Database HW3 report

2.1.1

Reference : https://segmentfault.com/a/1190000008131735

我的 7.sql

```
mysql> source 7.sql
ERROR:
Failed to open file '7.sql', error: 2
mysql> source brian/7.sql
+-------------+---------------+---------+
| position    | champion_name | max_kda |
+-------------+---------------+---------+
| DUO_CARRY   | Shaco         | 19.0000 |
| DUO_SUPPORT | Janna         |  3.8330 |
| JUNGLE      | Ivern         |  3.8764 |
| MID         | Ivern         |  3.7015 |
| TOP         | Sona          |  3.1538 |
+-------------+---------------+---------+
5 rows in set, 104 warnings (23.31 sec)

mysql> source brian/7.sql
+----+-------------+------------+------------+--------+---------------+------------+---------+-------------------+---------+----------+---------------------------------------------+
| id | select_type | table      | partitions | type   | possible_keys | key        | key_len | ref               | rows    | filtered | Extra                                       |
+----+-------------+------------+------------+--------+---------------+------------+---------+-------------------+---------+----------+---------------------------------------------+
|  1 | PRIMARY     | <derived4> | NULL       | ALL    | NULL          | NULL       | NULL    | NULL              | 1817575 |   100.00 | Using where; Using temporary; Using files   |
|  1 | PRIMARY     | C          | NULL       | eq_ref | PRIMARY       | PRIMARY    | 4       | new3_t.champion_id |       1 |   100.00 | NULL                                        |
|  1 | PRIMARY     | <derived2> | NULL       | ref    | <auto_key0>   | <auto_key0> | 15      | new3_t.kda        |      10 |   100.00 | NULL                                        |
|  4 | DERIVED     | P          | NULL       | ALL    | PRIMARY       | NULL       | NULL    | NULL              | 1817575 |   100.00 | Using temporary; Using filesort             |
|  4 | DERIVED     | S          | NULL       | eq_ref | PRIMARY       | PRIMARY    | 4       | hw1.P.player_id   |       1 |   100.00 | NULL                                        |
|  2 | DERIVED     | <derived3> | NULL       | ALL    | NULL          | NULL       | NULL    | NULL              | 1817575 |    72.90 | Using where; Using temporary; Using files   |
|  3 | DERIVED     | P          | NULL       | ALL    | PRIMARY       | NULL       | NULL    | NULL              | 1817575 |   100.00 | Using temporary; Using filesort             |
|  3 | DERIVED     | S          | NULL       | eq_ref | PRIMARY       | PRIMARY    | 4       | hw1.P.player_id   |       1 |   100.00 | NULL                                        |
+----+-------------+------------+------------+--------+---------------+------------+---------+-------------------+---------+----------+---------------------------------------------+
8 rows in set, 1 warning (0.00 sec)
```

Karl 7.sql

```
mysql> source karl/7.sql
+-------------+---------------+---------+
| position    | champion_name | kda     |
+-------------+---------------+---------+
| DUO_CARRY   | Shaco         | 19.0000 |
| DUO_SUPPORT | Janna         |  3.8330 |
| JUNGLE      | Ivern         |  3.8764 |
| MID         | Ivern         |  3.7015 |
| TOP         | Sona          |  3.1538 |
+-------------+---------------+---------+
5 rows in set (26.36 sec)

mysql> source karl/7.sql
+----+-------------+------------+------------+--------+---------------+------------+---------+-------------------------+---------+----------+-----------------------------------------------+
| id | select_type | table      | partitions | type   | possible_keys | key        | key_len | ref                     | rows    | filtered | Extra                                         |
+----+-------------+------------+------------+--------+---------------+------------+---------+-------------------------+---------+----------+-----------------------------------------------+
|  1 | PRIMARY     | <derived2> | NULL       | ALL    | NULL          | NULL       | NULL    | NULL                    |  744315 |   100.00 | Using where; Using filesort                   |
|  1 | PRIMARY     | <derived5> | NULL       | ref    | <auto_key0>   | <auto_key0> | 30      | grp.position,grp.sum_kda |      10 |   100.00 | Using index                                   |
|  5 | DERIVED     | <derived7> | NULL       | ALL    | NULL          | NULL       | NULL    | NULL                    |  744315 |   100.00 | Using temporary; Using filesort               |
|  7 | DERIVED     | par        | NULL       | ALL    | PRIMARY       | NULL       | NULL    | NULL                    | 1817575 |    40.95 | Using where; Using temporary; Using           |
|  7 | DERIVED     | champ      | NULL       | eq_ref | PRIMARY       | PRIMARY    | 4       | hw1.par.champion_id     |       1 |   100.00 | NULL                                          |
|  7 | DERIVED     | stat       | NULL       | eq_ref | PRIMARY       | PRIMARY    | 4       | hw1.par.player_id       |       1 |   100.00 | NULL                                          |
|  2 | DERIVED     | par        | NULL       | ALL    | PRIMARY       | NULL       | NULL    | NULL                    | 1817575 |    40.95 | Using where; Using temporary; Using           |
|  2 | DERIVED     | champ      | NULL       | eq_ref | PRIMARY       | PRIMARY    | 4       | hw1.par.champion_id     |       1 |   100.00 | NULL                                          |
|  2 | DERIVED     | stat       | NULL       | eq_ref | PRIMARY       | PRIMARY    | 4       | hw1.par.player_id       |       1 |   100.00 | NULL                                          |
+----+-------------+------------+------------+--------+---------------+------------+---------+-------------------------+---------+----------+-----------------------------------------------+
9 rows in set, 1 warning (0.00 sec)
```

