ACM Programming Contest 2015
Vika team

Nhung Ngo
ngo@inf.unibz.it
Free University of Bozen-Bolzano

Thai Son Nguyen
thaison.ngn@gmail.com
Karlsruhe Institute of Technology

TASK DESCRIPTION

Simulate the task of concurrency verification in DBMS
The system processes already executed transactions and only needs to efficiently check whether concurrent queries conflict with them.

- ❑ A list of insert and delete statements (*transactions*) is provided
- ❑ We need to figure out whether given predicates (*queries*) match the inserted or deleted data.

OUR MISSION

To build a program that:

- ❑ Can handle a massive amount of transactions & queries
- ❑ Run as fast as possible

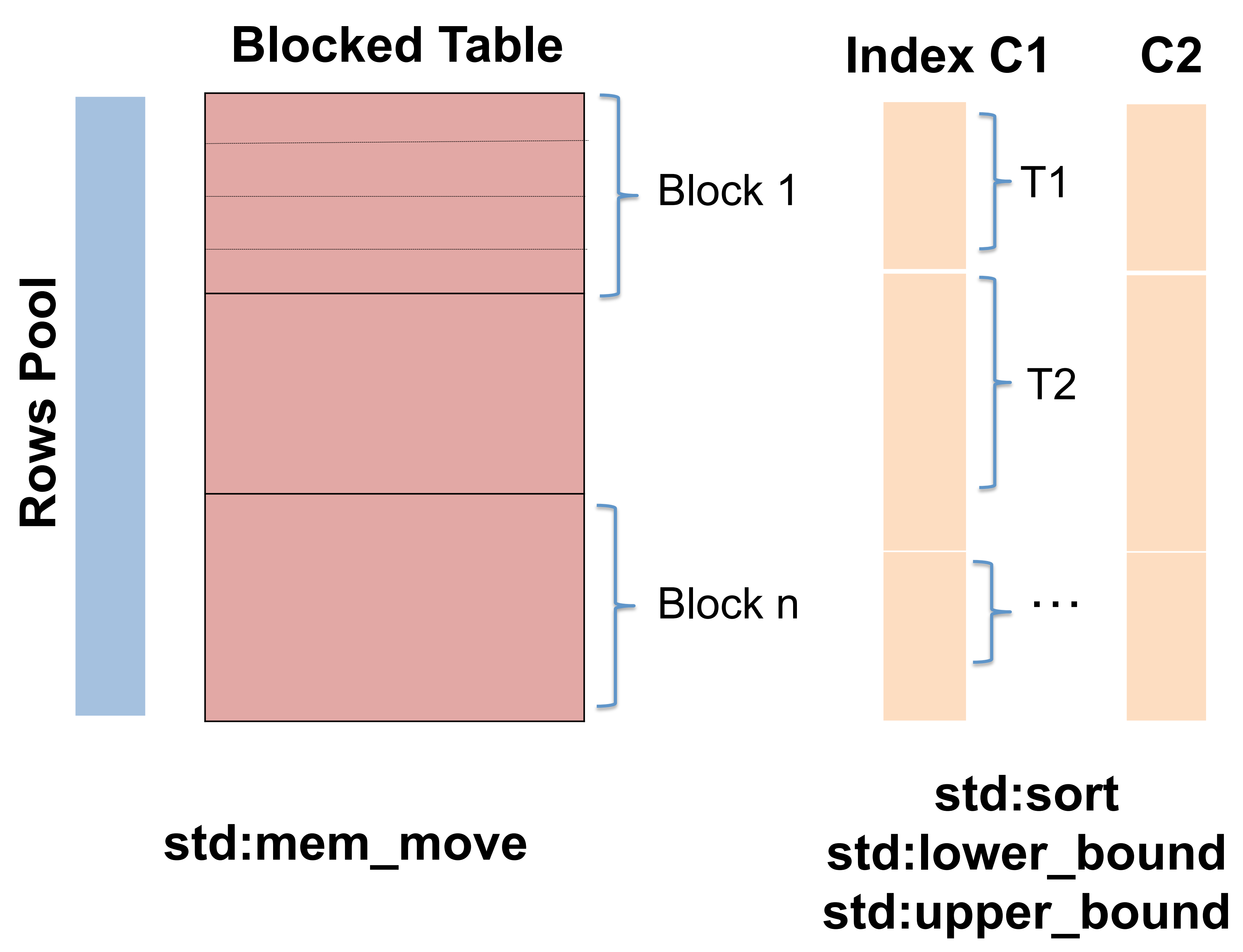
IMPLEMENTATION

- ❑ **Step 1**: Read and put transactions and queries into queues
- ❑ **Step 2**: Concurrently process transactions
- ❑ **Step 3:** Concurrently build indexes
- ❑ **Step 4**: Concurrently validate queries and corresponding transactions
- ❑ **Step 5:** Write results

