DASlab

Home | Syllabus | Schedule | Project | Self Evaluation | Class Notes | Forum & Announcements

HARVARD
School of Engineering
and Applied Sciences

CS165
Data Systems

Have fun learning to design and build modern data systems

class 20

updates 2.0

prof. Stratos Idreos

HTTP://DASLAB.SEAS.HARVARD.EDU/CLASSES/CS165/

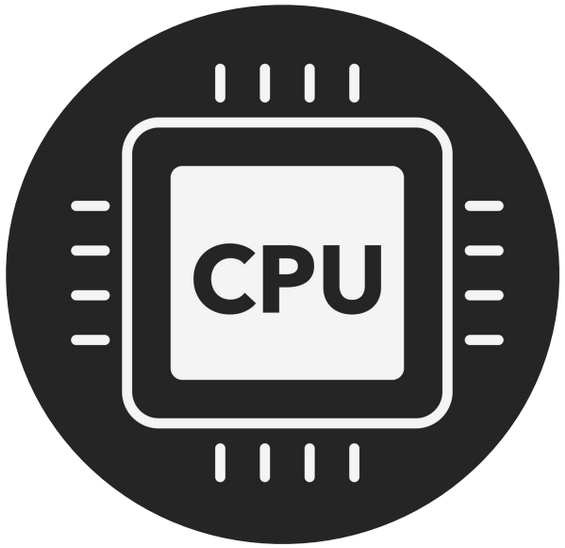
UPDATE table_name
SET column1=value1,column2=value2,...
WHERE some_column=some_value

INSERT INTO table_name
VALUES (value1,value2,value3,...)

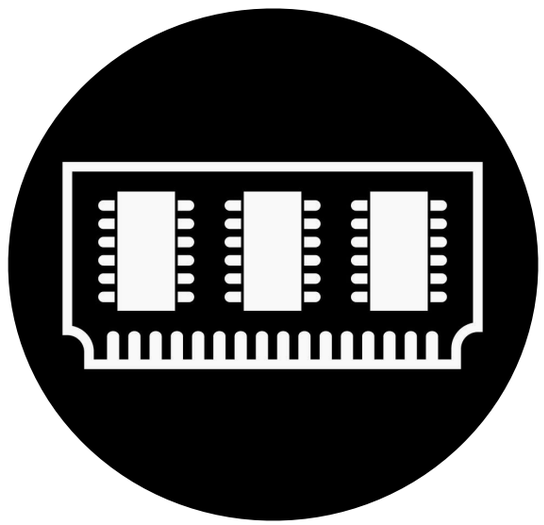
updates

the world has changed
a little bit by now…

updates



monitor CPU utilization

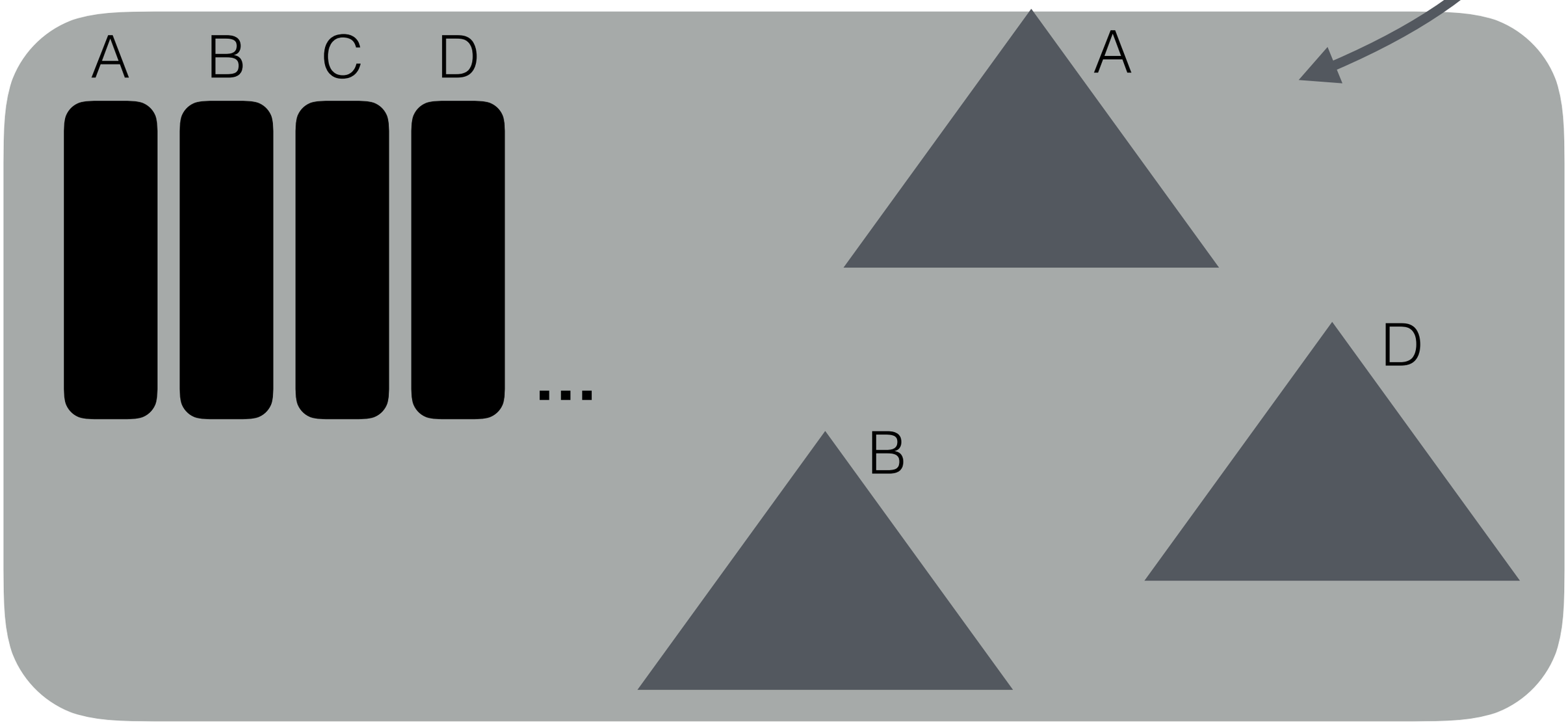


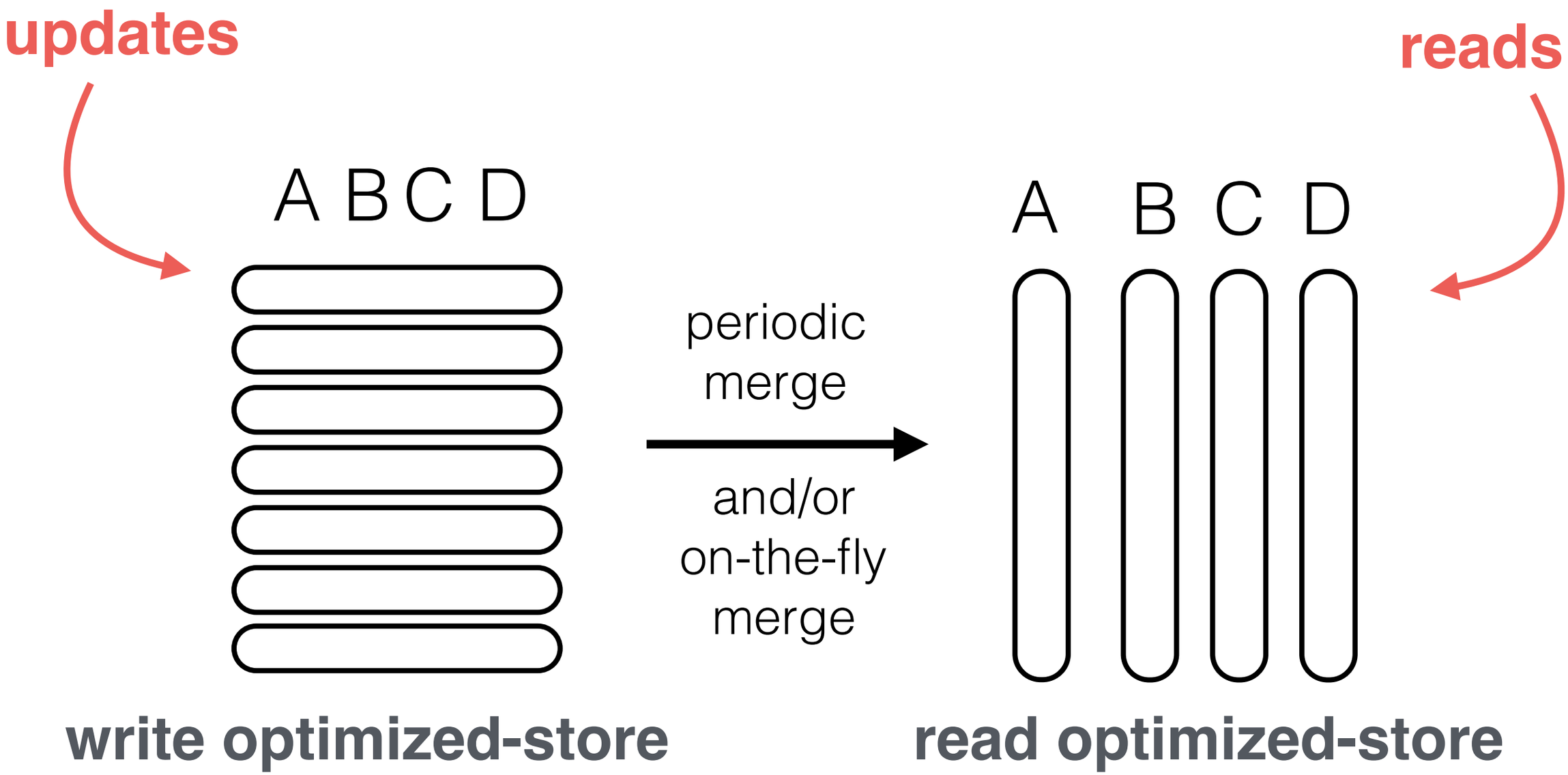
monitor memory hierarchy utilization

monitor clicks
(frequency, locations, specific links, sequences)