Yao 7.sql

```
mysql> source yao/7.sql
+-------------+---------------+---------+
| position    | champion_name | kda     |
+-------------+---------------+---------+
| DUO_CARRY   | Shaco         | 19.0000 |
| DUO_SUPPORT | Janna         |  3.8330 |
| JUNGLE      | Ivern         |  3.8764 |
| MID         | Ivern         |  3.7015 |
| TOP         | Sona          |  3.1538 |
+-------------+---------------+---------+
5 rows in set (11.50 sec)

mysql> source yao/7.sql
+----+-------------+------------+------------+--------+---------------+------------+---------+-------------------+---------+----------+-----------------------------------------------+
| id | select_type | table      | partitions | type   | possible_keys | key        | key_len | ref               | rows    | filtered | Extra                                         |
+----+-------------+------------+------------+--------+---------------+------------+---------+-------------------+---------+----------+-----------------------------------------------+
|  1 | PRIMARY     | <derived2> | NULL       | ALL    | NULL          | NULL       | NULL    | NULL              | 1192510 |    90.00 | Using where; Using temporary; Using filesort  |
|  1 | PRIMARY     | c          | NULL       | eq_ref | PRIMARY       | PRIMARY    | 4       | info.champion_id  |       1 |   100.00 | NULL                                          |
|  2 | DERIVED     | p          | NULL       | ALL    | PRIMARY       | NULL       | NULL    | NULL              | 1817575 |    65.61 | Using where; Using temporary; Using filesort  |
|  2 | DERIVED     | s          | NULL       | eq_ref | PRIMARY       | PRIMARY    | 4       | hw1.p.player_id   |       1 |   100.00 | NULL                                          |
+----+-------------+------------+------------+--------+---------------+------------+---------+-------------------+---------+----------+-----------------------------------------------+
4 rows in set, 1 warning (0.00 sec)
```

我的跟 karl 助教的 explain 差異不大 但我只用 8 個 select 而 karl 助教用 9 select。

Yao 助教則是只用了 4 個 select 完成。

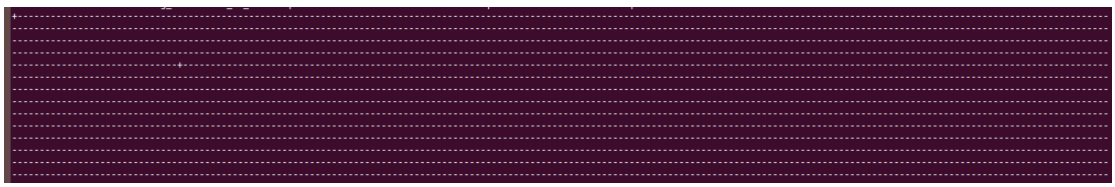
Type 性能比較:ALL < index < range ~ index_merge < ref < eq_ref < const < system
從上面可以知道，用越多 ALL type 會越慢。
並且在 extra 用到 Using temporary,using filesort 也很耗時
還有如果 filter 比例越低 這個 select 規模越小越快。
以上決定了 yao 的 query 效率較高。

2.1.2

我的 optimizer_trace:
首先先看時間

```
mysql> source brian/6.sql
+-------------+---------------+
| position    | champion_name |
+-------------+---------------+
| DUO_CARRY   | Caitlyn       |
| DUO_SUPPORT | Thresh        |
| JUNGLE      | Lee Sin       |
| MID         | Ahri          |
| TOP         | Riven         |
+-------------+---------------+
5 rows in set (15.45 sec)
```

