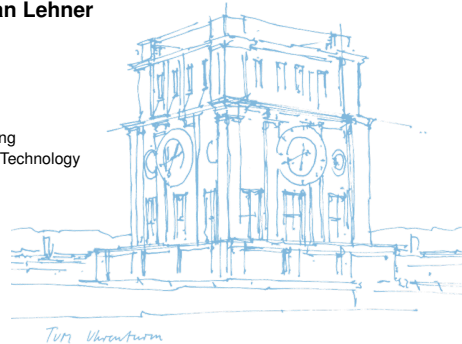


Database Systems on Modern CPU Architectures

Adrian Riedl, Felix Rinderer, Stefan Lehner

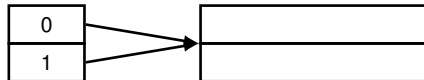
Tuesday 22nd July, 2025

Chair of Data Science and Engineering
TUM School of Computation, Information and Technology
Technical University of Munich



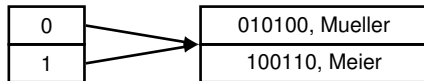
Insert the following tuples into an extendible hash table whose buckets can hold up to two entries.

ID	Name	hash
10	Müller	010100
25	Meier	100110
30	Schmidt	011110
18	Krause	010010
40	Schulz	000101
45	Kaufmann	101101



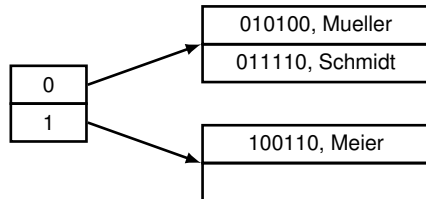
Insert the following tuples into an extendible hash table whose buckets can hold up to two entries.

ID	Name	hash
10	Müller	010100
25	Meier	100110
30	Schmidt	011110
18	Krause	010010
40	Schulz	000101
45	Kaufmann	101101



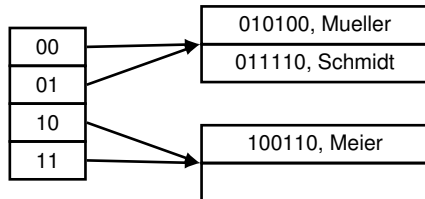
Insert the following tuples into an extendible hash table whose buckets can hold up to two entries.

ID	Name	hash
10	Müller	010100
25	Meier	100110
30	Schmidt	011110
18	Krause	010010
40	Schulz	000101
45	Kaufmann	101101



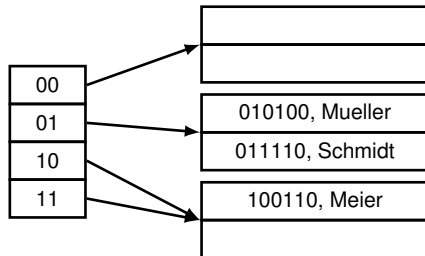
Insert the following tuples into an extendible hash table whose buckets can hold up to two entries.

ID	Name	hash
10	Müller	010100
25	Meier	100110
30	Schmidt	011110
18	Krause	010010
40	Schulz	000101
45	Kaufmann	101101



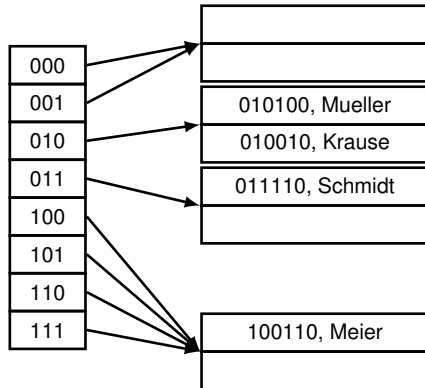
Insert the following tuples into an extendible hash table whose buckets can hold up to two entries.

ID	Name	hash
10	Müller	010100
25	Meier	100110
30	Schmidt	011110
18	Krause	010010
40	Schulz	000101
45	Kaufmann	101101



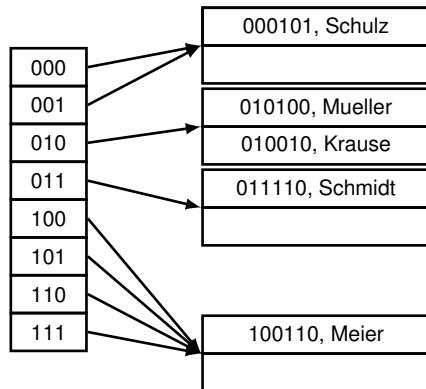
Insert the following tuples into an extendible hash table whose buckets can hold up to two entries.

ID	Name	hash
10	Müller	010100
25	Meier	100110
30	Schmidt	011110
18	Krause	010010
40	Schulz	000101
45	Kaufmann	101101



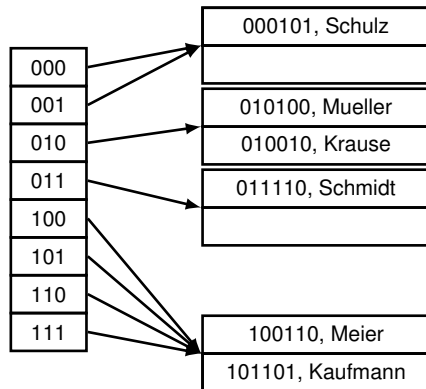
Insert the following tuples into an extendible hash table whose buckets can hold up to two entries.

ID	Name	hash
10	Müller	010100
25	Meier	100110
30	Schmidt	011110
18	Krause	010010
40	Schulz	000101
45	Kaufmann	101101

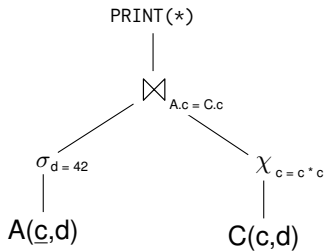


Insert the following tuples into an extendible hash table whose buckets can hold up to two entries.

ID	Name	hash
10	Müller	010100
25	Meier	100110
30	Schmidt	011110
18	Krause	010010
40	Schulz	000101
45	Kaufmann	101101



Generate (pseudo-)code for the following operator tree using the push model.



Draw the operator tree for the given code which was generated by the push model.

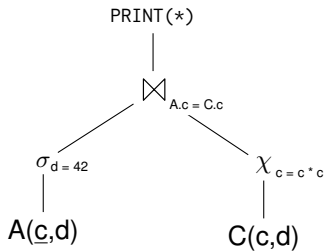
```
hs = HashSet()
ht = HashTable()

for r in R:
    hs.insert(r.a)

for s in S:
    if S.y > 42:
        if hs.contains(s.a):
            if not ht.contains(s.x):
                ht[s.x] = 1
            else
                ht[s.x] += 1

for e : ht:
    print(...)
```

Parallelize the given query plan by introducing `XchgHashSplit(n:m)` and `Xchg(n:m)` operators for **2 threads** as necessary.



Questions?