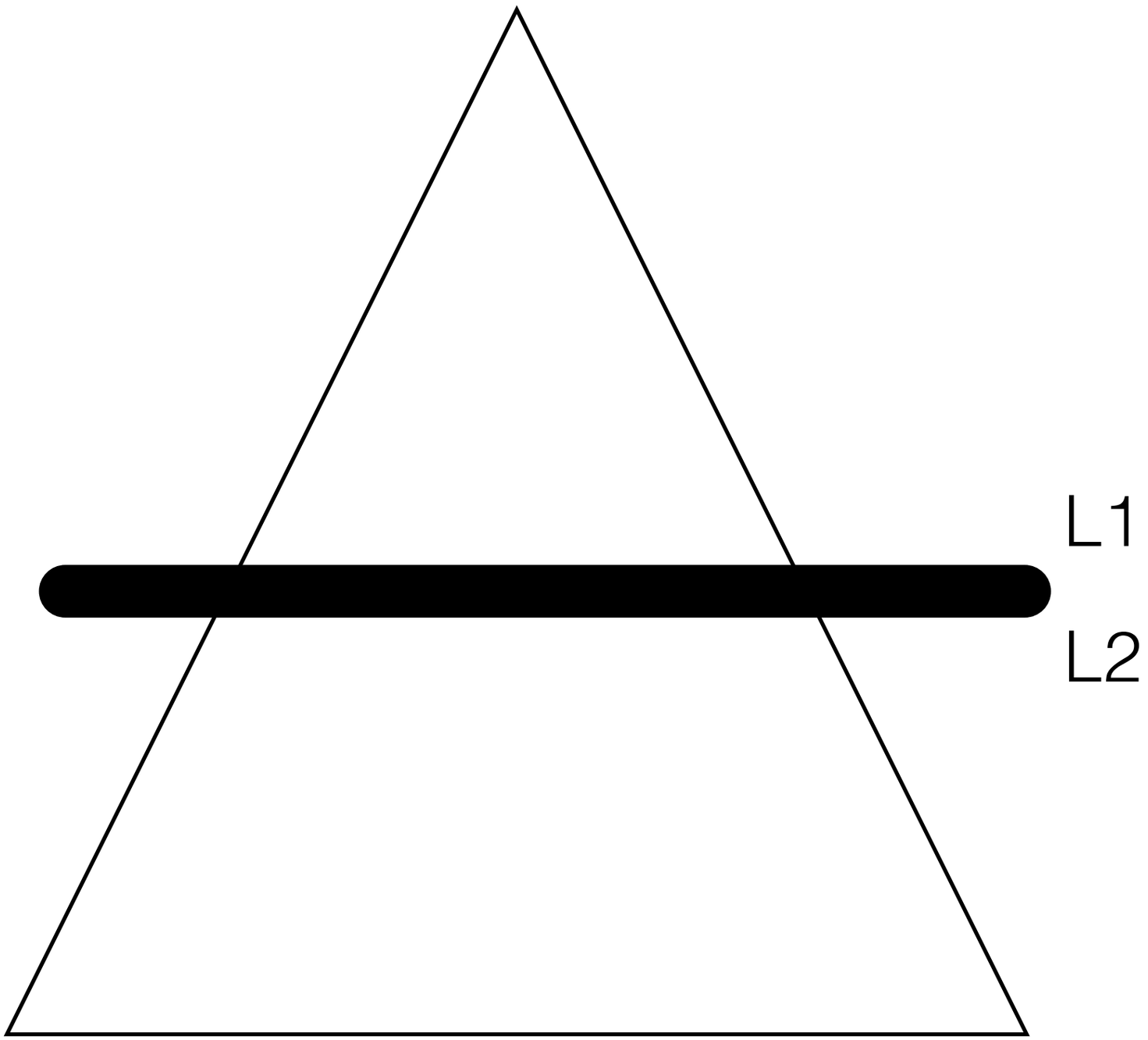
insert new entry (a,b,c,d,…) on table x

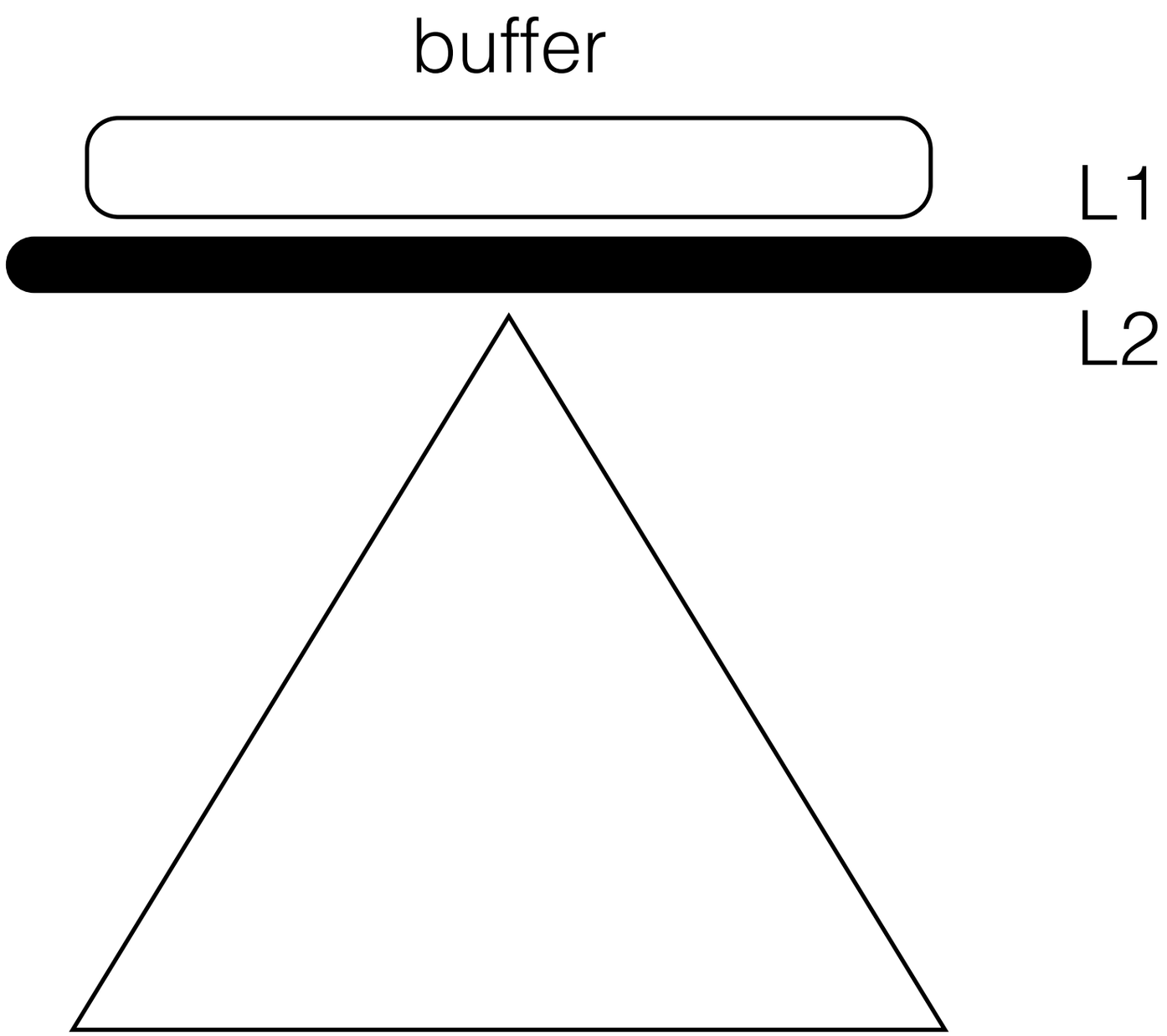
update N columns, K trees, statistics, …

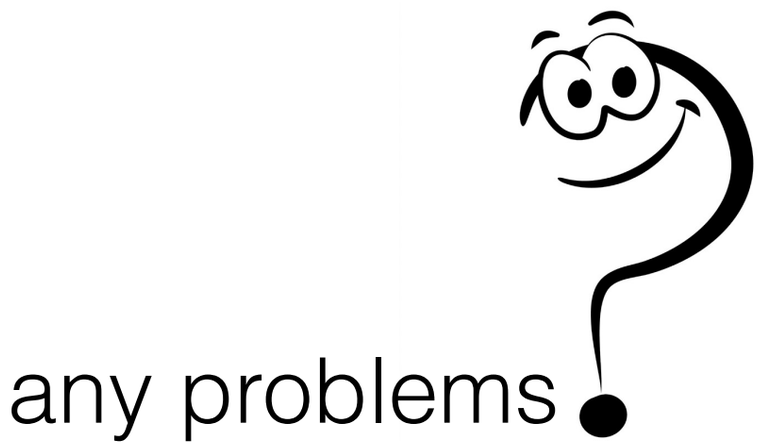


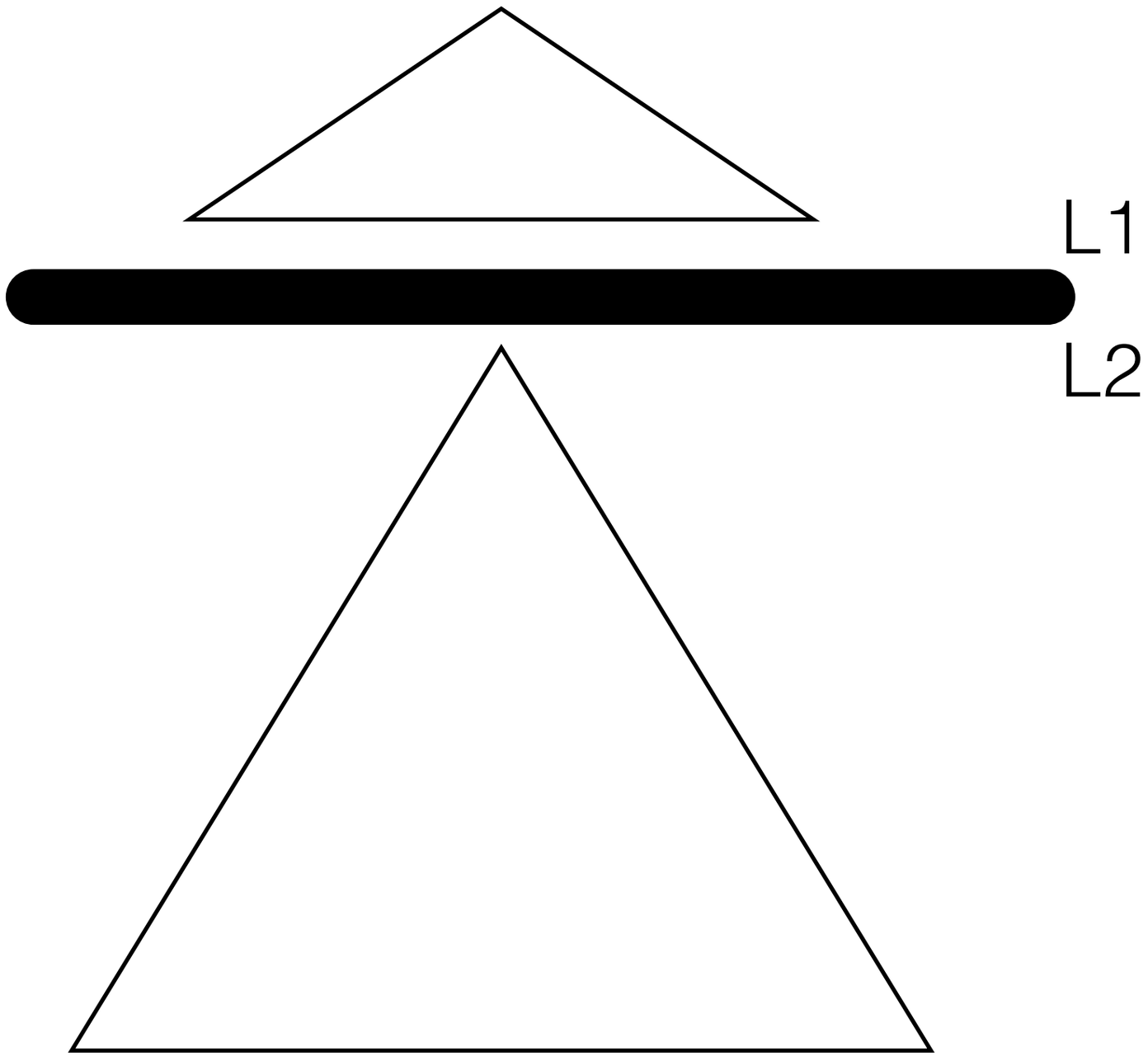


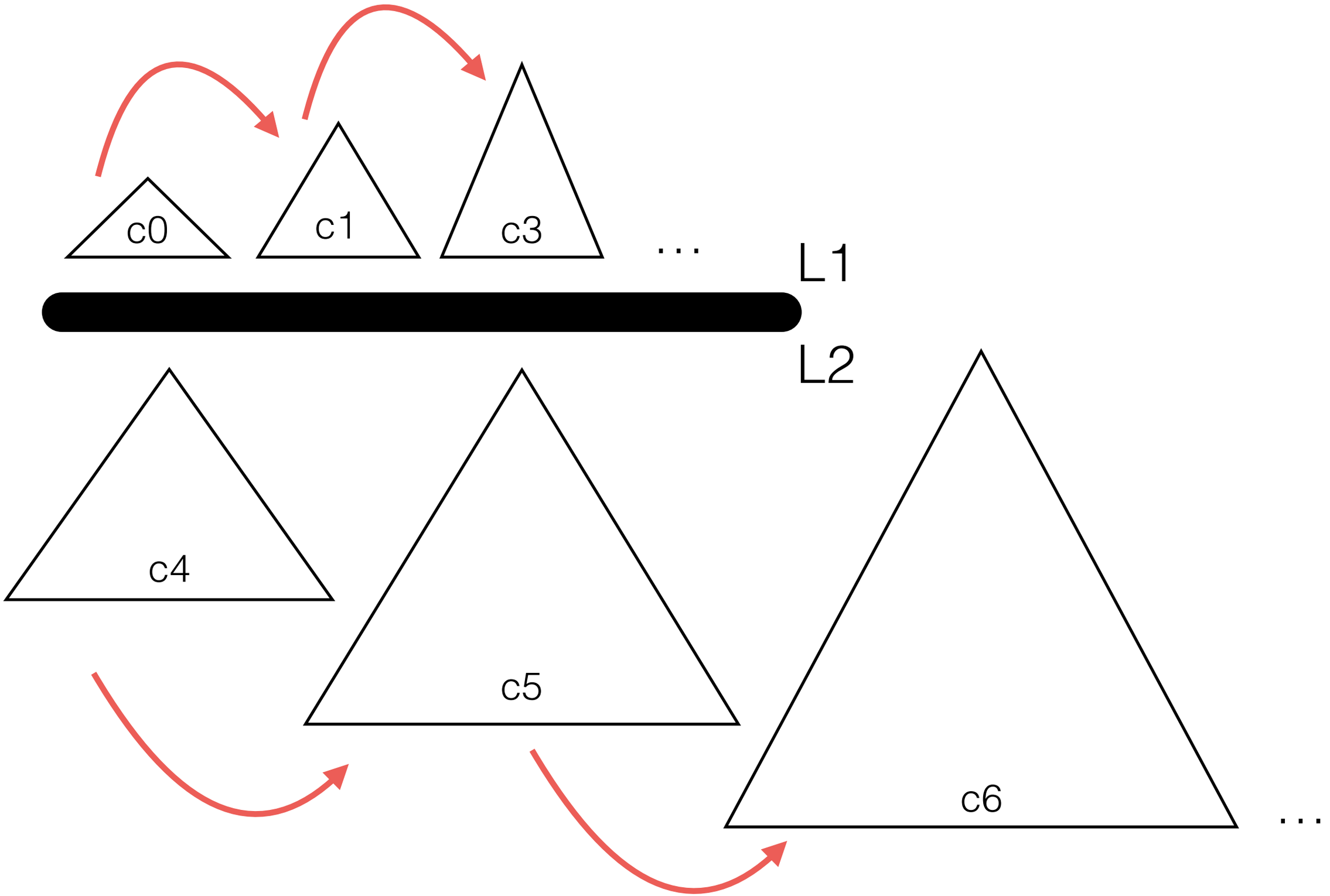
A case for fractured mirrors
Ravishankar Ramamurthy, David J. DeWitt, Qi Su
Very Large Databases Journal (**VLDBJ**), 2003


L1
L2


buffer
L1
L2

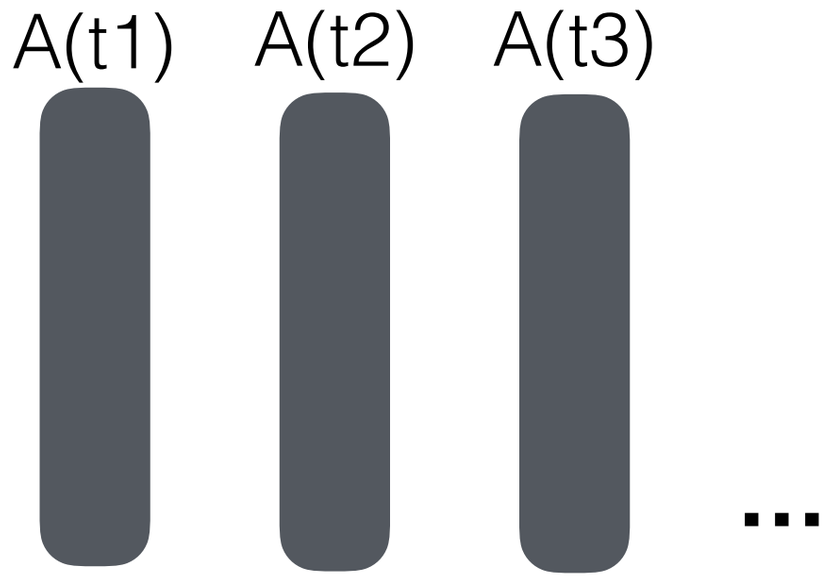

any problems ?