然後把 optimizier_trace 的 json 打開發現有點長，所以複製到 mac OS 存成 json 檔再用 VScode 開啟，然後因為要符合直立式螢幕，所以有些行我手動縮排。



無意義的行數我先不複製
這些都弄好後，json 如下

```
select T1.position,C.champion_name
from(
    select t1.position,max(t1.cnt) as max_cnt
    from(select P.position,P.champion_id,count(*) cnt
        from participant as P, match_info as M
        where M.duration between 2400 and 3000 and P.match_id=M.match_id
        group by P.position ,P.champion_id
        order by cnt desc
    )as t1
    group by t1.position
)as T1,
(
    select P.position,P.champion_id,count(*) cnt
    from participant as P, match_info as M
    where M.duration between 2400 and 3000 and P.match_id=M.match_id
    group by P.position ,P.champion_id
    order by cnt desc
)as T2,
champ as C
where T1.position=T2.position and T1.max_cnt=T2.cnt and C.champion_id=T2.champion_id and T1.position != "DUO"
        and T1.position != "SOLO" and T1.position != "NONE"
order by T1.position | {
  "steps": [
    {
      "join_preparation": {
        "select#": 1,
        "steps": [
          {
            "join_preparation": {
              "select#": 2,
              "steps": [
                {
                  "join_preparation": {
                    "select#": 3,
                    "steps": [
                      {
                        "expanded_query":
                        "/* select#3 */
                        select `P`.`position` AS `position`,
                        `P`.`champion_id` AS `champion_id`,count(0) AS `cnt`
                        from `participant` `P` join `match_info` `M`
                        where ((`M`.`duration` between 2400 and 3000)
                            and (`P`.`match_id` = `M`.`match_id`))
                        group by `P`.`position`,`P`.`champion_id` order by `cnt` desc"
                      }
                    ]
                  }
                },
                {
                  "derived": {
                    "table": " `t1`",
                    "select#": 3,
                    "materialized": true
                  }
                },
                {
                  "expanded_query": "/* select#2 */
                    select `t1`.`position` AS `position`,max(`t1`.`cnt`) AS `max_cnt`
                    from (
                      /* select#3 */
                        select `P`.`position` AS `position`,`P`.`champion_id` AS `champion_id`,count(0) AS `cnt`
                        from `participant` `P` join `match_info` `M`
                        where ((`M`.`duration` between 2400 and 3000) and (`P`.`match_id` = `M`.`match_id`))
                        group by `P`.`position`,`P`.`champion_id` order by `cnt` desc) `t1`
                        group by `t1`.`position`"
                }
              ]
            }
          },
          {
            "join_preparation": {
              "select#": 4,
              "steps": [
                {
                  "expanded_query": "/* select#4 */
                    select `P`.`position` AS `position`,`P`.`champion_id` AS `champion_id`,count(0) AS `cnt`
                    from `participant` `P` join `match_info` `M`
                    where ((`M`.`duration` between 2400 and 3000)
                      and (`P`.`match_id` = `M`.`match_id`)) group by `P`.`position`,`P`.`champion_id`
                    order by `cnt` desc"
                }
              ]
            }
          },
          {
            "derived": {
              "table": " `T1`",
              "select#": 2,
              "materialized": true
            }
          },
          {
            "derived": {
              "table": " `T2`",
              "select#": 4,
              "materialized": true
            }
          },
          {
            "expanded_query":
```

```
            "expanded_query": "/* select#1 */
              select `T1`.`position` AS `position`,`C`.`champion_name` AS `champion_name`
              from (
                /* select#2 */
                select `t1`.`position` AS `position`,max(`t1`.`cnt`) AS `max_cnt`
                from (
                  /* select#3 */
                  select `P`.`position` AS `position`,`P`.`champion_id` AS `champion_id`,count(0) AS `cnt`
                  from `participant` `P` join `match_info` `M`
                  where ((`M`.`duration` between 2400 and 3000)
                        and (`P`.`match_id` = `M`.`match_id`))
                  group by `P`.`position`,`P`.`champion_id`
                  order by `cnt` desc) `t1`
                group by `t1`.`position`) `T1`

                join (
                /* select#4 */
                  select `P`.`position` AS `position`,`P`.`champion_id` AS `champion_id`,count(0) AS `cnt`
                  from `participant` `P` join `match_info` `M`
                  where ((`M`.`duration` between 2400 and 3000) and (`P`.`match_id` = `M`.`match_id`))
                  group by `P`.`position`,`P`.`champion_id` order by `cnt` desc) `T2` join `champ` `C`
                where ((`T1`.`position` = `T2`.`position`)
                    and (`T1`.`max_cnt` = `T2`.`cnt`)
                    and (`C`.`champion_id` = `T2`.`champion_id`)
                    and (`T1`.`position` <> 'DUO')
                    and (`T1`.`position` <> 'SOLO')
                    and (`T1`.`position` <> 'NONE'))
                order by `T1`.`position`"
            }
          ]
        }
      },
      {
        "join_optimization": {
          "select#": 1,
          "steps": [
            {
              "join_optimization": {
                "select#": 2,
                "steps": [
                  {
                    "join_optimization": {
                      "select#": 3,
                      "steps": [
                        {
                          "condition_processing": {
                            "condition": "WHERE",
                            "original_condition": "((`M`.`duration` between 2400 and 3000)
                                                  and (`P`.`match_id` = `M`.`match_id`))",
                            "steps": [
                              {
                                "transformation": "equality_propagation",
                                "resulting_condition": "((`M`.`duration` between 2400 and 3000)
                                                       and multiple equal(`P`.`match_id`, `M`.`match_id`))"
                              },
                              {
                                "transformation": "constant_propagation",
                                "resulting_condition": "((`M`.`duration` between 2400 and 3000)
                                                       and multiple equal(`P`.`match_id`, `M`.`match_id`))"
                              },
                              {
                                "transformation": "trivial_condition_removal",
                                "resulting_condition": "((`M`.`duration` between 2400 and 3000)
                                                       and multiple equal(`P`.`match_id`, `M`.`match_id`))"
                              }
                            ]