OUR APPROACH

- ❑ Minimize memory allocations / de-allocations (avoid memory leaks)
- ❑ Use indexes as much as possible
- ❑ Parallelize every step

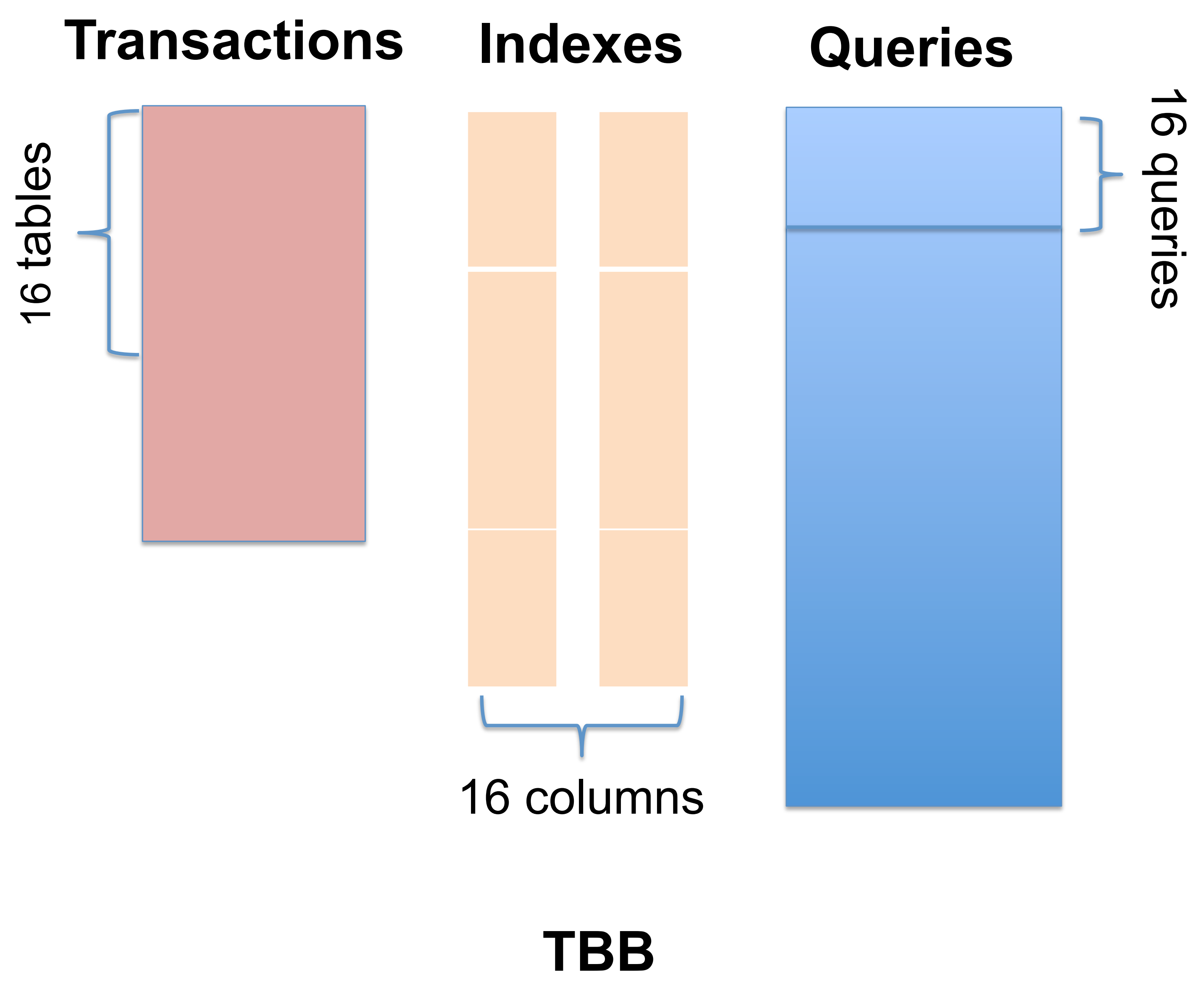
Data Structures


Blocked Table
Index C1
C2
Rows Pool
Block 1
Block n
T1
T2
. . .

std:mem_move

std:sort
std:lower_bound
std:upper_bound

Execution & Parallelism


Transactions
Indexes
Queries
16 tables
16 queries
16 columns

TBB

IMPROVEMENTS

- ❑ Use **max-min index** for low-cardinality columns
- ❑ Index tuples that are belonged to a **range of small transactions** instead of indexing tuples of each such transaction
- ❑ Try to find a good condition which **leads to a very small number of matched tuples** instead of combining query conditions using set intersection.

STATISTICS

1. **Language: C++/make/std lib**
2. **Lines of code: 771**
3. **Time spent: 1 month**
4. **Third party: Intel® Threading Building Blocks (TBB) for multiple threading.**

Fakultät für Informatik
Facoltà di Scienze e Tecnologie informatiche
Faculty of Computer Science