L1
L2


c0
c1
c3
...
L1
L2
c4
c5
c6
...

in-place updates: the cardinal sin

The Transaction Concept: Virtues and Limitations
Jim Gray, Tandem TR 81.3, 1981

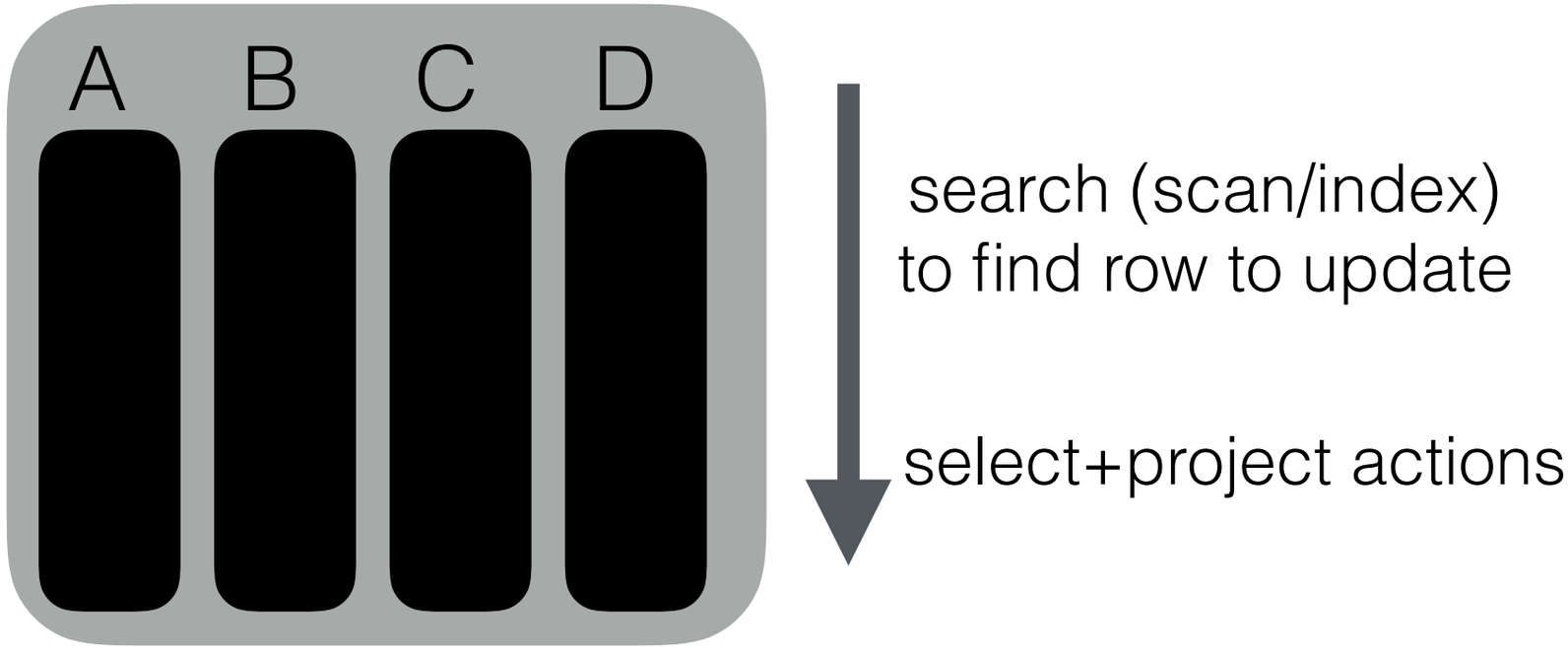


what can go wrong?

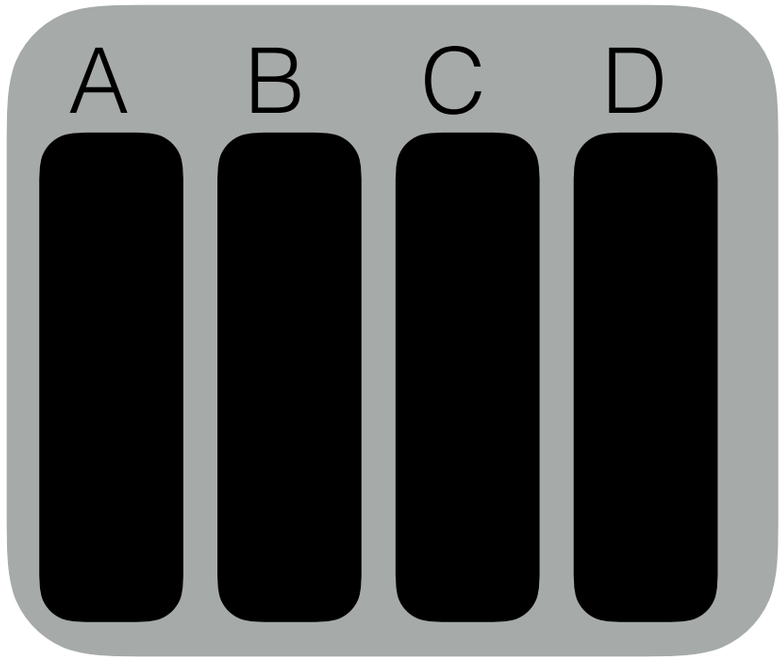
failures & >1 queries

update all rows
where A=v1 & B=v2
to (a=a/2,b=b/4,c=c-3,d=d+2)

update all rows
where A=v1 & B=v2
to (a=a/2,b=b/4,c=c-3,d=d+2)



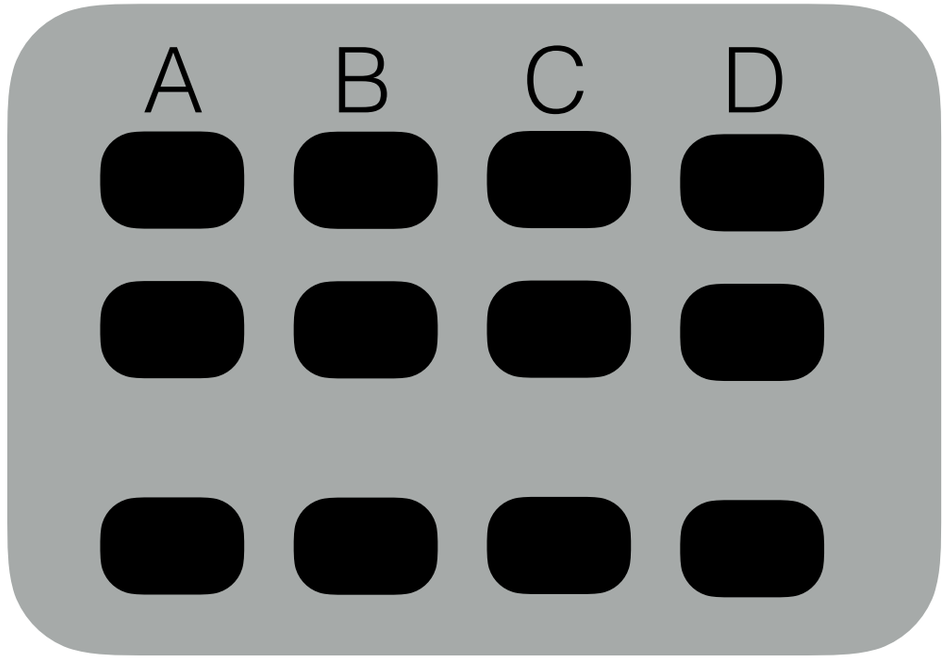
update all rows
where A=v1 & B=v2
to (a=a/2,b=b/4,c=c-3,d=d+2)



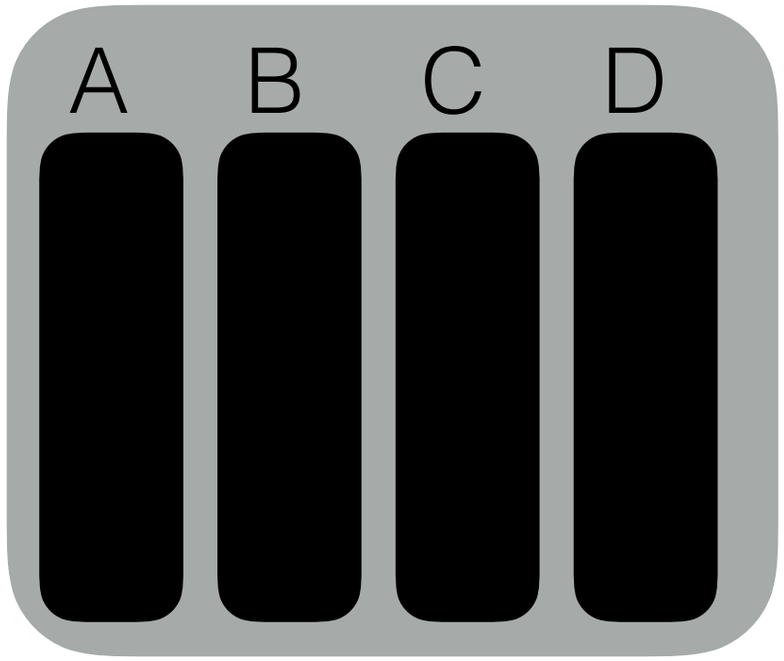
search (scan/index)
to find row to update

select+project actions

list of rowIDs (positions)
(sort)

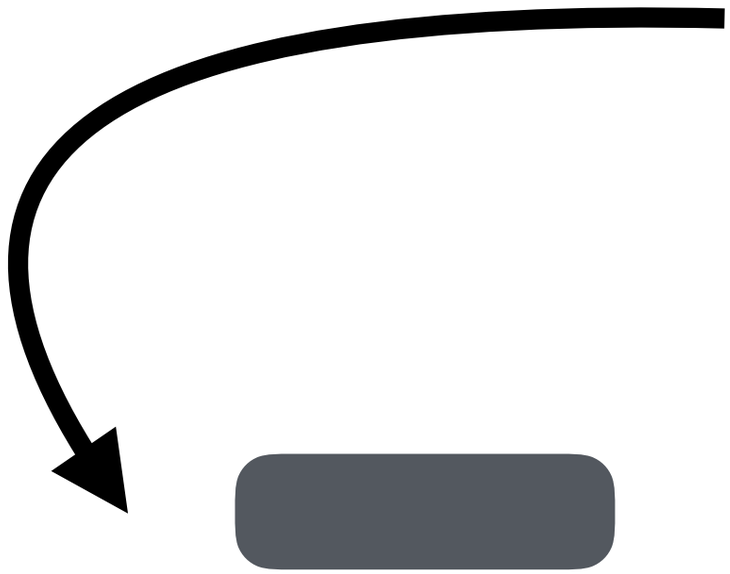


update all rows
where A=v1 & B=v2
to (a=a/2,b=b/4,c=c-3,d=d+2)

