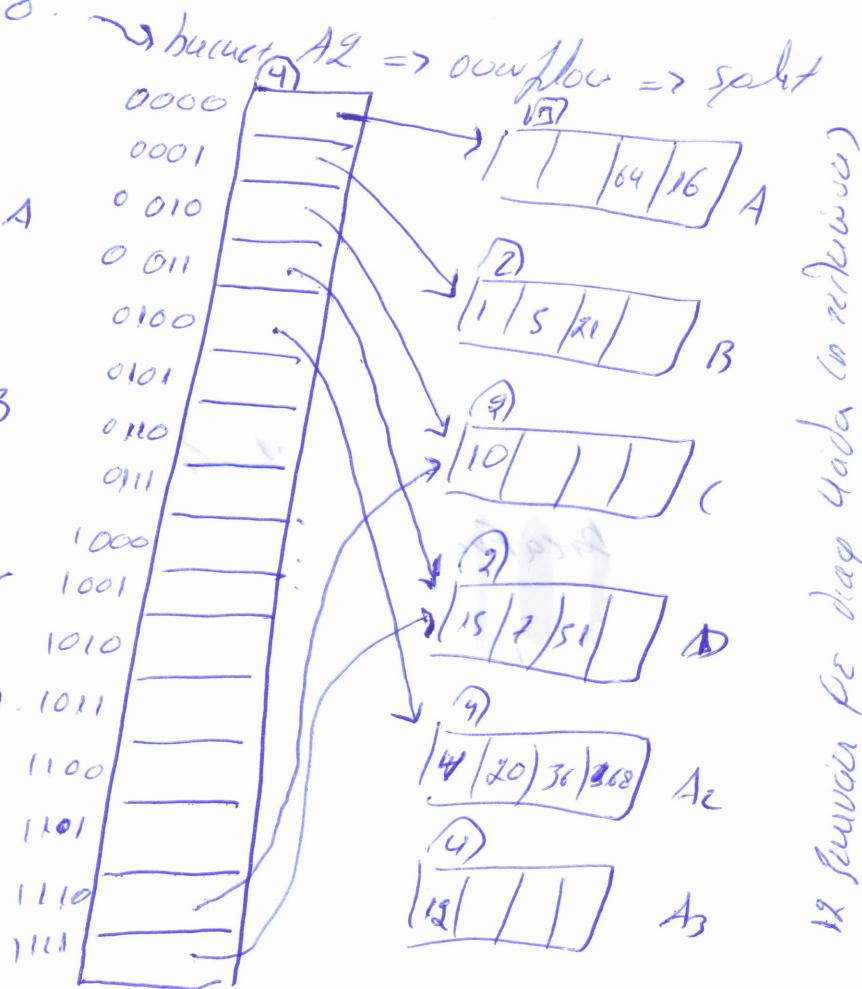
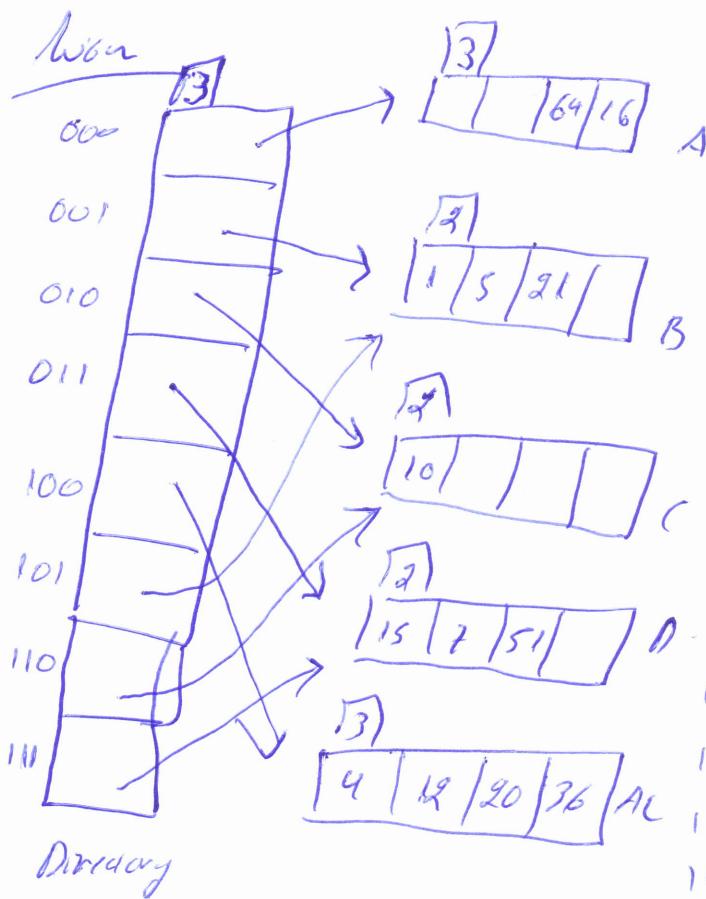
A) Entradas de entrada: 5 1 9 2 8 7 4 3 10 18 15 20 6  
46 35 50

EE adao B+ tree: Node com 4 entradas de cada Node

Node na folha tem que ser o insert(10) em insert(50)



(3) Com o níveis extendible hashing temos entradas fixas, Na  
busca em folhas em dígitos para um determinado binário  
de entrada para hash code 68.



H2 Sóvicia P. Diaz Vida (comentários)

## Oef 4

A) R will not return output on demand.

Inappropriate expressions or terms not appropriate for communication  
or type of the audience.

## Mist

Implementation of range search can change

Implementation of range search can change  
the time taken for an approximate  
range search. BFS and DFS can be used  
to find the range of data

No implementation → ~~BFS~~, O(1) DFS

B) Two approaches to solve problem is to ignore boundary values  
and consider them as separate entities, or to use range search  
to search for the ~~approximate~~ approximate value.

## Mist

1) lost update 2) incorrect summation.

## Oef 5

a) Newer version of Apache Spark is a framework of open  
RDD, DataFrames, narrow/wide ML

## Mist

many programming languages ( Scala, Java, Python, R )

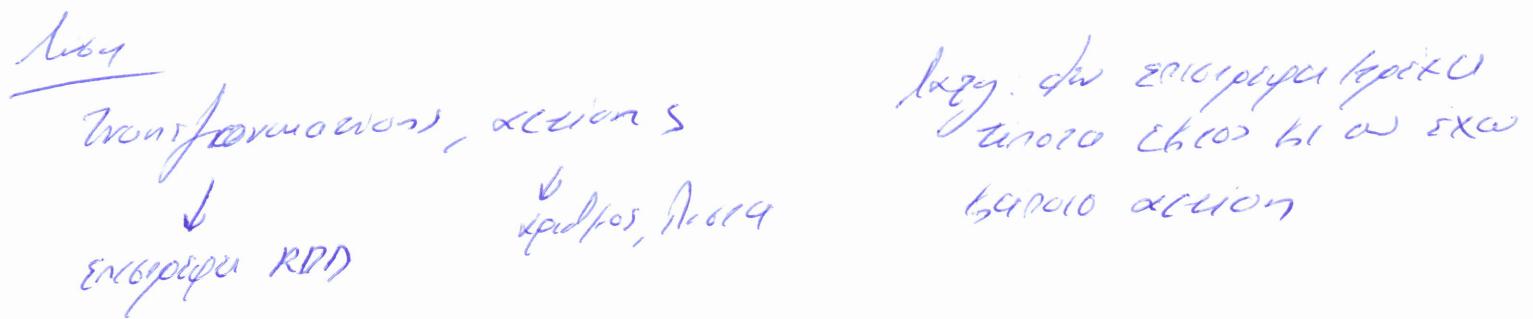
Architecture for big data processing

Distributed system

7BD 2-6d

Υποεμφύτευτη RDD ή αν DataFrames  
νόμιμη Διεύρυνσης σε εβδομάδες που φανερώνεται

B) Σε πότες η λειτουργίας χωρίζεται από τη λειτουργία που  
προσέρχεται σε RDD, πού να δινεται χαλιά εως?  
Η εφέντα σε το spark μεταξύ;



18-2 x. 7.5.

## Delete 20\*

- spoefaw ux napis no uelbo-1 noo exoo'
- idio parent
- dala 6 110 jax co Delete.

