class 19

updates

prof. Stratos Idreos

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

early/late tuple reconstruction, tuple-at-a-time, vectorized or bulk processing, intermediates format, pushing selects down, etc

scan, binary search, tuple reconstruction, min, max, search/update b-tree, join, etc.

arrays, columns, matrixes, rows, trees

cpu

memory

disk

algorithm/operators

data kernel

data

data

CS165, Fall 2015
Stratos Idreos
**UPDATE** table_name  
**SET** column1=value1,column2=value2,...  
**WHERE** some_column=some_value

**INSERT INTO** table_name  
**VALUES** (value1,value2,value3,...)
traditional applications
  e.g., banking

how many times per day do you send
update queries to your bank account
the world has changed a little bit by now…

updates
it is not just our data…
everything is data!
monitor CPU utilization

monitor memory hierarchy utilization

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

what & how
new data

analyze data as it arrives and react (standing queries)
merge incoming data with already archived data
see the correct up-to-date values
do not lose any updates (software/hardware failures)
>>1 updates concurrently
conflicting goals

(hardware and requirements change continuously and rapidly)

moving target

application requirements

hardware

performance

energy profile

budget
“Three things are important in the database world: performance, performance, and performance”

Bruce Lindsay, IBM
ACM SIGMOD  Edgar F. Codd Innovations award 2012

ture for both reads & writes
how to do fast (& correct) updates?
(more or less same way we do fast reads)

transactions
logging
lazy vs eager updates
fractured mirrors
locking
in-place or not
data (array)

value
data (array)

value

insertion
data (array)

delete

value
update

data (array)

value
data (array)

value

inserts, deletes, updates = deletes followed by inserts
data structure vs application updates

student{name, age, address, telephone, GPA, ...}
insert new entry \((a,b,c,d,\ldots)\) on table \(x\)

update \(N\) columns, \(K\) trees, statistics, \(\ldots\)
table x

A  B  C  D

secondary index on D

(to index or not to index?)

(more about this in last two classes)
Jim Gray, IBM, Tandem, DEC, Microsoft
ACM Turing award
ACM SIGMOD Edgar F. Codd Innovations award

100Kx disk
Pluto
2 years

100x memory
New York
1.5 hours

10x on board cache
this building
10 min

2x on chip cache
this room
1 min

registers
my head
~0
random access & page-based access

same for writes!

data value $x$... but have to read all of page 1

page1  page2  page3  ...
1. read input into stream buffer, hash and write to respective partition buffer
2. when input buffer is consumed, bring the next one
3. when a partition buffer is full, write to L2
update value x to y in page p of array z

Level 1

Level 2

page to update

update

cost

what if >1 updates
(no locking for now)
buffer $\ll 1$ updates to this page before pushing to L2

update

page to update
to structure or not to structure

insert v, delete v, update v to v’

- no order
  fixed-width & dense

- sorted
  fixed-width & dense

- sorted
  fixed-width with holes
row-store

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

column-store

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**update** row7=(A=a,B=b,C=c,D=d)

costs

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

A.inserts (id,value)
select(A,v1,v2)

- Scan A
- A.deletes (id)
- A.inserts (id,value)
- Compute diff
- Scan A
- A.del
- A.ins
- Unite ins and diff
- Result res
in-place updates: the cardinal sin

The Transaction Concept: Virtues and Limitations
Jim Gray, Tandem TR 81.3, 1981
update all rows
where A=v1 & B=v2
to (a=a/2, b=b/4, c=c-3, d=d+2)

how to perform updates efficiently and correctly?
_correctly=all or nothing

problems to worry about (?):
what if user/applications aborts?
what if power goes down?
what if there is an earthquake in our city?
what if aliens come to earth?

(assume simplified memory hierarchy)
all data fit in L2, not all data fit in L1
L2 is non-volatile, L1 is volatile
update all rows
where $A=v_1$ & $B=v_2$
to $(a=a/2, b=b/4, c=c-3, d=d+2)$

search (scan/index)
to find row to update

select+project actions
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)
update all rows where \( A = v_1 \) & \( B = v_2 \) to \( (a = a/2, b = b/4, c = c - 3, d = d + 2) \)

we know what to update but nothing happened yet
CPU

Level 1

Level 2

A B C D

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

read page in L1
update
persist to L2

if problem (power/abort)
before we write all pages
we are left with an inconsistent state
when is our the log or an update persistent?

disk persistent memory, e.g., disk?
when is our the log or an update persistent?

persistent memory, e.g., disk?

replicate to multiple machines?

machine 1
machine 2
machine 3
when is our the log or an update persistent?

**disk**

persistent memory, e.g., disk?

machine 1  machine 2  machine 3

city 1  city 2  city 3

replicate to multiple machines?

replicate to multiple machines
>1 clusters in >1 cities?
next class: transactions, ACID

what if >>1 update queries
at the same time

WAL & replication
Aurora: a new model and architecture for data stream management
Daniel J. Abadi, Donald Carney, Ugur Çetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, Stanley B. Zdonik
Very Large Databases Journal (VLDBJ), 2003

Enhanced stream processing in a DBMS kernel
Erietta Liarou, Stratos Idreos, Stefan Manegold, Martin Kersten
In Proc. of the International Conf. on Extending Database Technology (EDBT), 2013
Updating a cracked database
Stratos Idreos, Martin Kersten, Stefan Manegold

Positional update handling in column stores
Sándor Héman, Marcin Zukowski, Niels J. Nes, Lefteris Sidirourgos, Peter A. Boncz
In Proc. of the ACM SIGMOD Int. Conference on Management of Data, 2010

textbook: chapters 16, 17, 18

(also for next class)
updates

DATA SYSTEMS

prof. Stratos Idreos