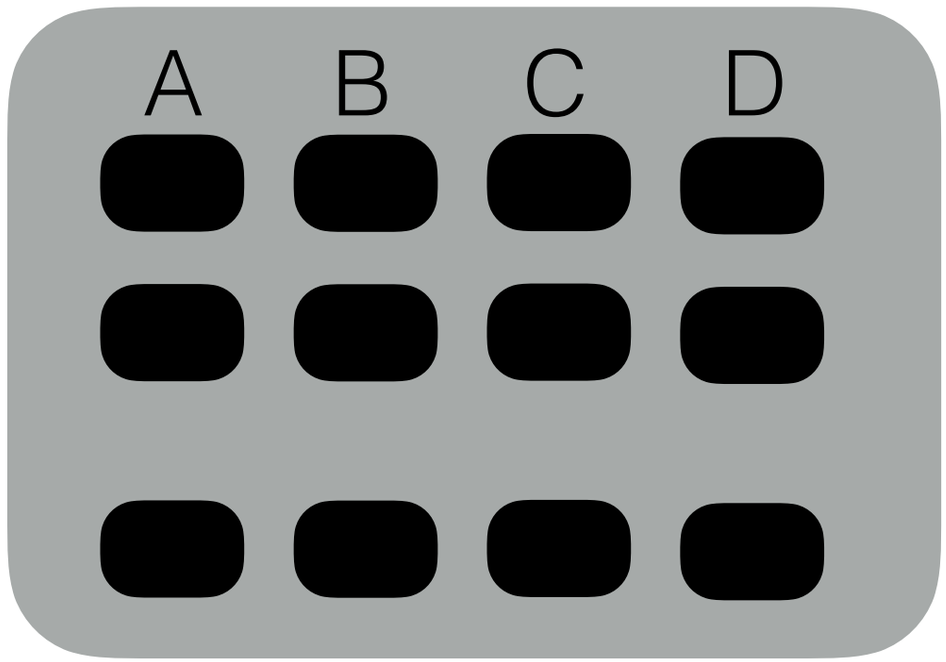


search (scan/index)
to find row to update

select+project actions

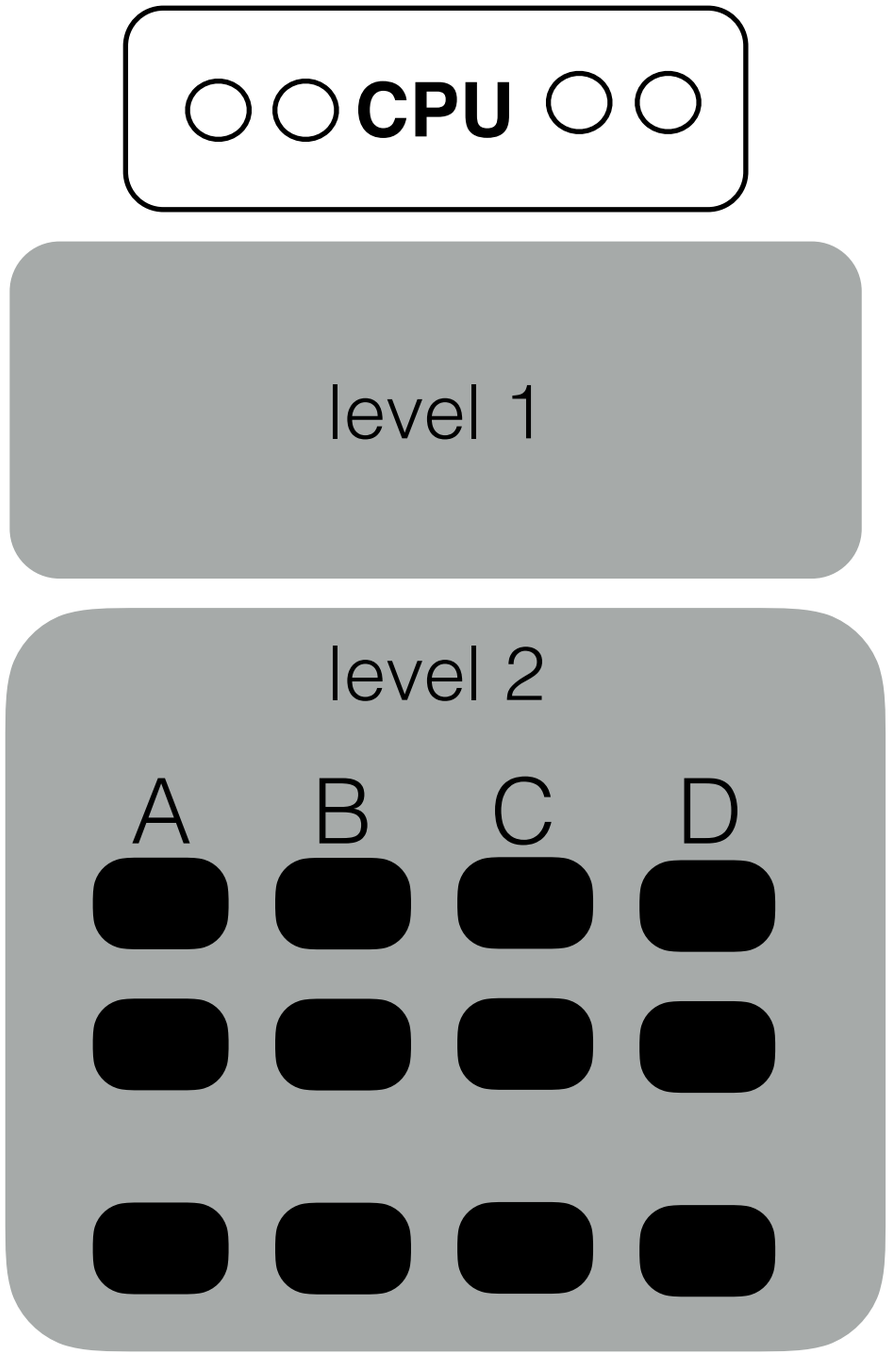


list of rowIDs (positions)
(sort)



we know what to update but nothing happened yet

update all rows where A=v1 & B=v2 to (a=a/2,b=b/4,c=c-3,d=d+2)

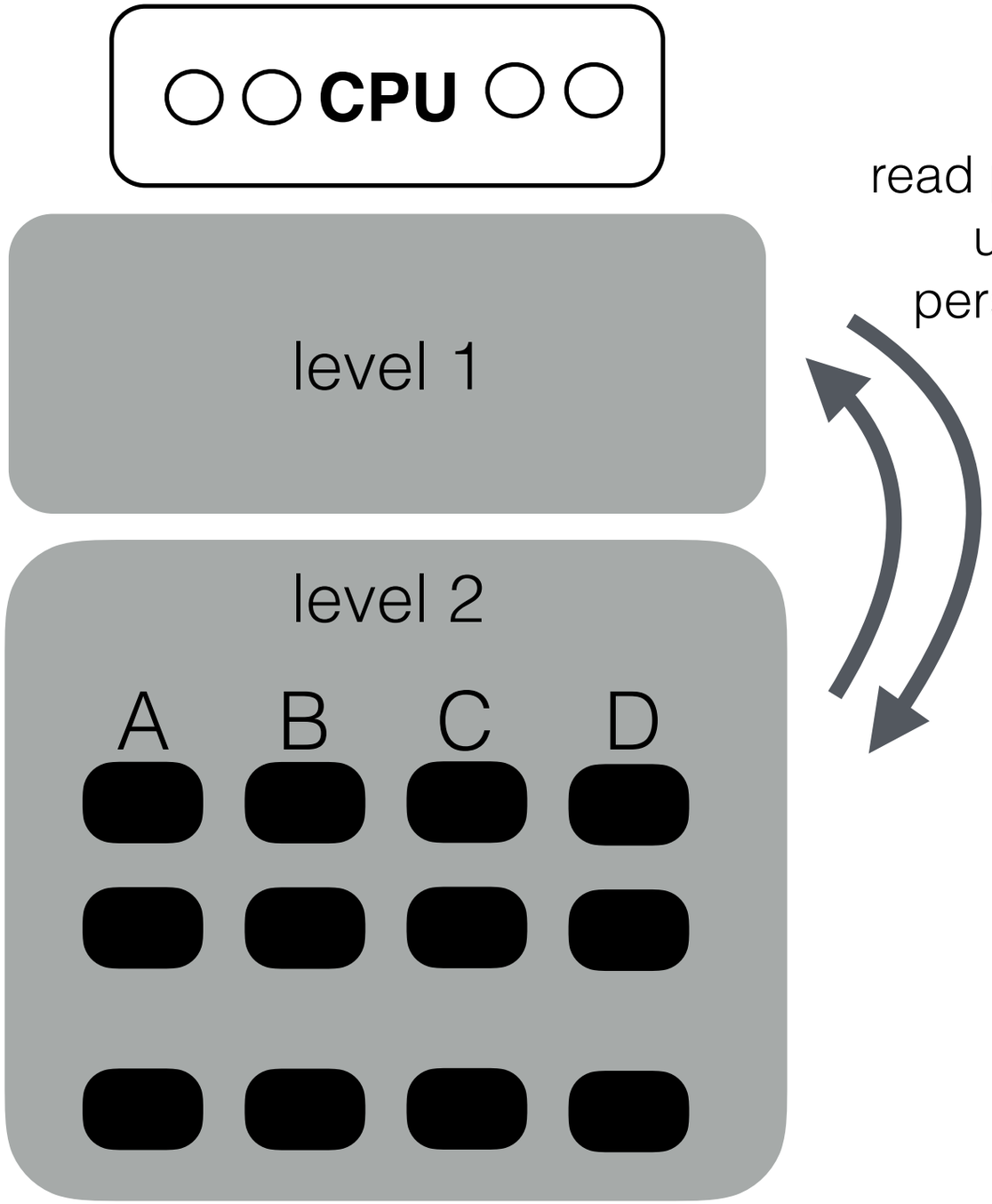


read page in L1
update
persist to L2

if problem (power/abort)
before we write all pages
we are left with an inconsistent state

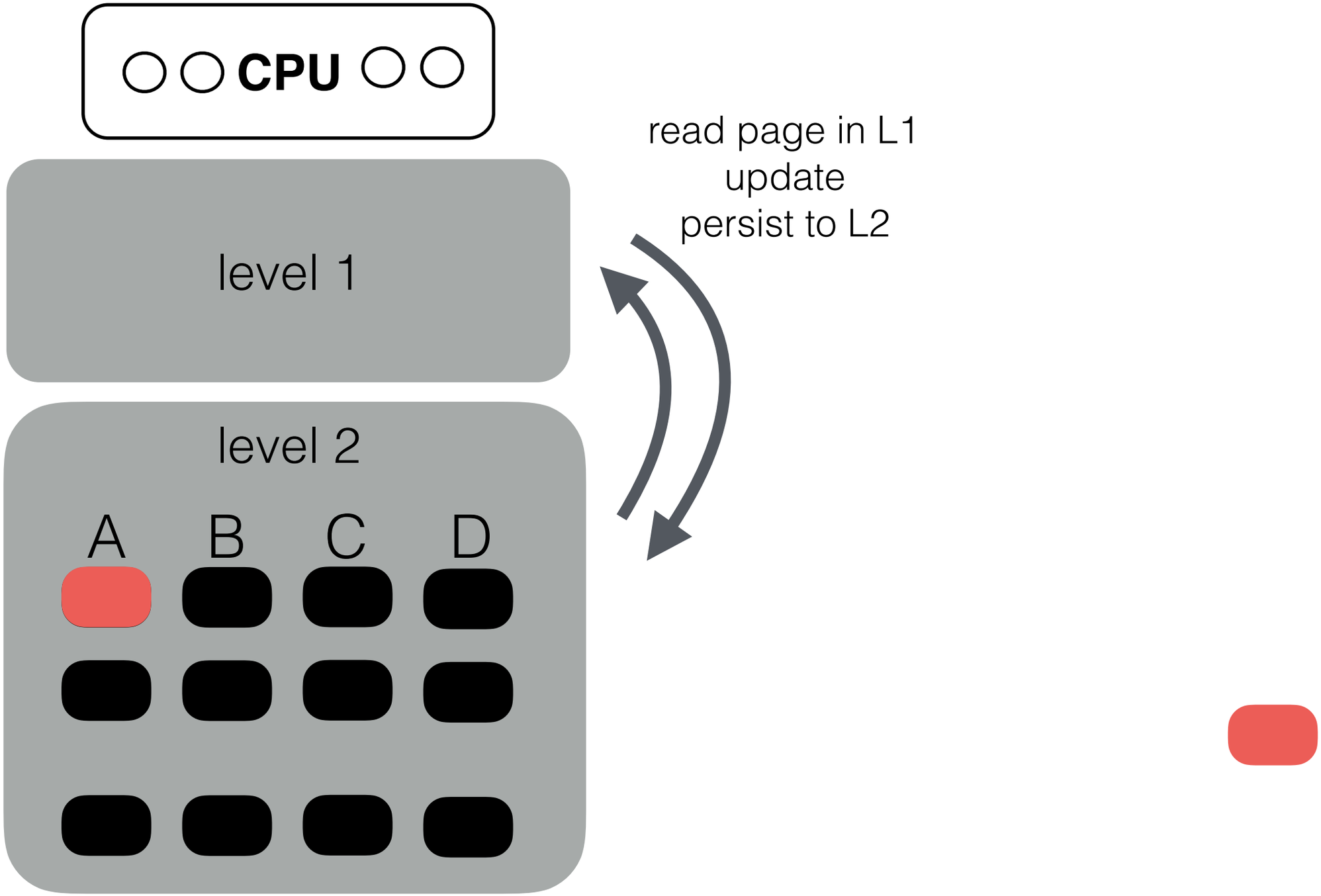
WAL: keep persistent notes as we go so we can resume or undo

update all rows where A=v1 & B=v2 to (a=a/2,b=b/4,c=c-3,d=d+2)