## Delete 20\*

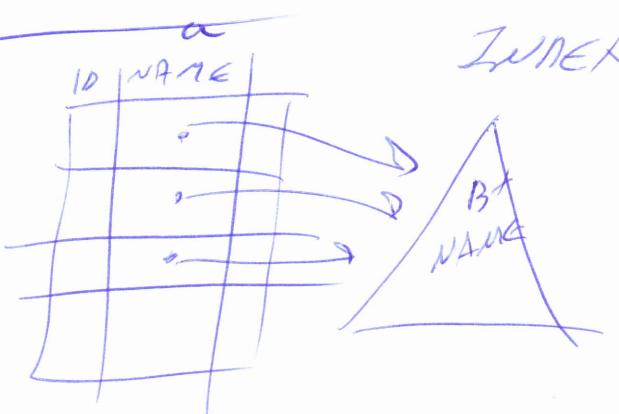
- Id properties ux organization (EZ25 EZ4 293)
- (ben addafer ux index, dalaan zo 24)

! On wapen an waan  
geforn bax (80%  
ux underflow).

Gawa underflow zo EZ1 303

Gawa merge kaa ux no napis → peninsulas  
zo ifd ux daspar  
(ux daa jadaan uxu)

- An ux tree uput 3-4, propis ux dasda alta insert  
taa ellere sognis ux ux xamabe ux addafer zo  
kijeda ux tree.



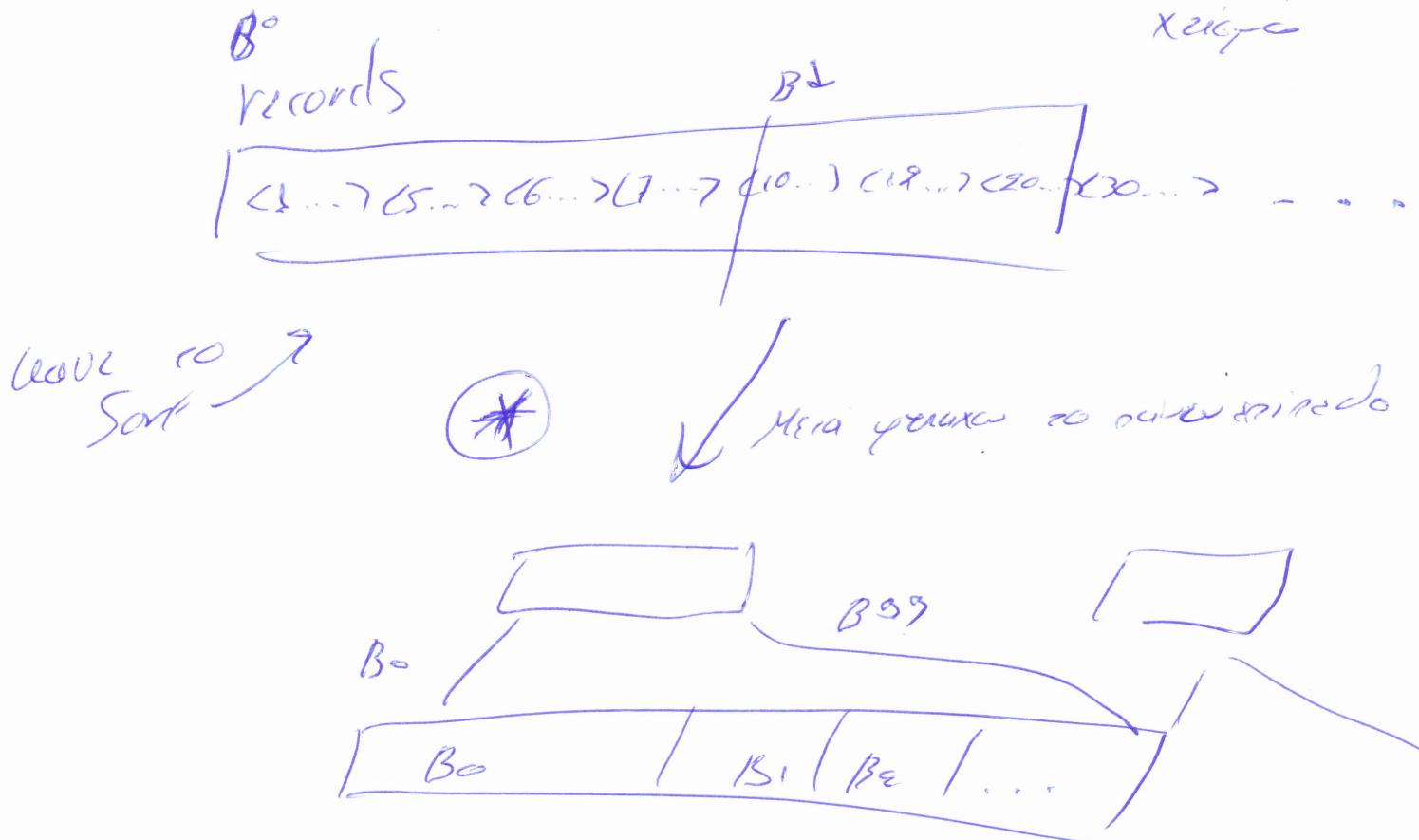
Jax ux xam index napis  
ux records 1-3  
freecord → insert (record 1)

Da co unga ja 1.000.000 naunds?  
Mogen ux eo napis no yepupas?

- Bulk Loading } Freecum zo celueno enkido,- } kui peta napis nolt ux napis

Bottom-up

Xειρού



Εναρκτήσω πόσης ουσία να γίνεται σε αυτό τον προγράμμα για διαχείριση.

→ Η Ι-Ι διαδικασία που είναι top-down χειρός δε σημαίνει ότι, όταν τα records είναι στην γέννηση, διαλέγεται το Bottom-up (για Sort και Merge extend Merge Sort).

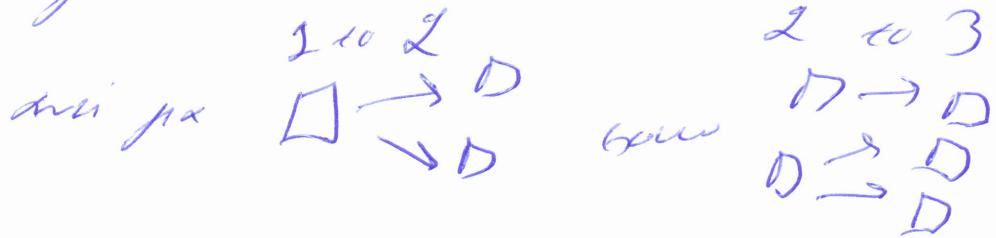
Ⓐ Η ουσία των γραμμών στη Storage Application  
κάτια σε 100%

Όπως η γραμμή είναι 80% και να μπορείται να αλλάξει  
στοιχεία από την γέννηση μέχεν delete / insert  
→ αναγράψει τα φαίλτα.  $B_{tree} < B^+$

→ Η διαδικασία σταθερίσει την είναι  $B$  ή  $B^+$  ή την ουσία της στοιχείων  
→  $B$  ή  $B^+$  στοιχείων που διατίθεται αναμερώντας (είναι διαδικασία που δεν έχει πάθη  
η Bottom-up για την παραγωγή)

18-26. 7Δ

B\*-tree : zo xronoforoi pa pygalo storage utilization  
xronoforoi den eni leftori karen overflow, an nawa lau  
genitori jenawas den



spisus 6003 to 4 ixw  
overflow 6602 po m'lo 2003  
Jadian no adu xpono adu  
etwa Leyaknya Storage Utilization, ta dianekin o Algoritmus  
Ebagajaris.

jadi den den pygalo storage  
% storage utilization  
m'lo 1402

~~Merupakan~~ → kew tu gtu vanu dianayu kan han burooi kawen Insert  
tu respona → Tu juu merge kan . . .  
→ pa ario xronoforoi ahel sys's am bebasana onus oo  
B\*

Nivales Katalogstrukturen

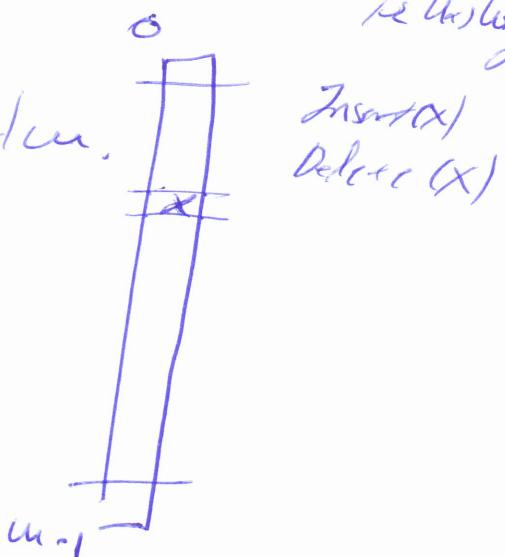
## Hesling

$$h(x) = ((2x + b) \bmod m) \bmod n.$$

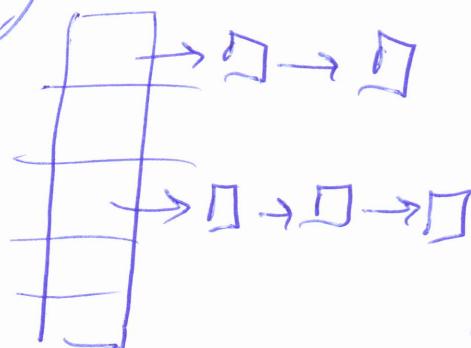
"χαρακτής"  $\rightsquigarrow$   $x$   
convex  
or specific