                          }
                        },
                        {
                          "substitute_generated_columns": {
                          }
                        },
                        {
                          "table_dependencies": [
                            {
                              "table": "`participant` `P`",
                              "row_may_be_null": false,
                              "map_bit": 0,
                              "depends_on_map_bits": [
                              ]
                            },
                            {
                              "table": "`match_info` `M`",
                              "row_may_be_null": false,
                              "map_bit": 1,
                              "depends_on_map_bits": [
                              ]
                            }
                          ]
                        },
                        {
                          "ref_optimizer_key_uses": [
                            {
                              "table": "`participant` `P`",
                              "field": "match_id",
                              "equals": "`M`.`match_id`",
                              "null_rejecting": false
                            },
                            {
                              "table": "`match_info` `M`",
```

```
            "field": "match_id",
            "equals": "`P`.`match_id`",
            "null_rejecting": false
          }
        ]
      },
      {
        "rows_estimation": [
          {
            "table": "`participant` `P`",
            "table_scan": {
              "rows": 1817575,
              "cost": 6376
            }
          },
          {
            "table": "`match_info` `M`",
            "table_scan": {
              "rows": 182186,
              "cost": 545
            }
          }
        ]
      },
      {
        "considered_execution_plans": [
          {
            "plan_prefix": [
            ],
            "table": "`match_info` `M`",
            "best_access_path": {
              "considered_access_paths": [
                {
                  "access_type": "ref",
                  "index": "PRIMARY",
                  "usable": false,
                  "chosen": false
                },
                {
                  "rows_to_scan": 182186,
                  "access_type": "scan",
                  "resulting_rows": 20241,
                  "cost": 36982,
                  "chosen": true
                }
              ]
            },
            "condition_filtering_pct": 100,
            "rows_for_plan": 20241,
            "cost_for_plan": 36982,
            "rest_of_plan": [
              {
                "plan_prefix": [
                  "`match_info` `M`"
                ],
                "table": "`participant` `P`",
                "best_access_path": {
                  "considered_access_paths": [
                    {
                      "access_type": "ref",
                      "index": "match_id",
                      "rows": 9.953,
                      "cost": 241750,
                      "chosen": true
                    },
                    {
                      "rows_to_scan": 1817575,
                      "access_type": "scan",
                      "using_join_cache": true,
                      "buffers_needed": 1,
                      "resulting_rows": 1.82e6,
                      "cost": 7.36e9,
                      "chosen": false
                    }
                  ]
                },
                "condition_filtering_pct": 100,
                "rows_for_plan": 201458,
                "cost_for_plan": 278732,
                "chosen": true
              }
            ]
          },
          {
            "plan_prefix": [
            ],
            "table": "`participant` `P`",
            "best_access_path": {
              "considered_access_paths": [
                {
                  "access_type": "ref",
                  "index": "match_id",
                  "usable": false,
                  "chosen": false
                },
                {
                  "rows_to_scan": 1817575,
                  "access_type": "scan",
                  "resulting_rows": 1.82e6,
                  "cost": 369891,
```

```
                    "chosen": true
                   }
                  ]
                },
                "condition_filtering_pct": 100,
                "rows_for_plan": 1.82e6,
                "cost_for_plan": 369891,
                "pruned_by_cost": true
               }
              ]
             },
             {
              "attaching_conditions_to_tables": {
               "original_condition": "((`P`.`match_id` = `M`.`match_id`)
                                      and (`M`.`duration` between 2400 and 3000))",
               "attached_conditions_computation": [
               ],
               "attached_conditions_summary": [
                {
                 "table": "`match_info` `M`",
                 "attached": "(`M`.`duration` between 2400 and 3000)"
                },
                {
                 "table": "`participant` `P`",
                 "attached": null
                }
               ]
              }
             },
             {
              "clause_processing": {
               "clause": "ORDER BY",
               "original_clause": "`cnt` desc",
               "items": [
                {
                 "item": "count(0)"
                }
               ],
               "resulting_clause_is_simple": false,
               "resulting_clause": "`cnt` desc"
              }
             },
             {
              "clause_processing": {
               "clause": "GROUP BY",
               "original_clause": "`P`.`position`,`P`.`champion_id`",
               "items": [
                {
                 "item": "`P`.`position`"
                },
                {
                 "item": "`P`.`champion_id`"
                }
               ],
               "resulting_clause_is_simple": false,
               "resulting_clause": "`P`.`position`,`P`.`champion_id`"
              }
             },
             {
              "refine_plan": [
               {
                "table": "`match_info` `M`"
               },
               {
                "table": "`participant` `P`"
               }
              ]
             }
            ]
           }
          },
          {
           "substitute_generated_columns": {
           }
          },
          {
           "table_dependencies": [
            {
             "table": " `t1`",
             "row_may_be_null": false,
             "map_bit": 0,
             "depends_on_map_bits": [
             ]
            }
           ]
          },
          {
           "rows_estimation": [
            {
             "table": " `t1`",
             "table_scan": {
              "rows": 201458,
              "cost": 10082
             }
            }
           ]
          },
          {
           "considered_execution_plans": [
            {
```