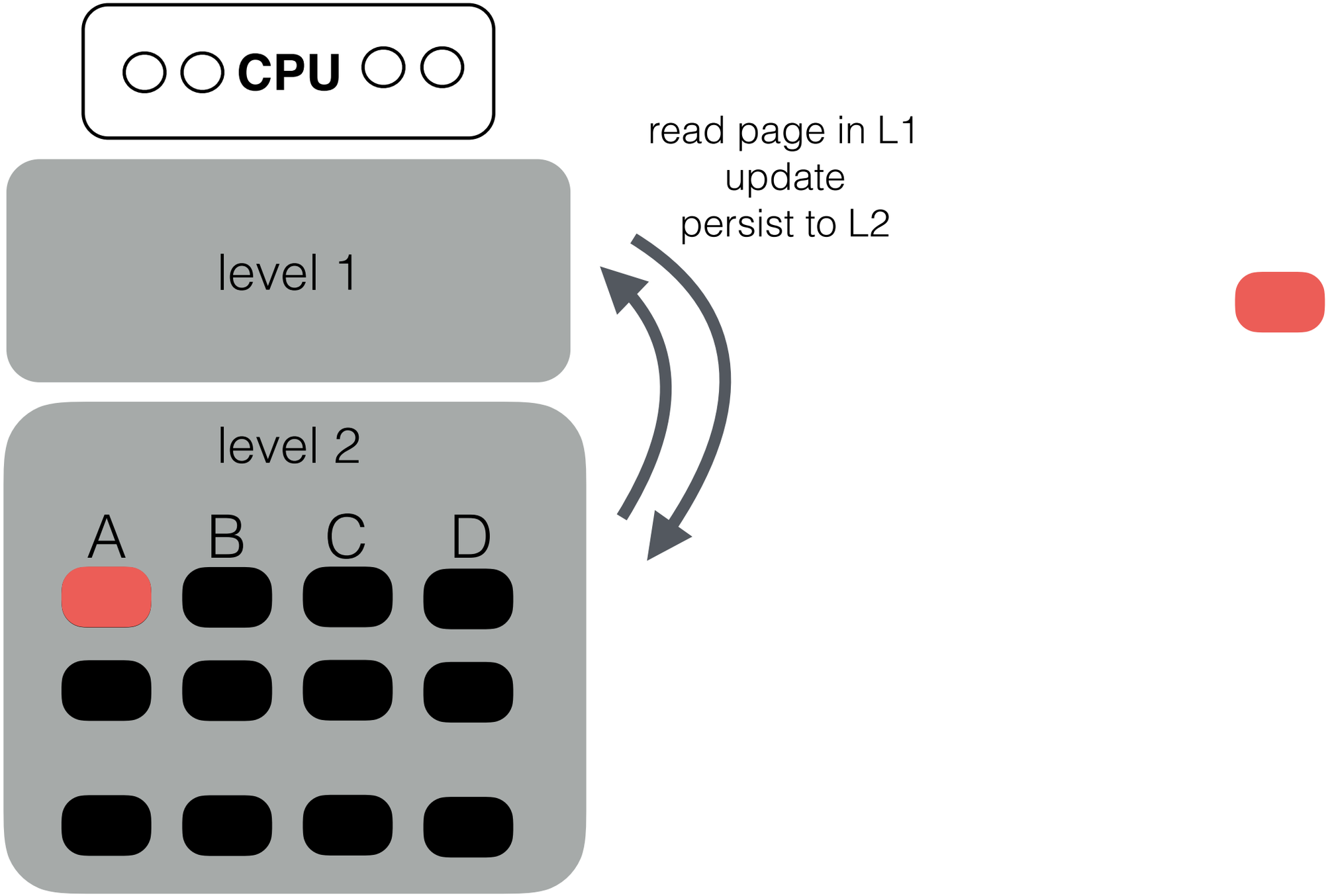


read page in L1
update
persist to L2

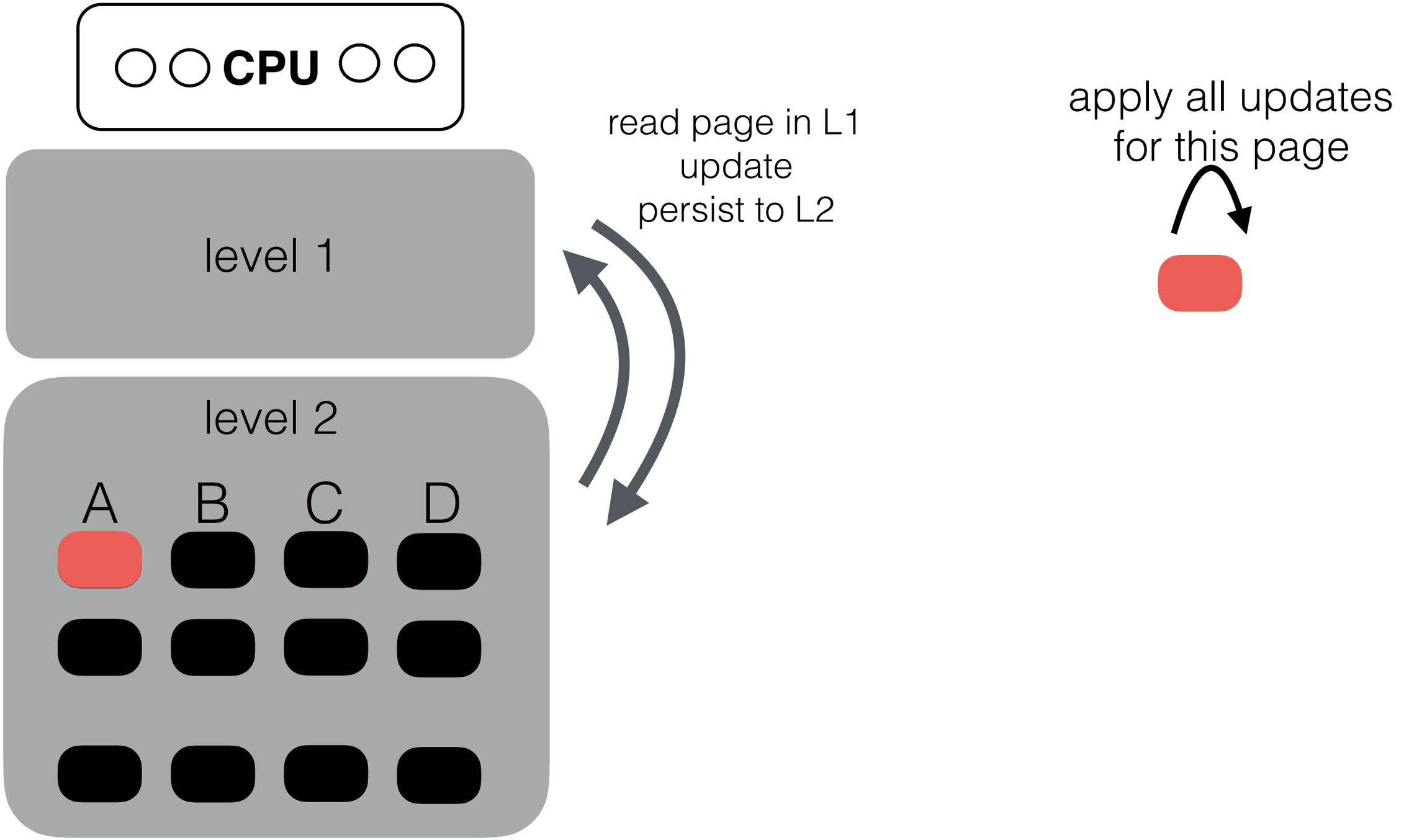
update all rows where A=v1 & B=v2 to (a=a/2,b=b/4,c=c-3,d=d+2)



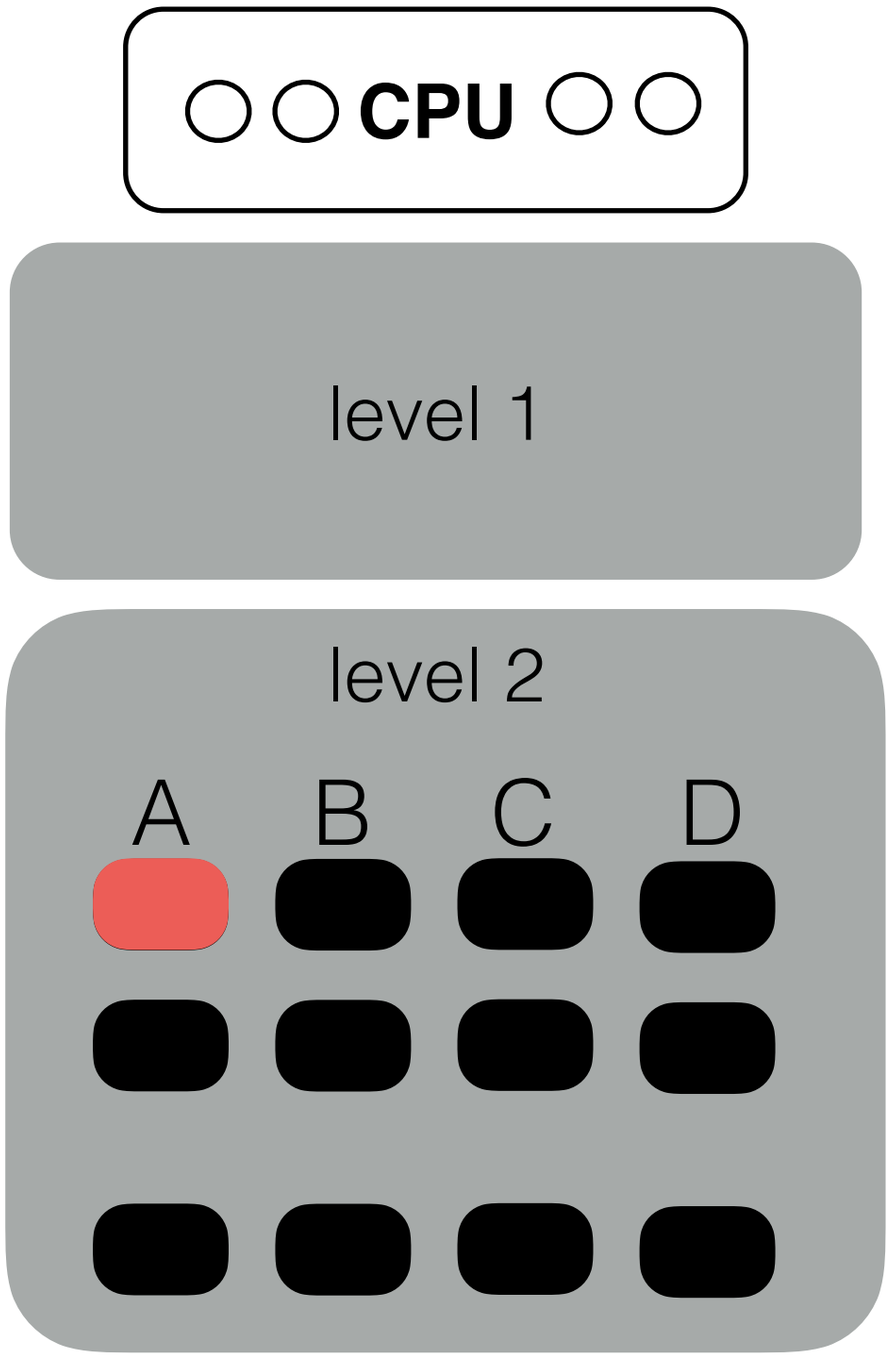
update all rows where A=v1 & B=v2 to (a=a/2,b=b/4,c=c-3,d=d+2)



update all rows where A=v1 & B=v2 to (a=a/2,b=b/4,c=c-3,d=d+2)



update all rows where A=v1 & B=v2 to (a=a/2,b=b/4,c=c-3,d=d+2)