Min asymptote

O(s)  $\Rightarrow$  constant  
relationship



clusing



$x \neq g$

$$h(x) = h(g)$$

Der Index  $\Rightarrow$  regelbar aufwärts  
jeweils die Substrukturen zu ändern



$\rightarrow$  die Größe von gewo. Blöcken sind fixe  
bei Secondary memory  $h(x)$  wird nicht  
angetroffen

$\leftrightarrow$  Agar kann es PB da sein nicht Insert bzw.  
Delete, so  $\times$  da speziell die Spezifikation  
der Struktur für die Substrukturen zu ändern

$\rightarrow$  Erweiterbarer Katalogstrukturen (durch Verlängerung)  
(extendable Catalog)

& Graphische Katalogstrukturen

18-2c

Hastings or Secondary Manz.

## Eneuratos (excepcionales)

Eulerian for direct o



$x = 10$

$$U(10) = 0, 1, 0, 0, \dots, 0$$

$$U(z_0) = 0$$

$$U(13) = 11100 \dots 1$$

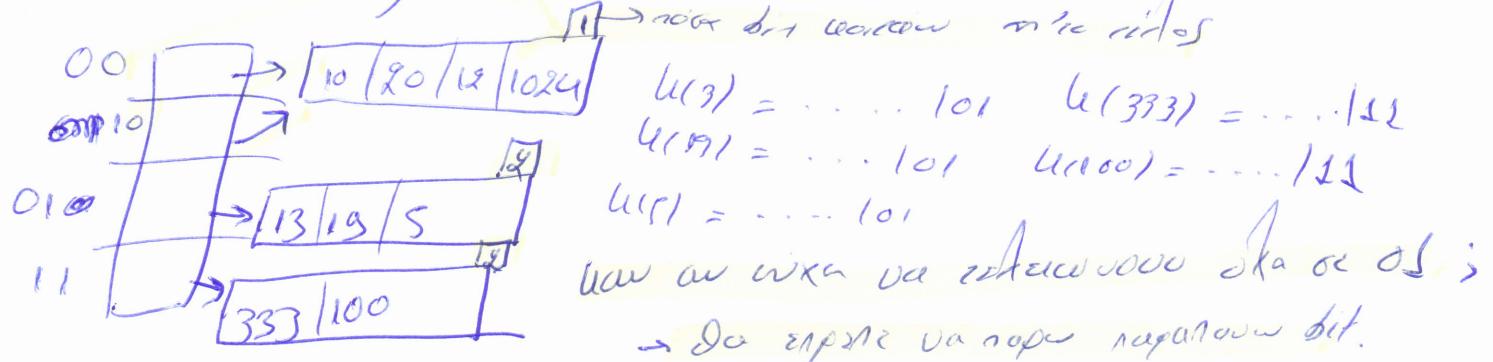
6(19)

As 'n offer sou ek gevra naas, daar baie meer van

Now Insert(100) per  $b(100) = 0011\ldots$

Tuan;

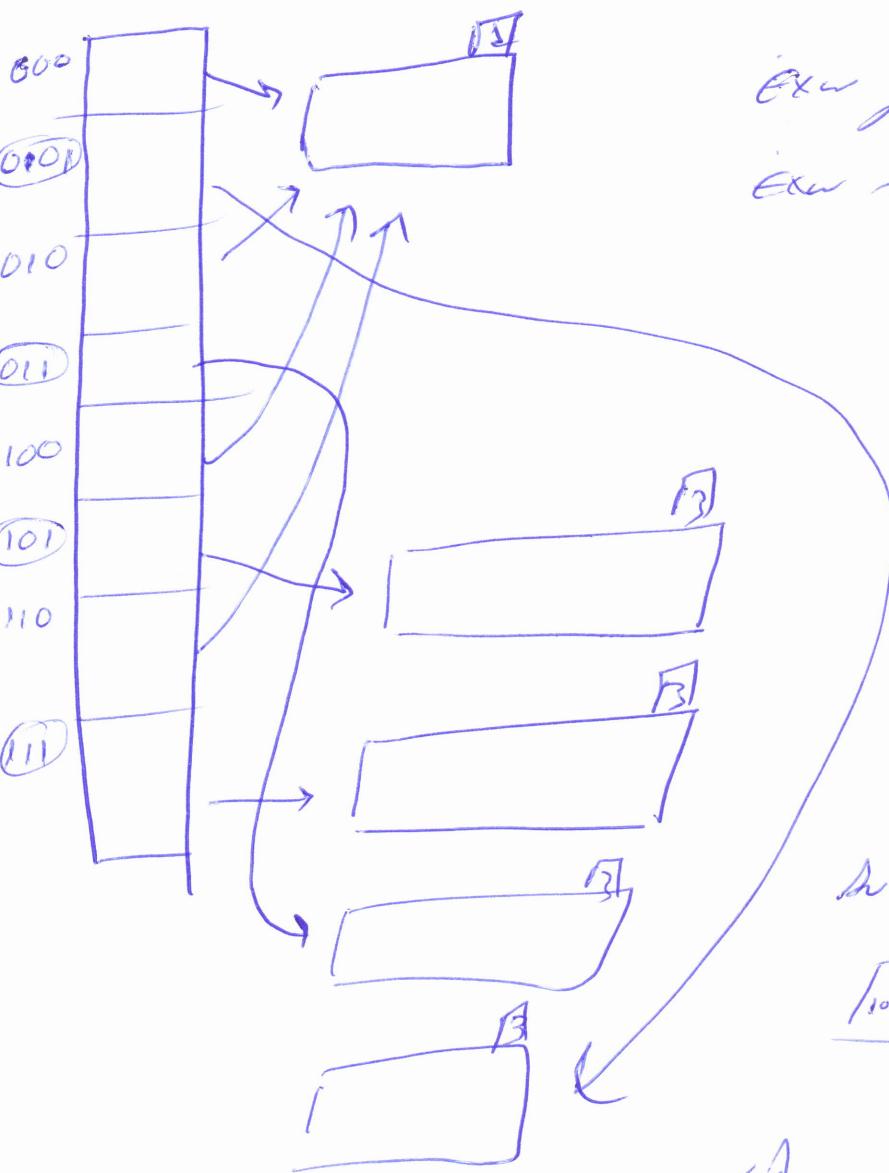
low on *Dendrophylax* see notes (xref to the relevant bit)



↳ Nu är vika va resten voo ska se os;  
→ Ja synste va noge napatow sit.

3

disk access  
overflows



new global # bits = 3

new address local # bits

To executable code,  
you extend the  
memory too, can write  
back without overflow

In order overflow is

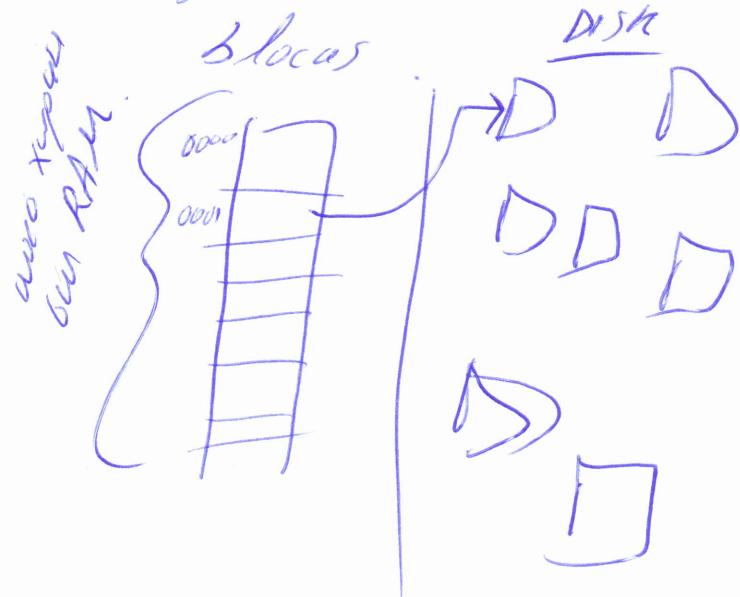
$\begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline 10 & 12 & 20 & 1024 & \\ \hline \end{array}$  after insert (6)  
operations do  
new disk access

data register co  $\begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline 10 & 12 & 20 & 1024 & \\ \hline \end{array}$  error

→ good access to memory, even if no end of block

more bytes of memory in the partition than the address!

overflow of the memory will occur, memory goes to



block XES

new memory 4-bit

loc = ..... 0001

you have no available block

1-block access! O(1)

bad address

new disk access  
block Y

18-2 d.

