Prof. Dr. T. Grust, Benjamin Dietrich, Denis Hirn SS 2020

DB2

Forum: https://forum-db.informatik.uni-tuebingen.de/c/ss20-db2

Assignment 6 (09.06.2020)

Submission: Tuesday, 16.06.2020, 10:00 AM

Relevant videos: up to `DB2 - Chapter 08 - Video #32`.

https://tinyurl.com/DB2-2020

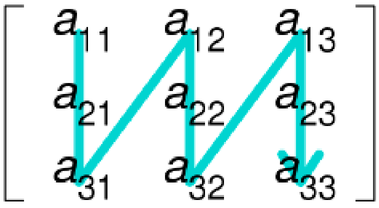
1. [15 Points] **Loop Swapping Optimization**

Processing a wide table stored as C array, the order of processing can have significant impact on performance. Consider the following table `tbl` of random integers stored *row-wise* in an C array:

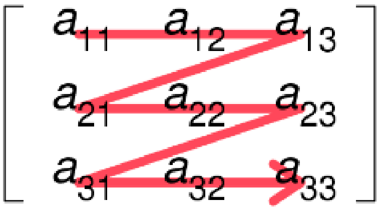
```
unsigned int *tbl;

/* initialize table row-by-row with random values */
for (r = 0; r < rows; r++)
    for (c = 0; c < cols; c++)
        tbl[r * cols + c] = (unsigned int) random();
```

If we want to add up all elements of the table, we can process the elements following two different traversal strategies:

Column-major Order


a_{11} a_{12} a_{13}
a_{21} a_{22} a_{23}
a_{31} a_{32} a_{33}

Row-major Order


a_{11} a_{12} a_{13}
a_{21} a_{22} a_{23}
a_{31} a_{32} a_{33}

The choice of loop order has significant impact on performance:

(a) Extend the program in `loop-swapping.c` to compute and print the overall sum of all cells in table `tbl` in two different ways: Using two nested loops processing array `tbl`

   i. in *row-major* order and

   ii. in *column-major* order.

(b) Further extend your program to measure and print the execution time of both variants.

(c) Compile the program with flag `-O2` and run a test with 1,000,000 rows and 100 columns. Describe the results and explain the difference in performance of both variants.

2. [15 Points] **Logical Conjunction**

In the lectures we learned how a MAL program would apply *logical disjunction* found in SQL queries with operator `OR` (Chapter 8 on Slide 9 and file `predicate-evaluation-monetdb.txt`). Your task is to implement the following SQL query, which makes use of the *logical conjunction* operator `AND`, as a MAL program:

```
SELECT t.a, t.b
FROM   ternary AS t
WHERE  t.a % 2 = 0 AND t.c < 3;
--         ↑               ↑
--         p1              p2
```

The definition of table `ternary` with 10^7 rows is given in `/assignments/assignment06/ternary.sql`.

In `/assignments/assignment06/conjuction.mal` you can find an incomplete MAL program to start with. Write two alternative implementations of the query above:

(a) *Alternative 1*: Apply predicate $p_1 =$ `t.a % 2` first and filter the remaining BAT elements by predicate $p_2 =$ `t.c < 3` afterwards.

(b) *Alternative 2*: Apply predicate p_2 first and filter the remaining BAT elements by predicate p_1 afterwards.

(c) Use `mclient` with options `-l msql` and `-t clock` to run both programs on the given table. Compare the execution time of *Alternative 1* and *Alternative 2* and explain any significant difference.

Notes:

- Use `io.print(...)` to align an print the final result columns.
- Try to keep your MAL code comprehensible. Use comments (`#...`) and choose descriptive variable names.