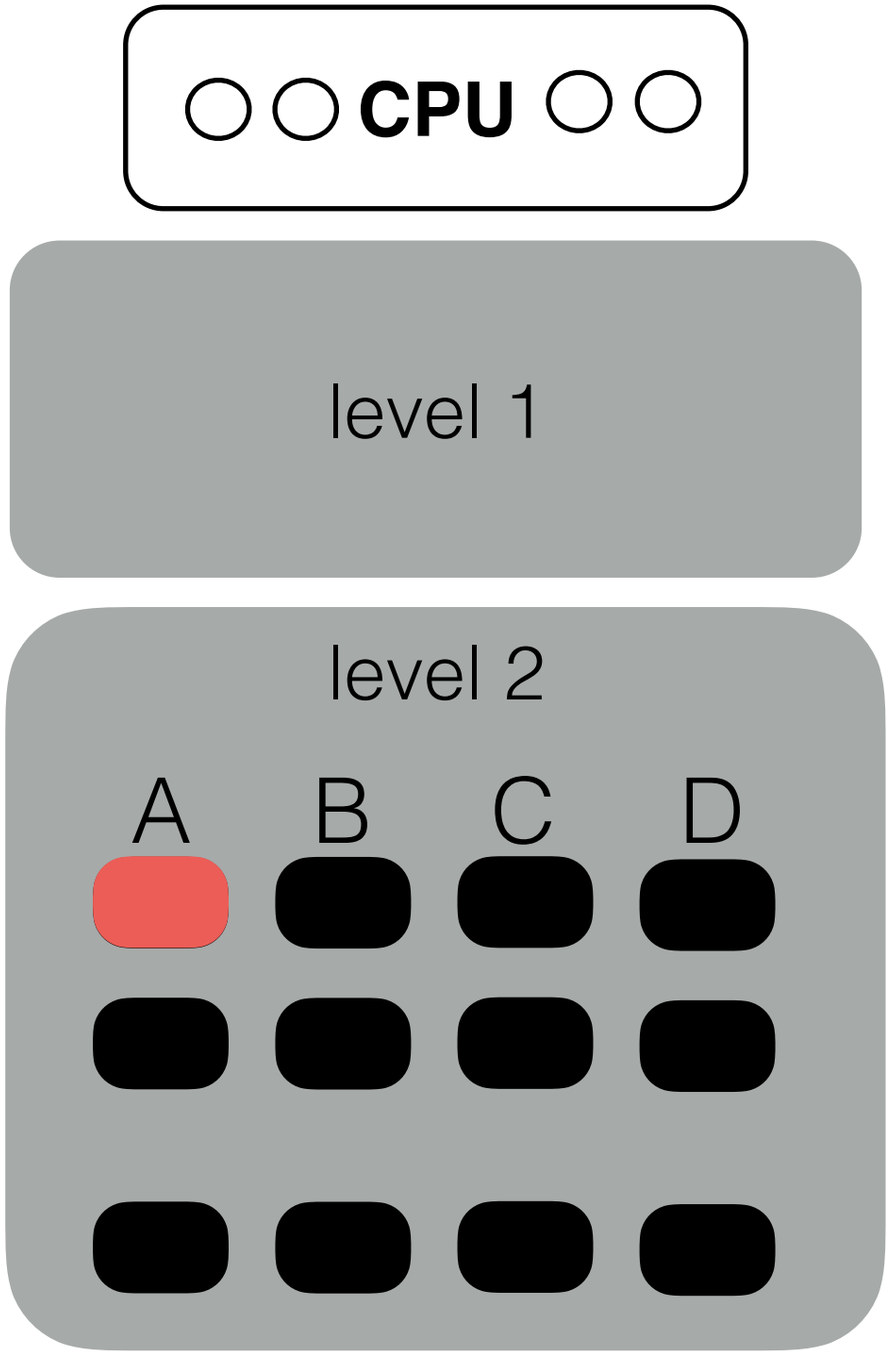


read page in L1
update
persist to L2

apply all updates
for this page

N updates in
persistent storage

update all rows where A=v1 & B=v2 to (a=a/2,b=b/4,c=c-3,d=d+2)

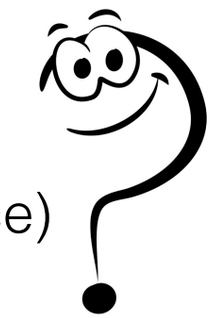


read page in L1
update
persist to L2

apply all updates
for this page

N updates in
persistent storage

(update in place/out of place)
problems when failure

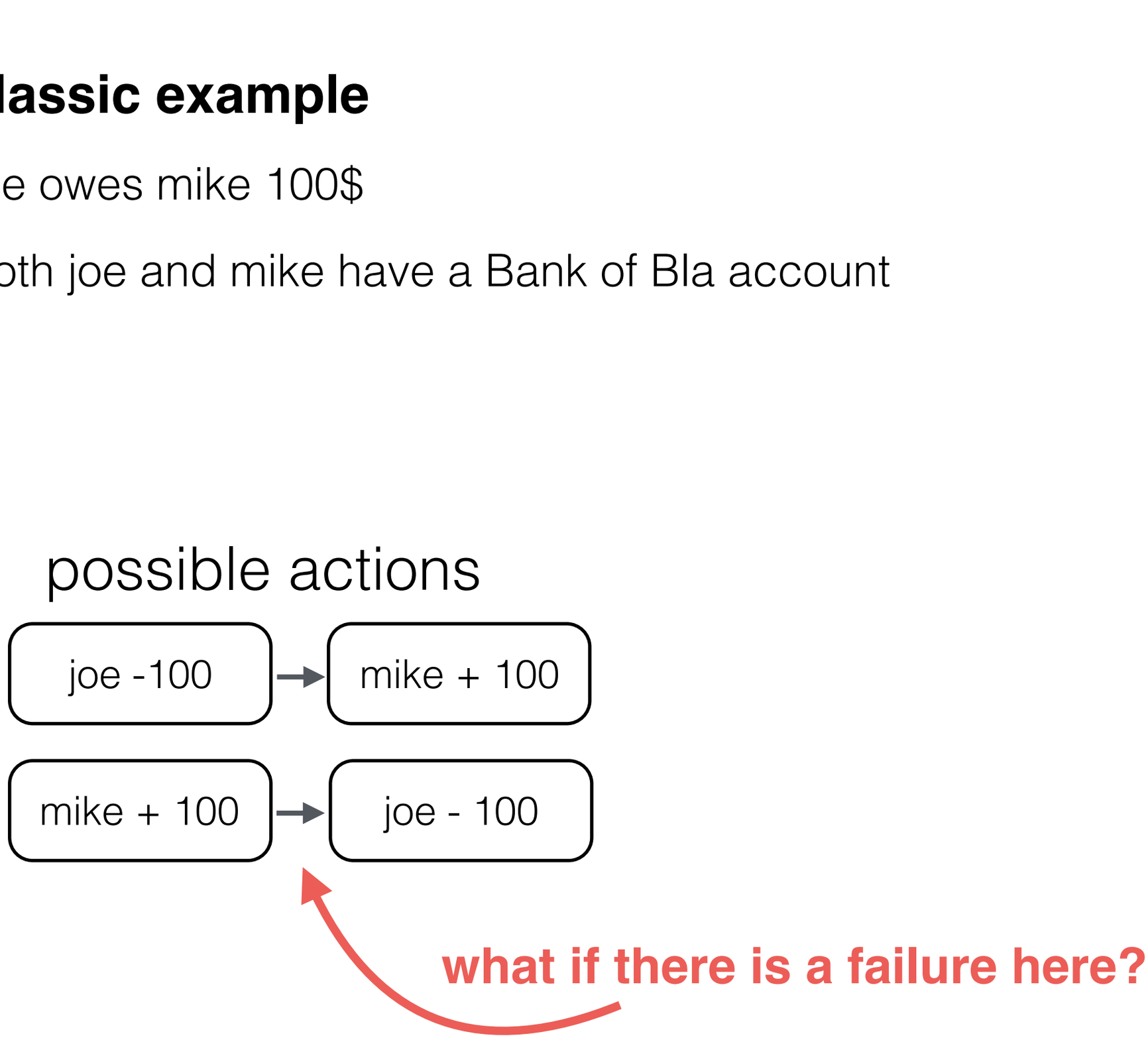


classic example

joe owes mike 100$

both joe and mike have a Bank of Bla account

possible actions



classic example

joe owes mike 100$

both joe and mike have a Bank of Bla account

possible actions

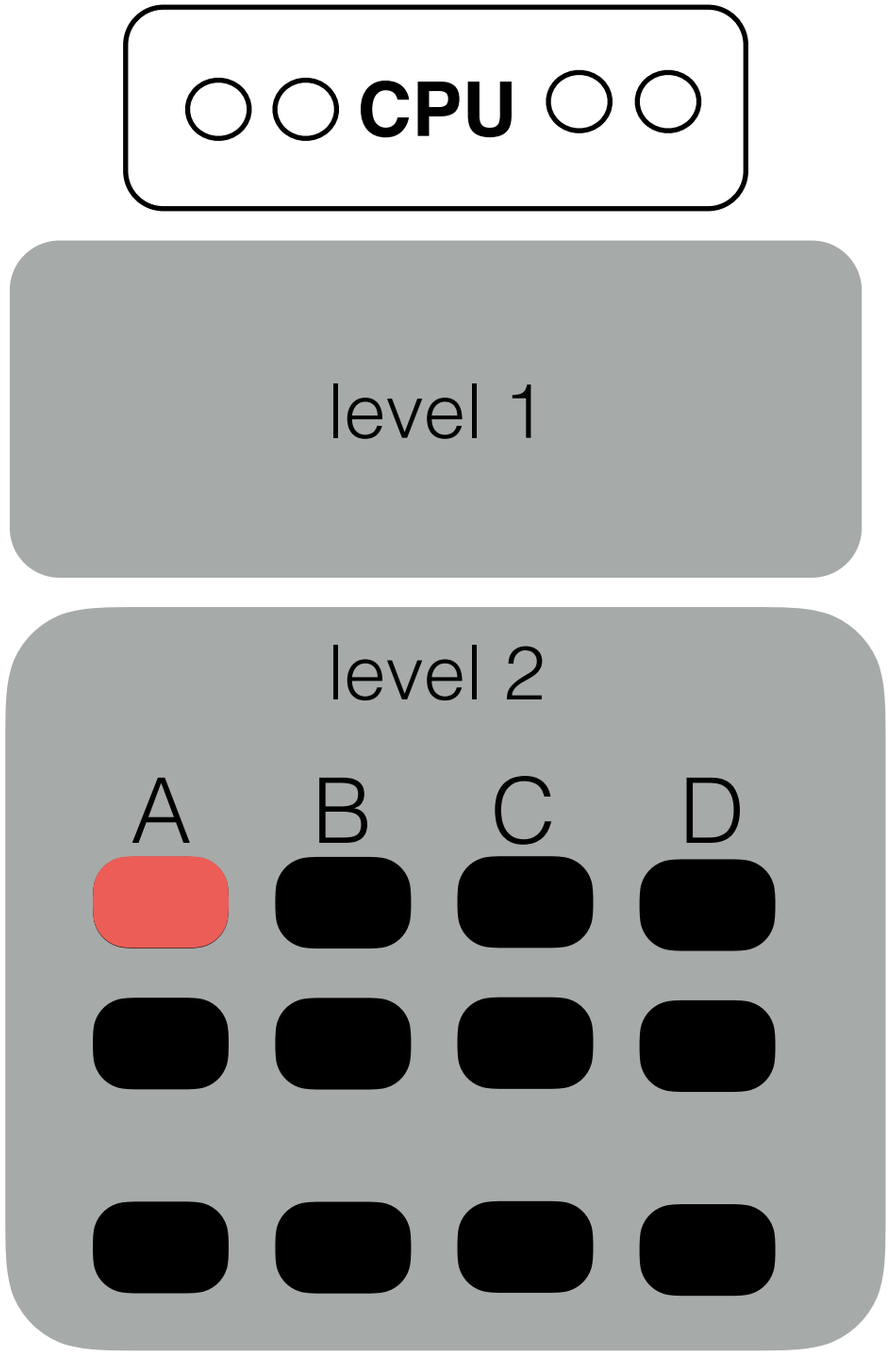
joe -100 → mike + 100

mike + 100 → joe - 100

actually
1)read mike; 2) mike+100; 3) write new mike;

what if there is a failure here?

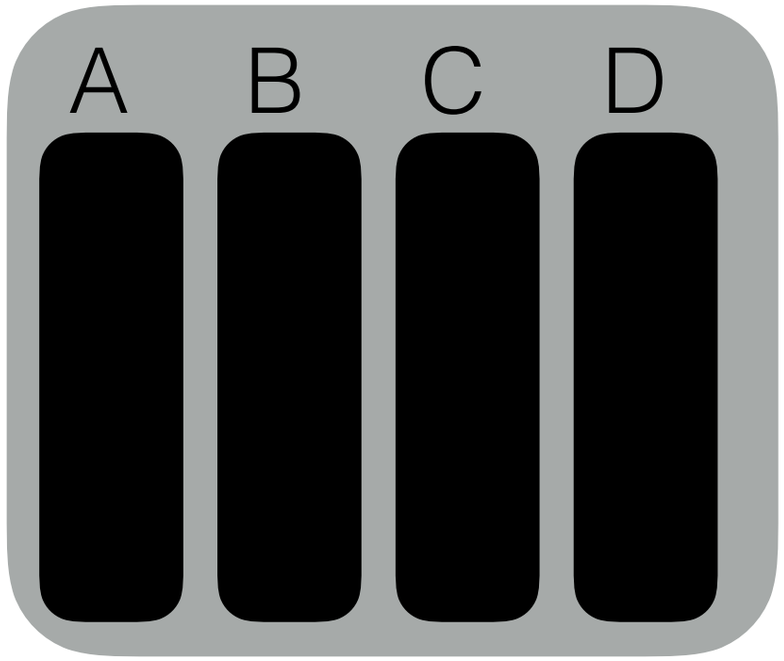
update all rows where A=v1 & B=v2 to (a=a/2,b=b/4,c=c-3,d=d+2)



read page in L1
update
persist to L2

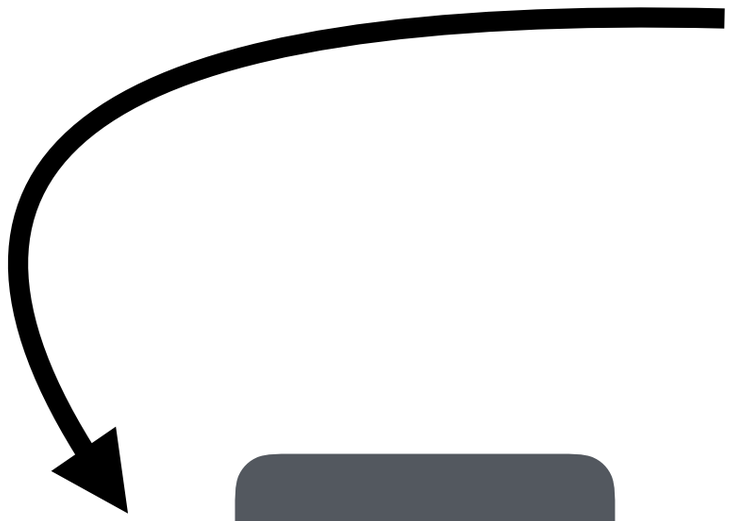
what do we need to remember (log)

update all rows
where A=v1 & B=v2
to (a=a/2,b=b/4,c=c-3,d=d+2)