As o index de xepoer om RAR pra dictar se  
no back ou bse, da exw 2-blau access.

(as no blau sive se haffer de exw o-blau access!)

### - Hash Based Indexes -

→ Mapa am px de dis conteudo bit de rapsia da L pista.

Dele unxioaria oco hestling, px amr 5000  
unxionadur han oxi xupam!

No rapsia am no hestling nro de enxio pista  
de de elxa unxioaria da exw addost qd. Gous amr  
unxioaria 10 bits, da exw 0, da se dicta enxio  
nroam 20 x amr Custo function.

Se fomow se redimido 10 bits, nro de unxioaria ca exw, da :

XX+XX XXXXX } nro de exw idia  
JJJJJJJJJJ }  $(\frac{1}{2})^{10} = \frac{1}{2^{10}}$  opa de hestling  
aproximada de exw colisions



TBD 25/9/22

Key 11) Βαθυεπονηματική στατιστική.

Άριθμος M/E = λογ γραντ των επιλογών  
το γενέριο.

Βαθυεπονηματικό: το οποίο δεν είναι ένα απλό βαθύ πάρα  
είναι κρανοθόρακας πάρα,  
σημειώνεται στην αναδόμηση

1. Λογ σελεκτικούς γραντ των επιλογών index  
\* δεν είναι στατιστική γραντ.

3 ACBEC

ii. Αναφορικής λύσης  
μηδενικούς τιμών πάρα;

6. Μηδενικής σελεκτικότητας  
(μέτρη) ή μηδενικής πάρας  
που πάρει την έναστρη

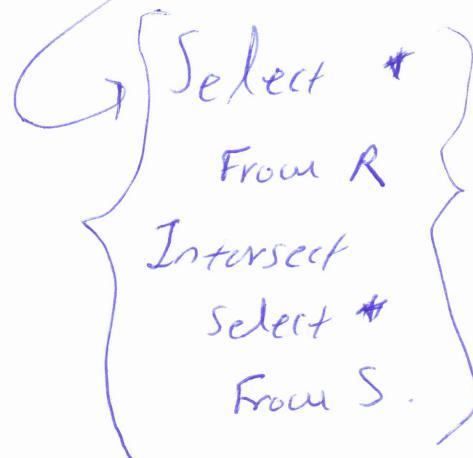
δεκα (60(X))  
τετραπλή  $\Delta_{x,y,z}(60(X))$  (I)  
 $60(\Delta_{x,y,z}(X))$  (II)  
δε γραντ με τα κρανοθόρακα πάρα  
ας είναι α, b, c  
(4x n kαι με 3)

Αν έχει ΑΝΒΛΔ η μηδενικής σελεκτικότητας  
per records  $10^6$   $10^6$   $10^6$  & λαρυγγικής σελεκτικότητας της Ε, η μηδενικής  
πάρα πάρει records της ΑΝΒ  $\rightarrow 10^6$  records σεν  
κερδίζεις.  
Ενώ ΑΝΔ  $\approx 2$  η μηδενικής σελεκτικότητας.

probabilistic

2 niveau)

$$\frac{R}{\alpha \ b \ c} \quad R \cap S$$



$$\frac{S}{\alpha \ b \ c}$$

( $|S| >> |R|$ )  
Métodos populares  
Síntesis de datos  
con R

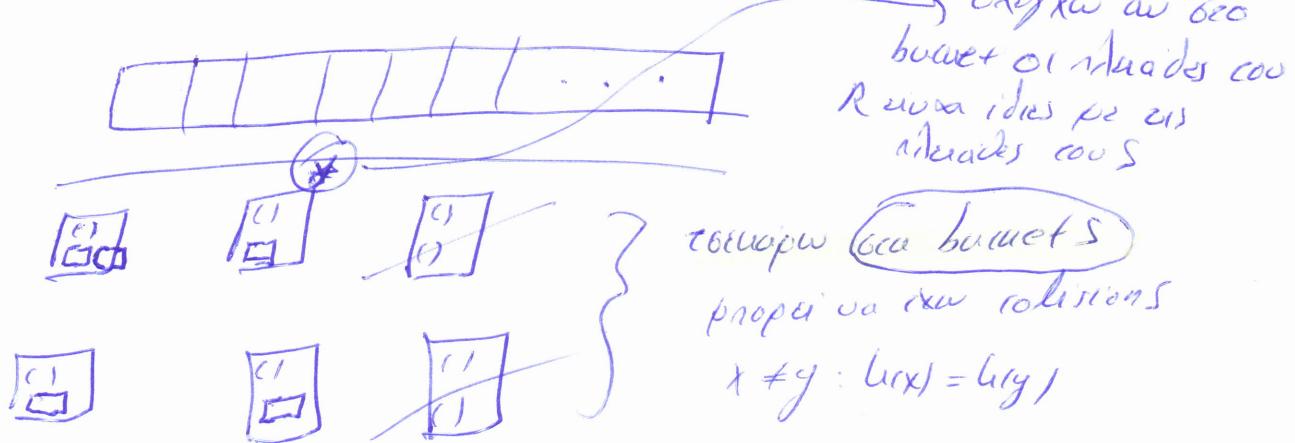
Baseline Algorithms  
na unidade de tempo de R  
bem como tempo de S  
 $\propto n^2$  e  $O(n^2)$

- Depois da block  
não:  $O(n^2)$  da vez  
dá-la da vez não  
aparece

2) Quaisquer hash  $h(x, b, c) \rightarrow$  hash key ou é bompre  
ou seja, hash index

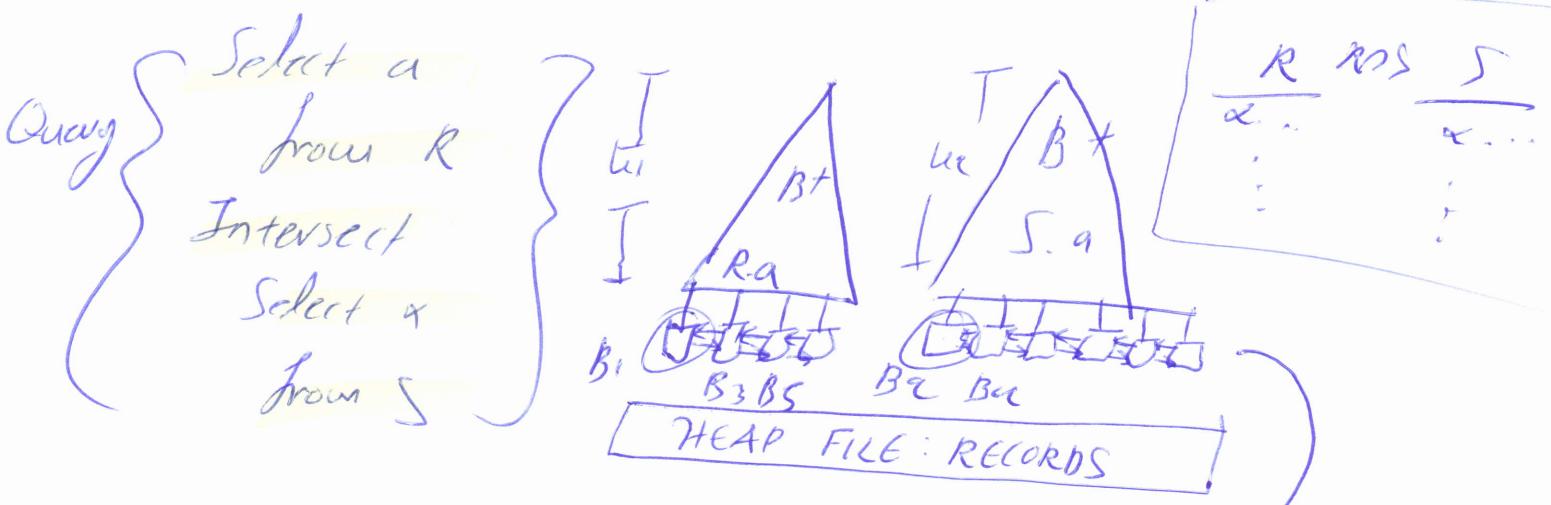
Qualquer hash Index (S)  $\rightarrow$  acessos: bloco de S  
+ records de R  
probe  $H(S)$   $\rightarrow$   $|R|$  I/O