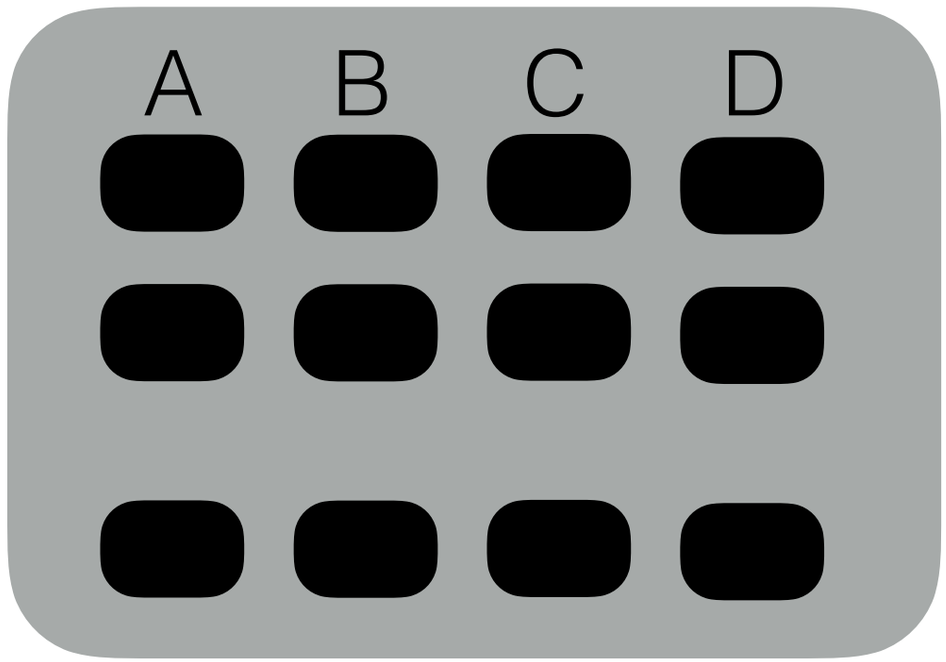


search (scan/index)
to find row to update

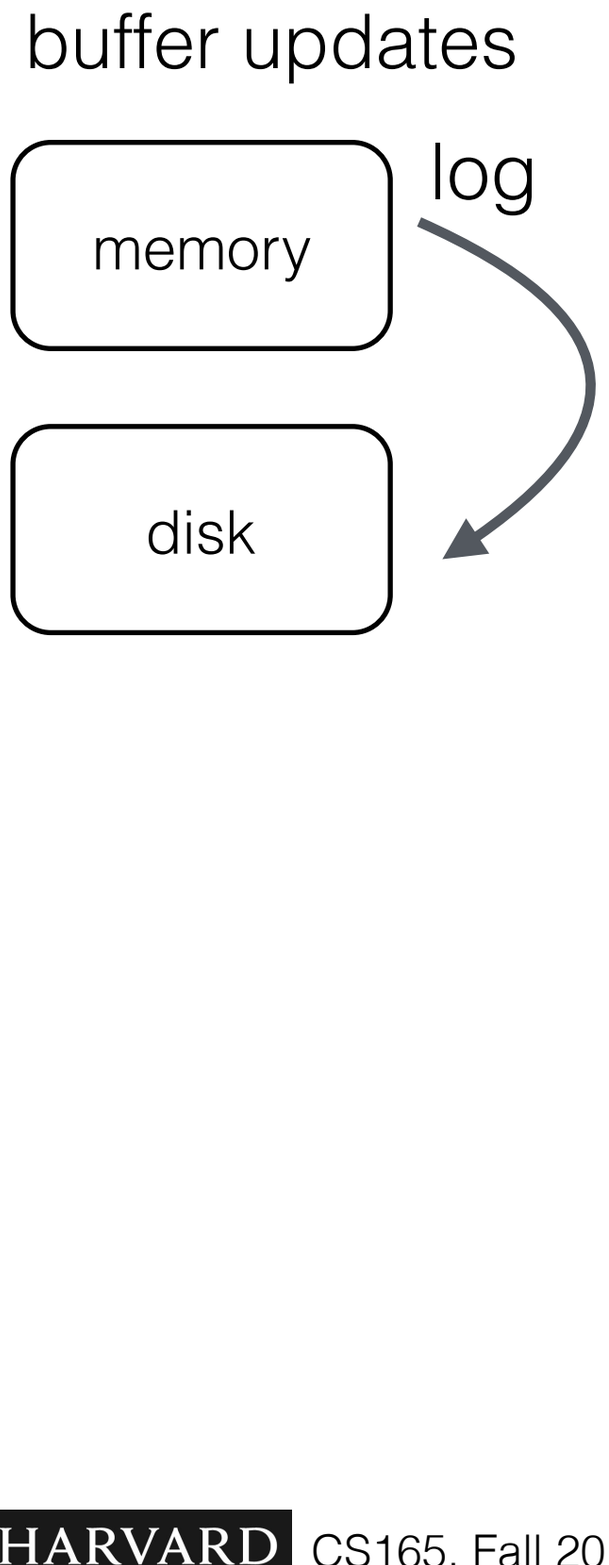
select+project actions

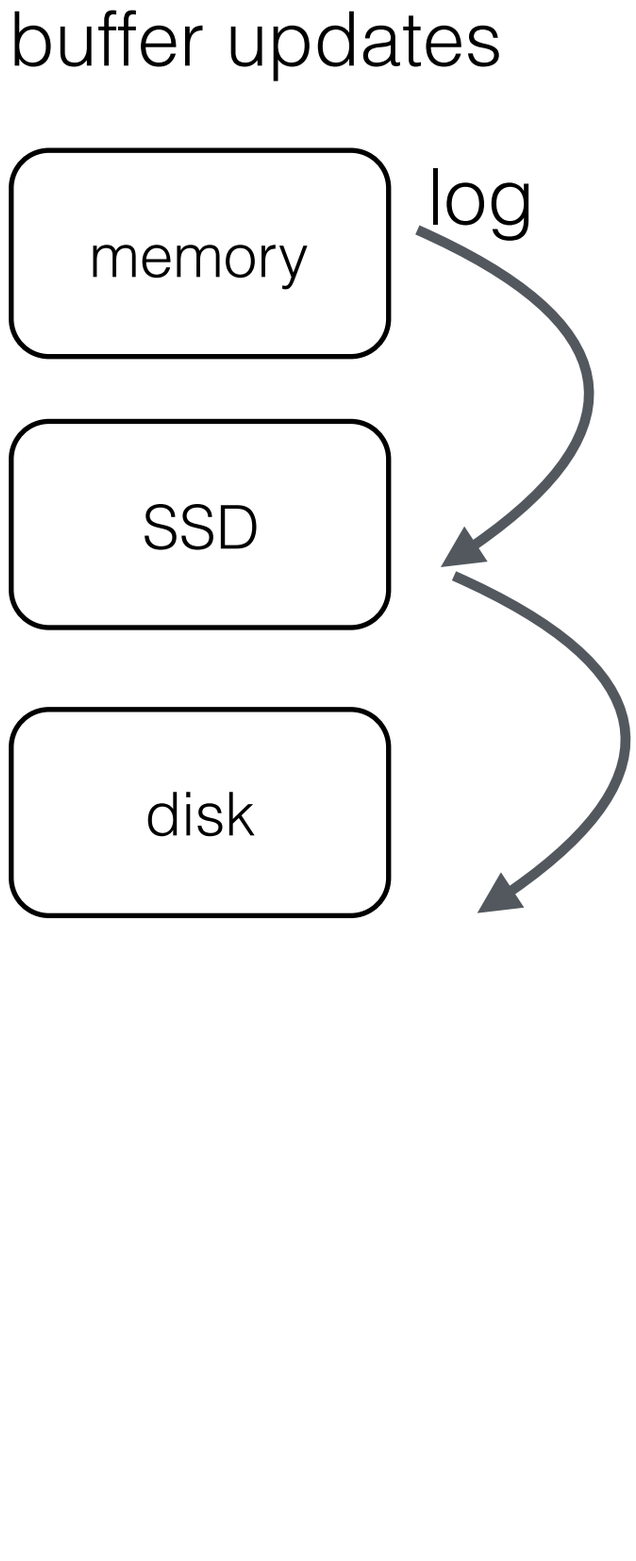


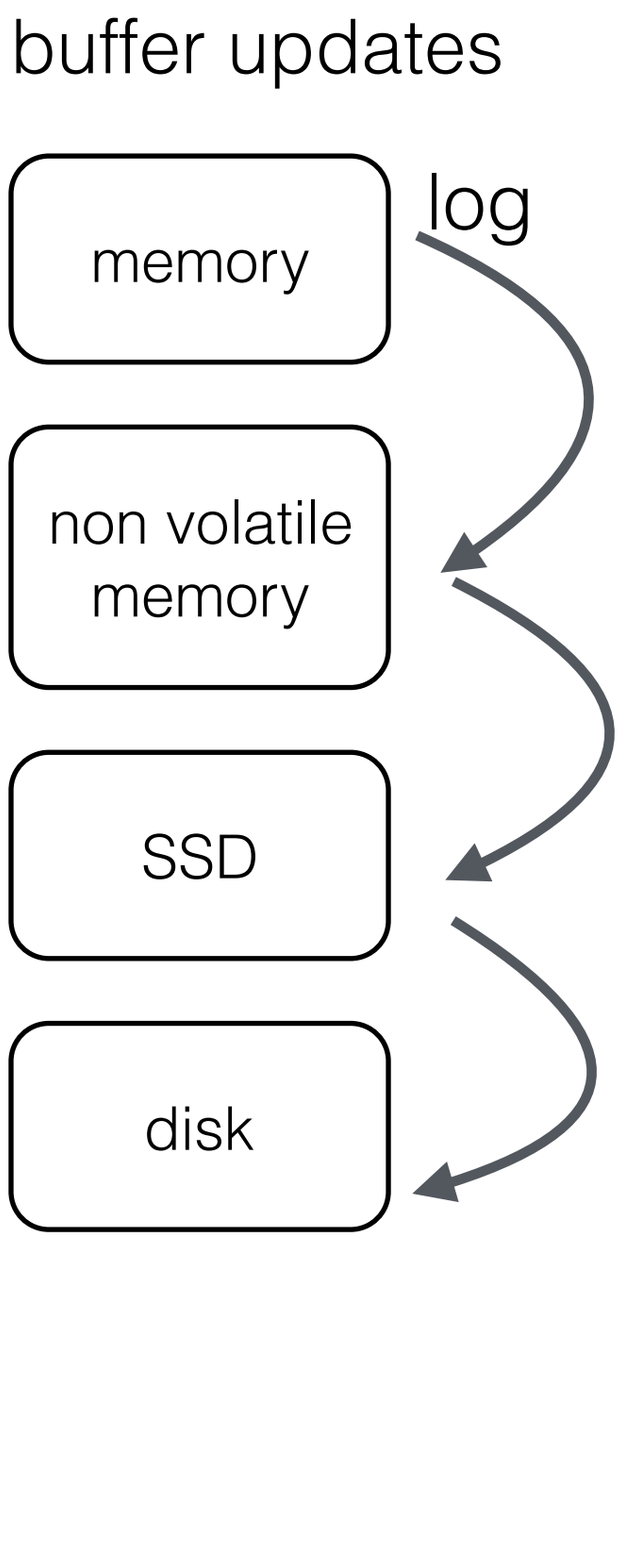
list of rowIDs (positions)
(sort)



restart: get set of positions again, and either resume from last written page
or undo all previously written pages


buffer updates
memory
log
disk


buffer updates
memory
log
SSD
disk


buffer updates
memory
log
non volatile
memory
SSD
disk

flash disks

read/write asymmetry

need to erase a block to write it

finite number of erase cycles

MaSM: efficient online updates in data warehouses

Manos Athanassoulis, Shimin Chen, Anastasia Ailamaki, Phillip Gibbons, Radu Stoica
ACM SIGMOD Conference, 2011

C Mohan, IBM Research
ACM SIGMOD Edgar F. Codd Innovations award 1993

(rumor has it he has his own Facebook server)
(also check his noSQL lecture)