3) Qualquer global hash index tem pra conosco



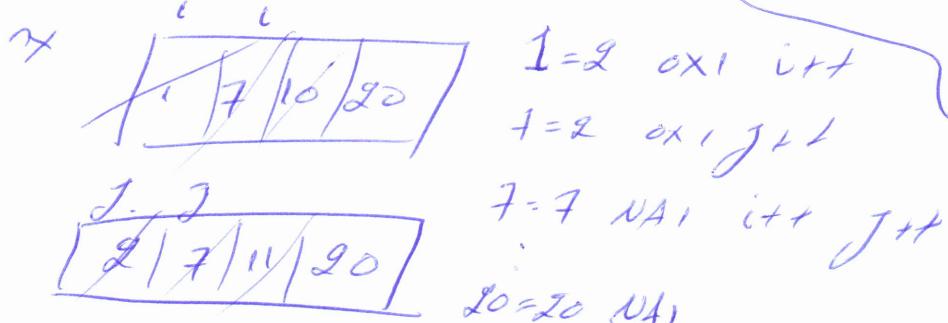
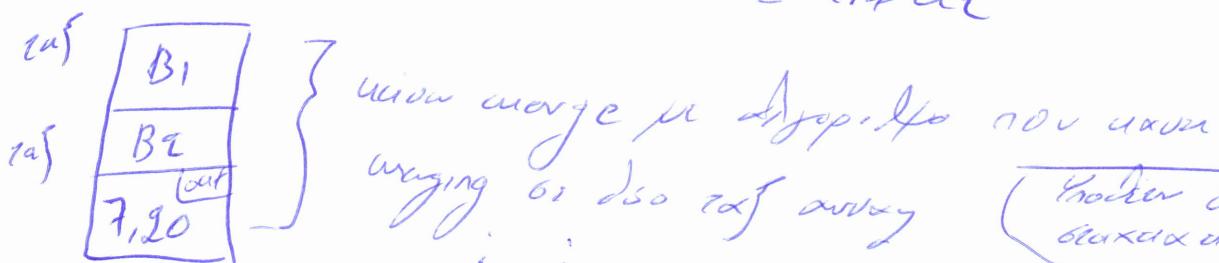
TBD 25/2.

Probabilis exkov pera pia oaths em & (komo be oom/pe  
vora/ra ro Interred)

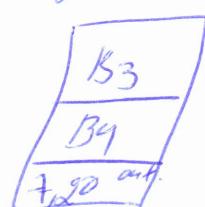


Ende avan rehosphatik  
ipx Ende eten sekoji doo rehosphatik  
muanur.

- Bantuan ro rehosphatik gudo zon kade lec spes
  - 4.1.10 + 4.2.1.10 = 4.1.12



→ Merging of B3 and B4.



av ehanhdu uanolo m x ro B3 qopanow ro B5  
kun buntifur en diudnabia

kuudnabia uanolo → qopita B + R.a kum qopita B + R.S.a  
kun di xpanapondan uadonan ro Heap File!

Ενιαία ή επιφύλαξ

Select a

From R

Order by  $\alpha$

$R(a, b, c, d)$   
→ exec index plan a.

av ή exw  $B^+$  to order by  $\alpha$  ή exw γέφυρα.  
→ exw παραγόμενη

av exx order by  $b$  → διν το exw παράχθη σε λαζαρέ  
apo dia γέφυρα με παραγόμενη.  
δικια να υπάρχει παραγόμενη.

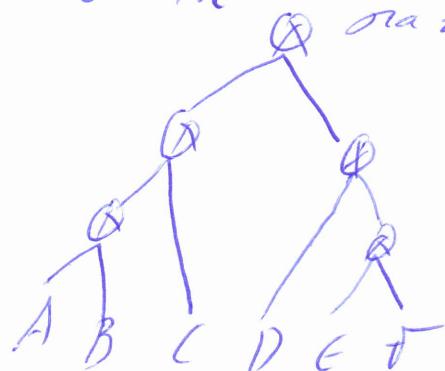
Πρόβλημα βελτιστοποίησης.

Exw

$A \times B \times C \times D \times E \times F$ . nvalues σ.Α.  
nκαι μηχανική ή exw wklz exn

πα του καθηλωτικού

$A \times B = K_{AB}$  τα δύο  
nκαι μηχανική ή exn μηχανικής  
πα τα του σπειρ.

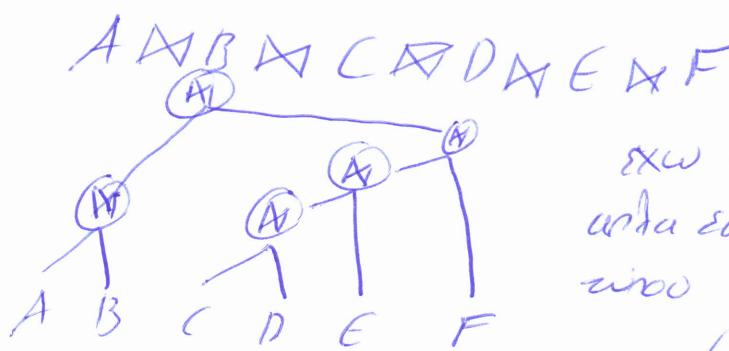


| δια προπονια σε κάθιση  
| τω διποτι! του μετανώ  
| δια προπονια σε κάθιση  
| δια προπονια σε κάθιση  
| του αναρριχήσεις του  
| καθηλωτικού  
- δια προπονια σε κάθιση  
(τα δικά δικά σε κάθιση)

→ δια προπονια σε κάθιση τα καθηλωτικά των ατόμων σε δια προπονια σε κάθιση της συνάντησης των ατόμων σε κάθιση.

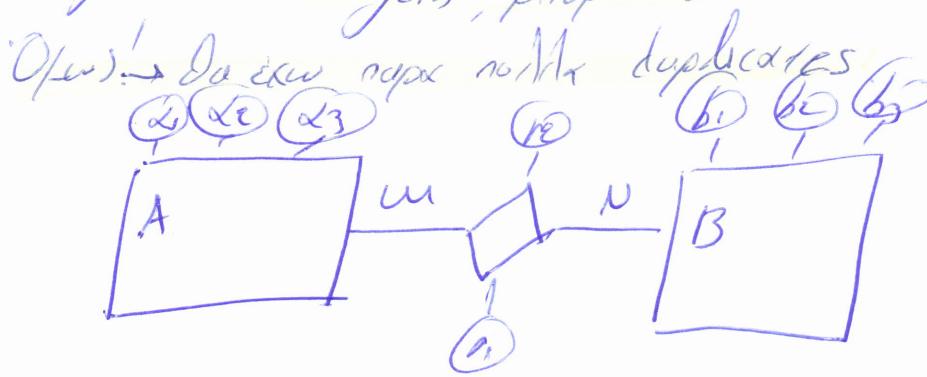
TBD

25/2c. Lais bius bespreken over



exw synoptika bethuizeninon  
anta eis exw rapaferous  
moo exw B+;  
propri va uaw Gesti;

ta ophakienon ra join, propri va bant dha os sre nivau.



ra bant dha os sre  
nivau

\*

\*

\*

avva niva  
ida

Iew de xradan join, dha fodeva noltje synoptika  
xopo (rajoin, 100 xopo)

an deden va uaw update noltje va synoptika  
niva niva noltje, onte zo go na exw  
gwo idio nivau Iew elan ikon.

Eriscaun bethuizeninon

abo ouprapa posta = abo nivau posta, na va  
kayu antes!

kanous 3 → nivau nivau  
nivau nivau nivau

Select \*  
From R,S

→ dat noltje where  
⇒ noltje elan noltje

Klausur 4

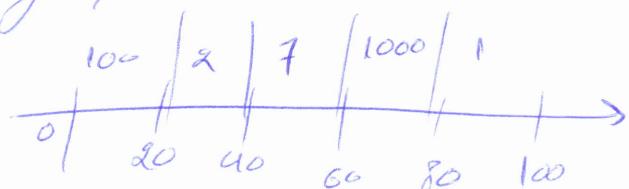
Erweiterung von JOIN nach den zu präzisierenden Maßnahmen.

Klausur 5

zu Subquery und WHERE & GROUP BY

## Erweiterungen

Summary views

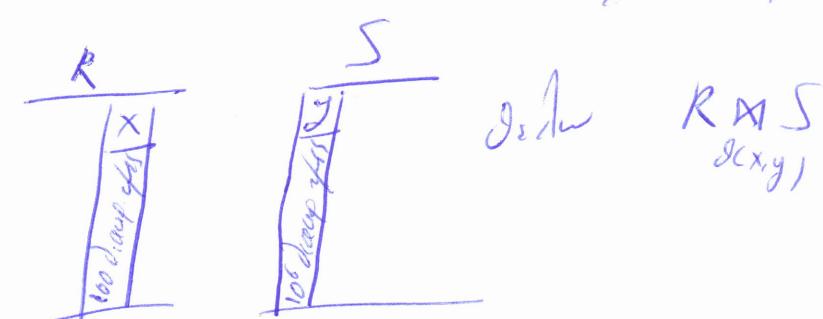


oder für nicht katholische mit christlichen.

an exm

R	b	c	d	Boolean
0				spaltenweise laufen $\beta$ + der Boolean.
0				ergibt eine oder exkl. Fkt. $\oplus$ etc.
1				0 indexiert auf die Zeilen per rechten Diag. etc.
1				$d(1, \alpha)$
:				die Jm. Linien für endgültiges R zu einer der Zeile.
				etc. fürs Gendans.

To mind. zwei Hauptviews müssen



10000 records (R), (S)  
 $|R| = 10^{20}$  m  $|S| = 10^{20}$   
 optimaler Durchlauf

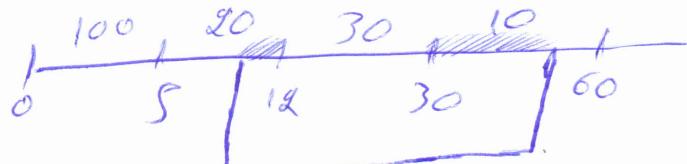
Dies führt zu mind. ein Hauptview mit proper da exm untersetzen erfordert zwei natfns JOIN.

TBD 25/2d.

~~Algebra~~ Με χριστιανής λεπτομέρειας:

Λαραδόφτες  $a \in [10, 50]$   $\rightarrow$  τοιχος στην ανασκόπηση

Έκπτωση 1000 γράμμα



είναι με ενδιαγράφων οι γένει  
του παραπάνω σχεδίου.

1) Η έκπτωση κατά 30

μηδενικής σταθερότητας.

μηδενικής σταθερότητας, η μηδενικής σταθερότητας είναι πλέον

βαρύτερη με αναδοτούντα είναι λεπτομέρειες στοιχείων.

περίπτωση 10 και 12 η έκπτωση στα 5%

και περίπτωση 30 και 50 στα 10%

η έκπτωση στα 5%

$$\boxed{30} / 5 = 6.$$

είναι μεταξύ των δύο περιπτώσεων

Piece-Wise Λεγόμενη

↪ ανάδοτον διάτοκη σταθερότητα, αλλά  
ανάδοτον στη σύγχρονη λεπτομέρεια.



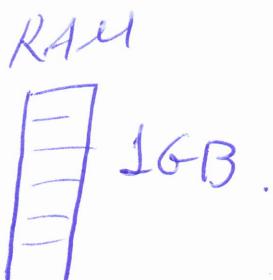
External Merge Sort [Sorting data when all of the data does not fit in your RAM.]

(Merge-Sort)  $\rightarrow O(n \log n) \rightarrow O(n \lg n)$

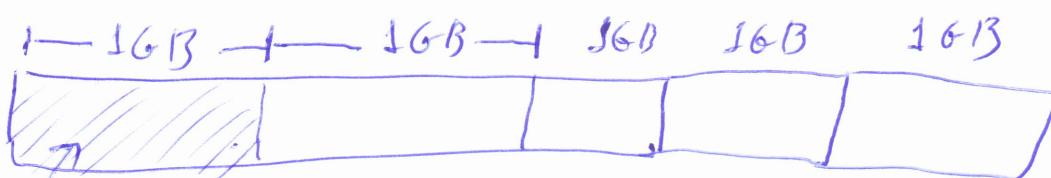
Inception Sort  $\rightarrow O(n^2)$

Quick Sort  $\rightarrow O(n \lg n)$

I have 5GB of Data that I want to Sort.  
 ↳ Files in Disk (2nd storage)

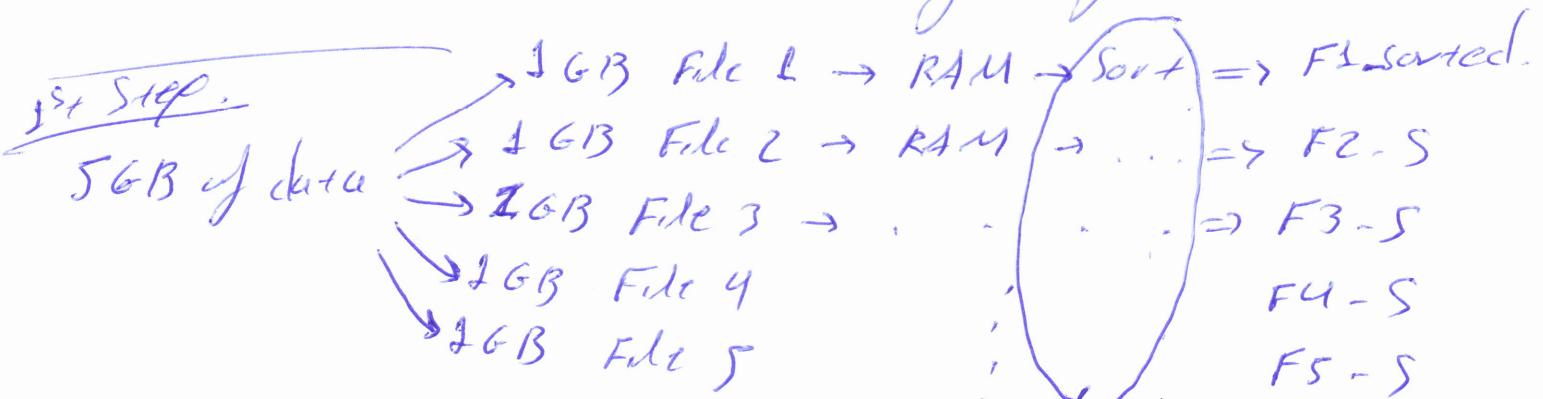


CPU RAM Disk: I can't load all of the data to main memory.

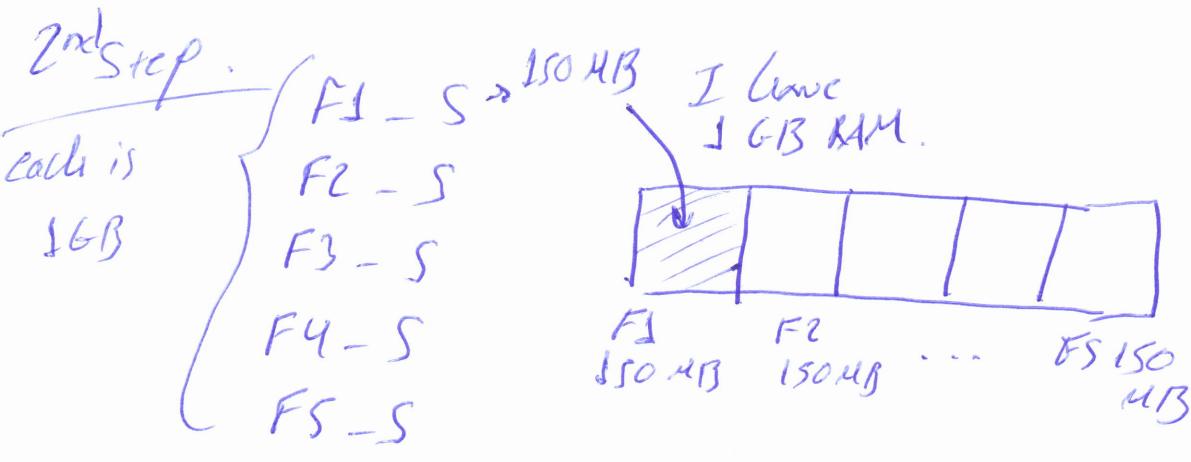


I can load this and sort it \* to my RAM.

$\rightarrow$  I can't store more than 1GB data  $\rightarrow$  Moving data from Disk to RAM is really costly in time.

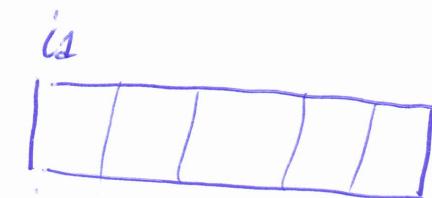


↑ can use any sorted algorithm (e.g. quicksort)

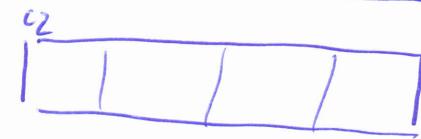


$$150 \times 5 = 750 \text{ MB}$$

and 250 MB free in RAM.