ARIES: A Transaction Recovery Method Supporting Fine-Granularity Locking and Partial Rollbacks Using Write-Ahead Logging
C. Mohan, Donald J. Haderle, Bruce G. Lindsay, Hamid Pirahesh, Peter M. Schwarz
ACM Transactions on Database Systems, 1992

classic example

joe owes mike 100$

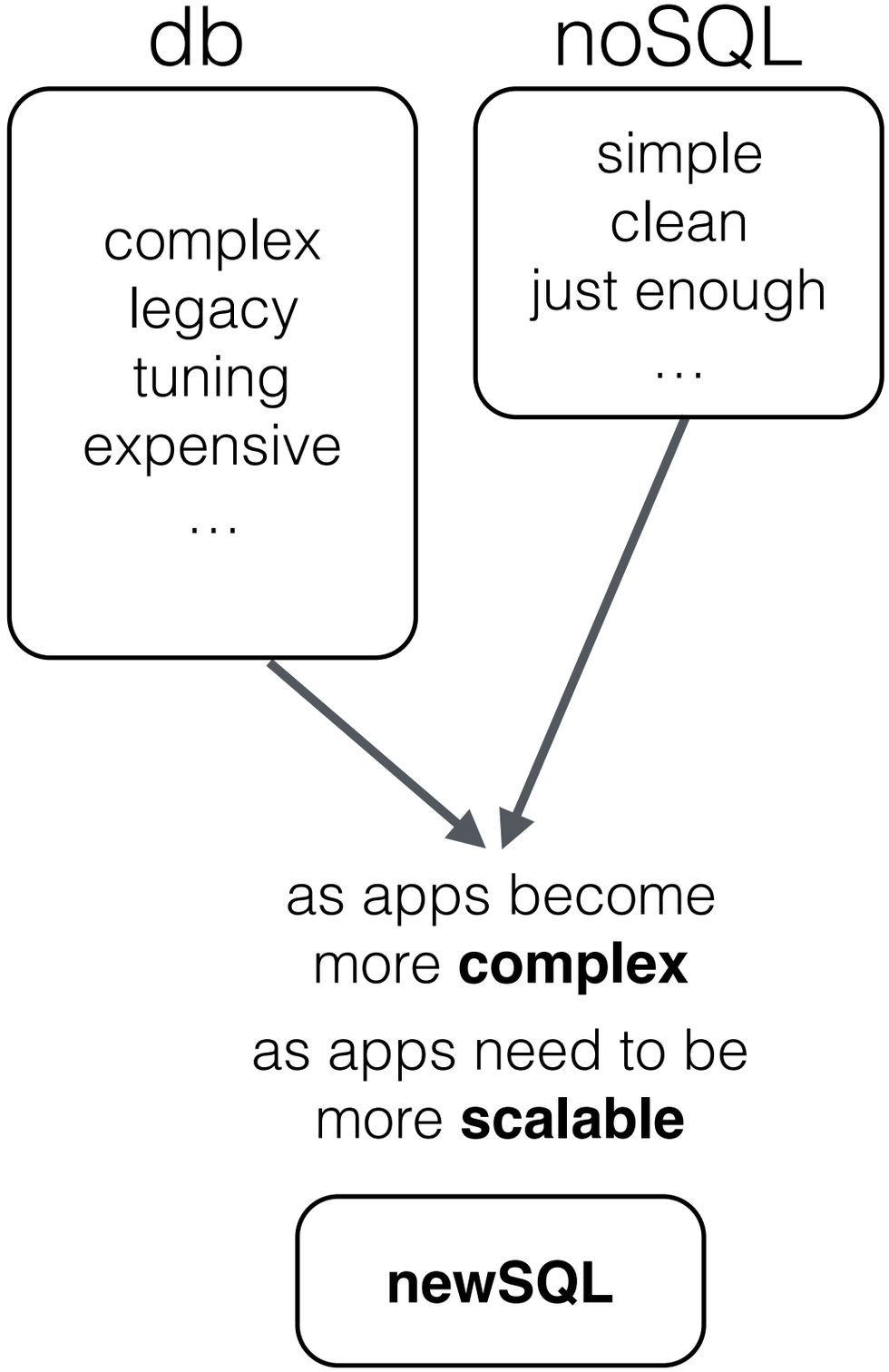
both joe and mike have a Bank of Bla account

what about logging and recovery during e-shopping

e.g., shopping cart, wish list, check out

Quantifying eventual consistency with PBS
Peter Bailis, Shivaram Venkataraman, Michael J. Franklin,
Joseph M. Hellerstein, Ion Stoica
Communications of the ACM, 2014

why **noSQL**



Mesa: Geo-Replicated, Near Real-Time, Scalable Data Warehousing
Ashish Gupta et al.
International Conference on
Very Large Databases (**VLDB**), 2014

F1: A Distributed SQL Database That Scales
Jeff Shute et al.
International Conference on
Very Large Databases (**VLDB**), 2013

(more in 265)

all or nothing

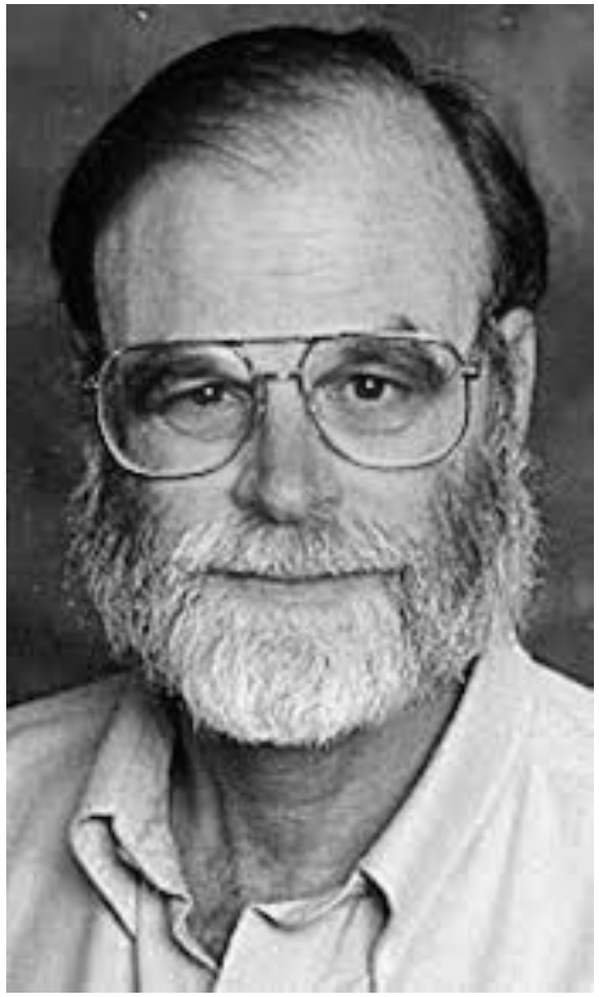
remember what we changed so we can undo or resume (depends on applications semantics)

transaction
any database query that
can/should be seen as a single task

parse
optimize

plan(
read X
write Z
…
)

Atomicity
Consistency
Isolation
Durability



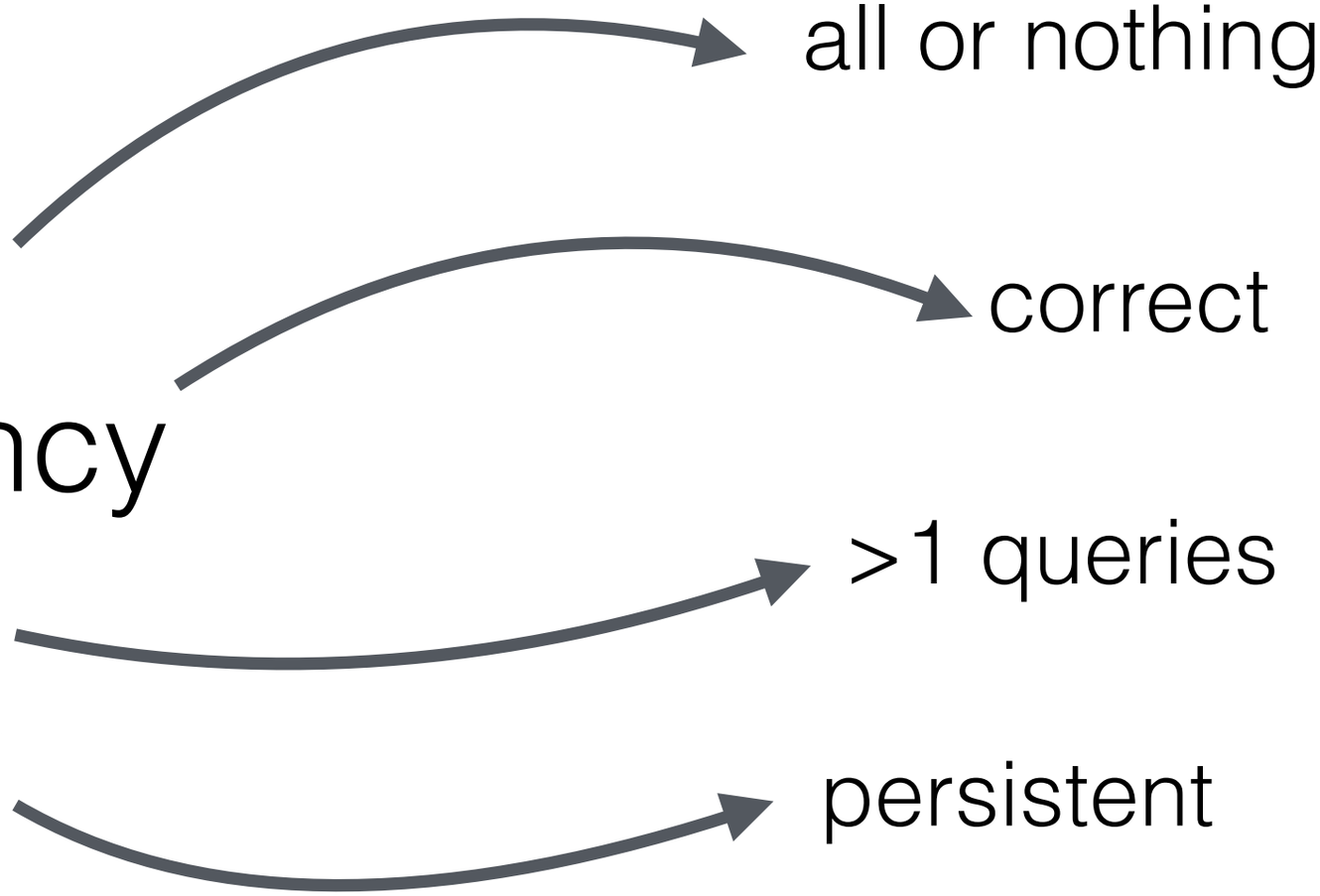
Jim Gray, IBM, Tandem, DEC, Microsoft
ACM Turing award
ACM SIGMOD Edgar F. Codd Innovations award

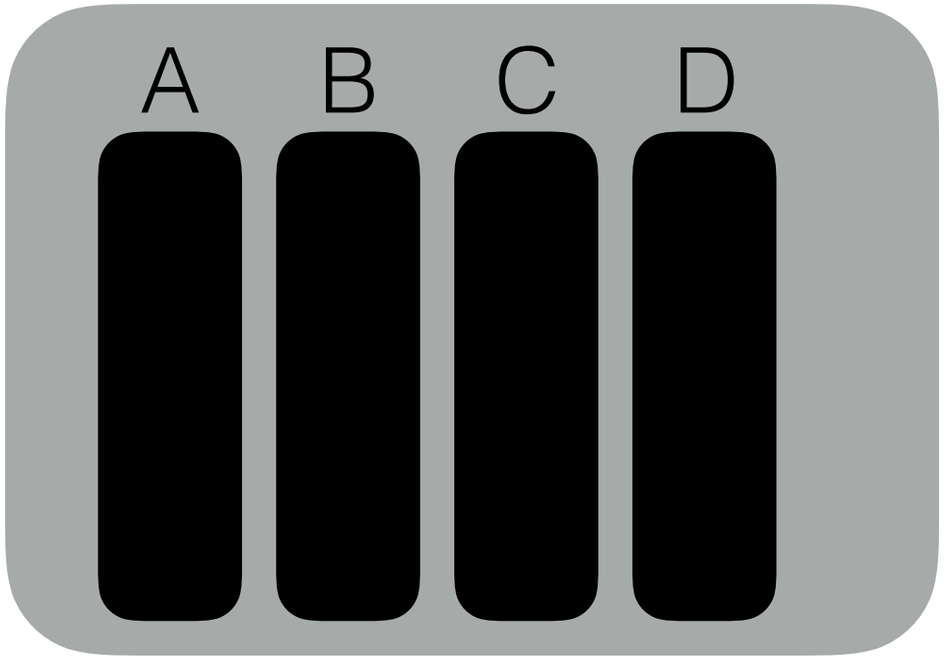
Atomicity

Consistency

Isolation

Durability





queries arrive concurrently
q1) select A>v1, B=B+b, C=C-c
q2) select B>v2, avg(C)
q3) select C<v3 & B>v4, A=C
q4) write more queries

Goal:
process queries in parallel
leave db in a consistent state
return correct results

to be released after class

to be released after class

to be released after class

to be released after class

textbook: chapters 16, 17, 18

updates 2.0

DATA SYSTEMS

prof. Stratos Idreos