150 MB of  $F1 - S$

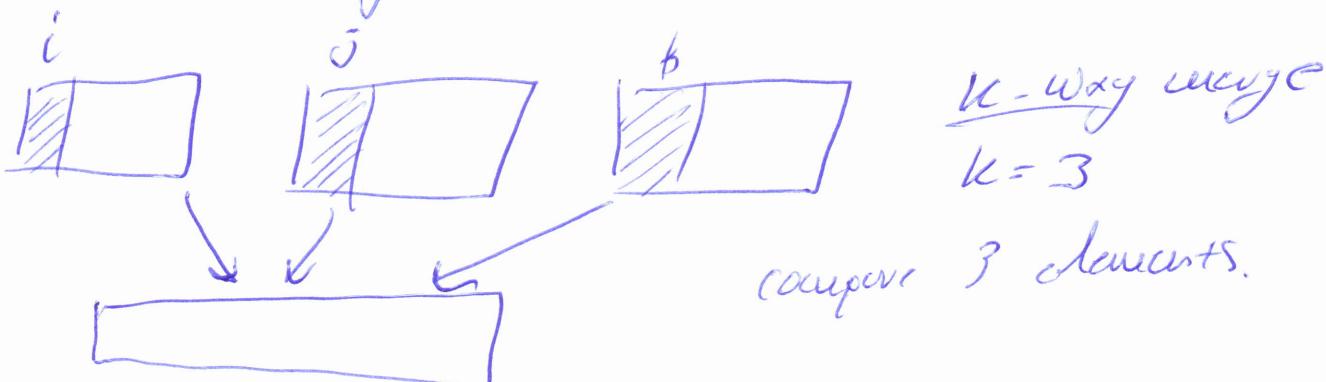
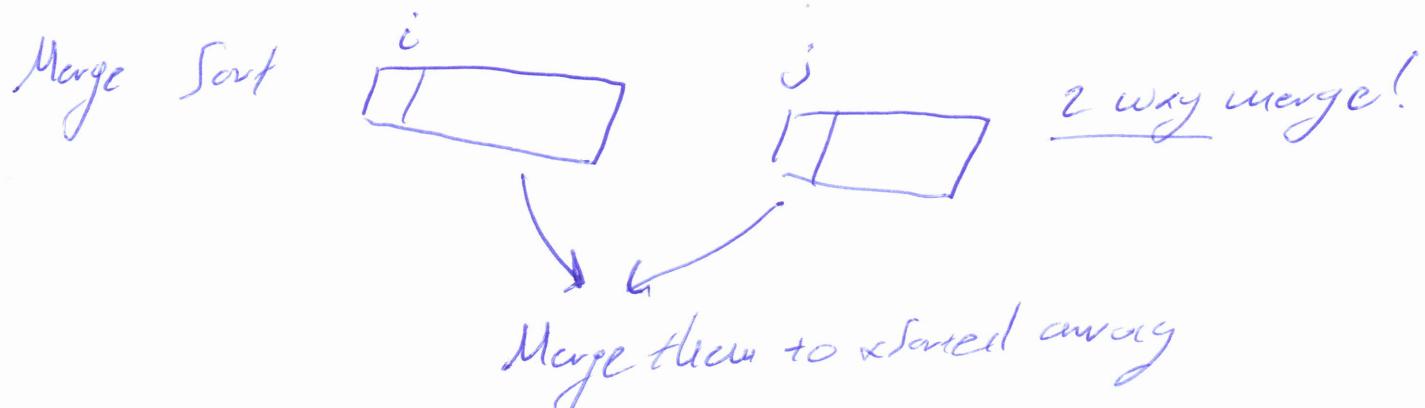


150 MB of  $F2 - S$



150 MB of  $F5 - S$

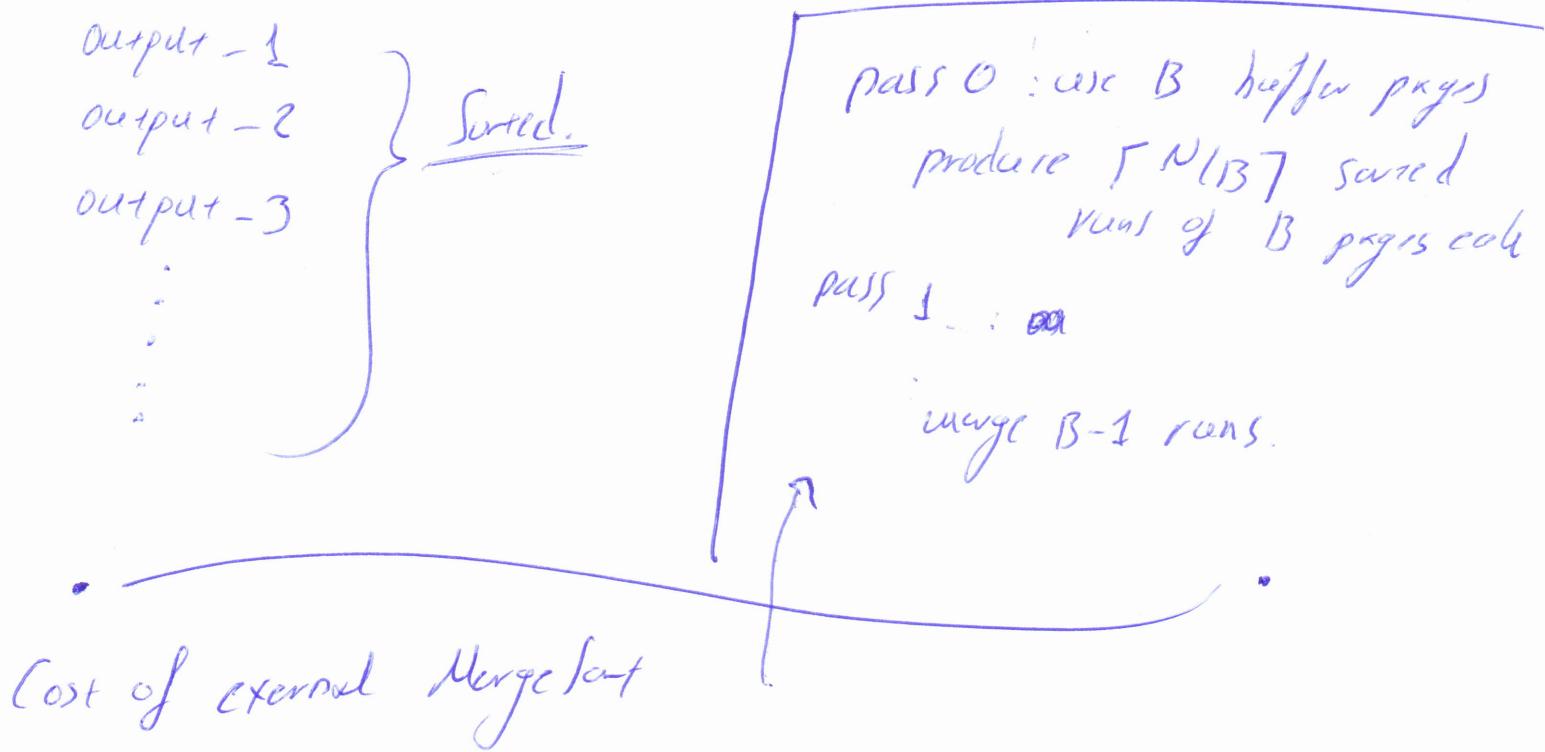
} 5-way merge  
from  $i_1, i_2, i_3, \dots, i_5$  and  
placement  
250 free MB!



→ My 250 MB will fully filled  $\Rightarrow$  we write it to disk. free 250 MB  
 File: Output 1 .

at some point for all of the 5 runs will be passed!  
 to union empty (a 150 MB) for union leading to  
 empty 150 MB too avoid overflow file.

→ Do exec in Output<sub>1</sub>, Output<sub>2</sub> ... in order da exec TA = NOMH -  
 MENA!  
 250 MB each.



Number of passes:  $1 + \lceil \log_{B-1} \lceil N/B \rceil \rceil$   
 Cost =  $2N * (\# \text{ of passes})$

Example

5 buffer pages, to sort 108 page file.  
 pass 0:  $\lceil 108/5 \rceil = 22$  sorted runs of 5 pages each  
 last run = 3 pages

$$\begin{aligned} n &= 1 + \lceil \log_5 \lceil \frac{108}{5} \rceil \rceil \\ &= 1 + \lceil \log_5 22 \rceil \\ &= 1 + \lceil 2.22 \rceil = 3 + 1 \end{aligned}$$

pass 1:  $\lceil 22/4 \rceil = 6$  sorted runs of 20 pages each  
 last run = 8 pages

pass 2 =  $\lceil 6/4 \rceil = 2$  sorted runs, 80 pages and 28 pages

pass 3 = sorted file of 108 pages

pass 0 : ceil(108/5) = 22 , 5 pages per run

pass 1 : ceil(22/4) = 6 , 20 pages per run

pass 2 : ceil(6/4) = 2 , 80 pages per run

pass 3 : ceil(2/4) = 1, done , 1 run for 108 pages.

---

I/Os of External Mergesort =  $2(M+N)$

3 buffer pages  
6 page file

1. Split int  $2 \times 3$  page file  
and sort it in memory.

$$2^*(3+3) = 24 \text{ I/O}$$

2. Merge each pair of sorted clusters  
with external Mergesort

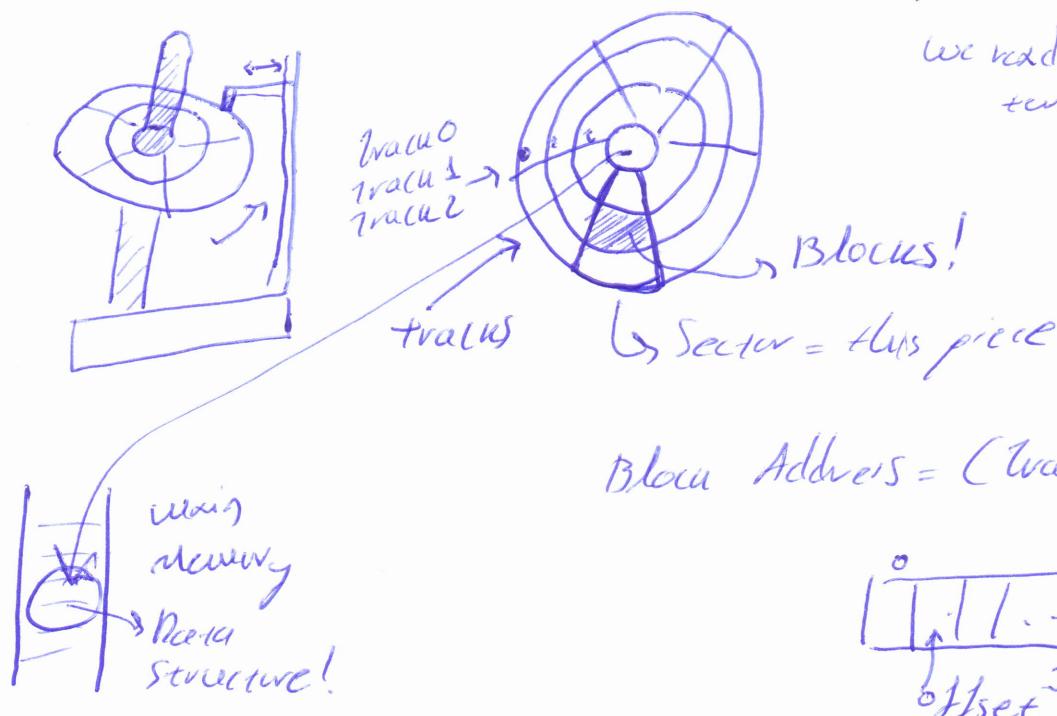
$$2^*(3+3) = 24 \text{ I/O}$$

3. Total I/O = 24!

Never before!

# B-Trees & B<sup>+</sup> Trees

## > Disk Structure

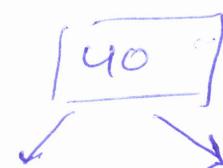


## B-Trees

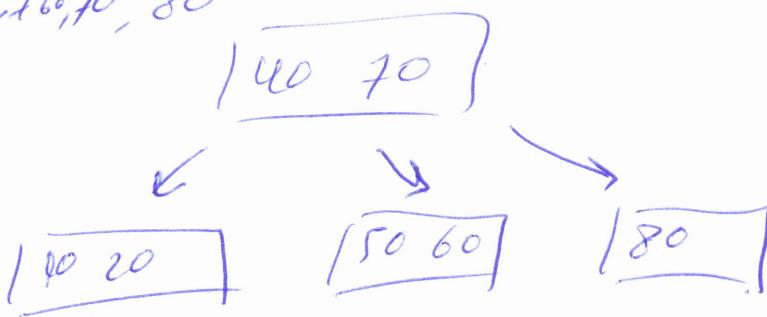
1.  $\lceil m/2 \rceil$  children
2. root has minimum 2 children!
3. All leaves are at the same level.

Keys 10, 20, 40, 50

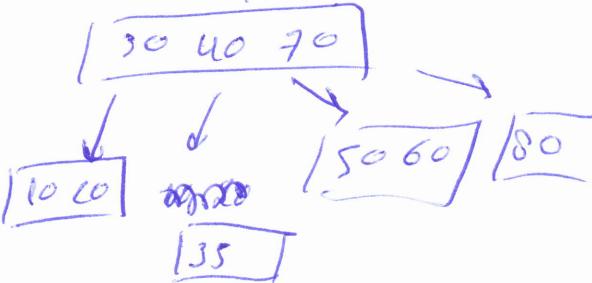
$m=4$

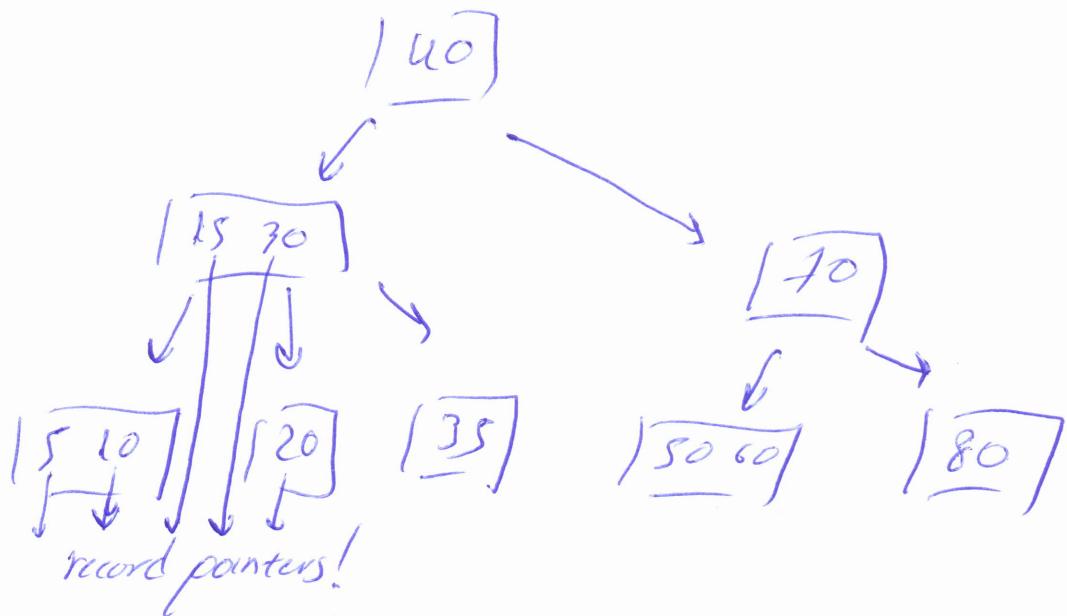


Insert 60, 70, 80



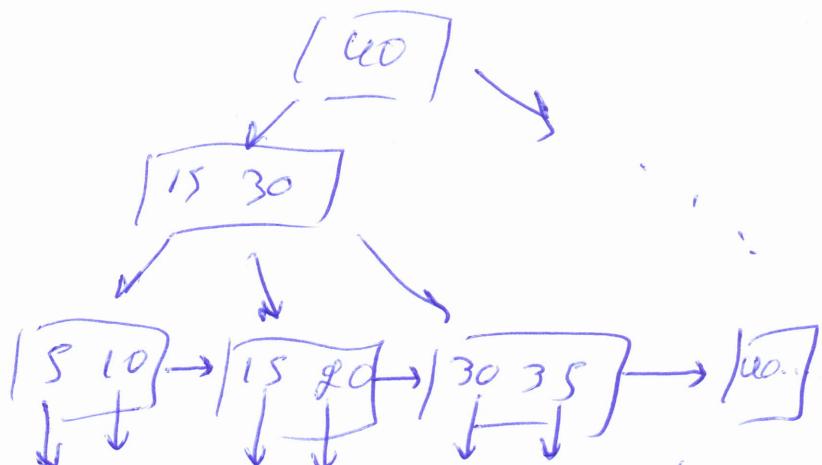
Insert 30, 35





In  $B^+$  tree we don't have record pointers from every node, but only from leave nodes

~~go to  $B^+$  tree now and copy it!~~



leave nodes are connected as linked list!