可以看到一個 select 下面還有 subselect，並且 select #4,5,8,9 各重複兩次，這就是造成時間為 17sec 的原因

karl:
助教的 json 看起來很短
先來看看執行時間

```
mysql> source karl/6.sql
+--------------+---------------+
| position     | champion_name |
+--------------+---------------+
| DUO_CARRY    | Caitlyn       |
| DUO_SUPPORT  | Thresh        |
| JUNGLE       | Lee Sin       |
| MID          | Ahri          |
| TOP          | Riven         |
+--------------+---------------+
5 rows in set (15.46 sec)
```

Json:

```
explain
select final.position, champ.champion_name
from
(
    select list.position, list.champion_id
    from
    (
        select position, champion_id, count(champion_id) as cnt
        from
        (
            select match_id
            from match_info
            where duration >= 2400 and duration <= 3000
        ) as dur_match,
        (
            select match_id, champion_id, position
            from participant
            where position = 'TOP' or position = 'MID' or position = 'JUNGLE'
            or position = 'DUO_CARRY' or position = 'DUO_SUPPORT'
        ) as par
        where par.match_id = dur_match.match_id
        group by position, champion_id
        order by cnt desc
    ) as list,
    (
        select grp_pos_champ.position, max(grp_pos_champ.cnt) as cnt
        from
        (
            select position, champion_id, count(champion_id) as cnt
            from
            (
                select match_id
                from match_info
                where duration >= 2400 and duration <= 3000
            ) as dur_match,
            (
                select match_id, champion_id, position
                from participant
                where position = 'TOP' or position = 'MID' or position = 'JUNGLE'
                or position = 'DUO_CARRY' or position = 'DUO_SUPPORT'
            ) as par
            where par.match_id = dur_match.match_id
            group by position, champion_id
            order by cnt desc
        ) as grp_pos_champ
        group by grp_pos_champ.position
    ) as tmp
    where list.position = tmp.position
    and list.cnt = tmp.cnt
) as final, champ
where final.champion_id = champ.champion_id
order by final.position | {
  "steps": [
    {
      "join_preparation": {
        "select#": 1,
        "steps": [
          {
            "join_preparation": {
              "select#": 2,
              "steps": [
                {
                  "join_preparation": {
                    "select#": 3,
                    "steps": [
                      {
                        "join_preparation": {
                          "select#": 4,
                          "steps": [
                            {
                              "expanded_query": "/* select#4 */ select `match_info`.`match_id` AS `match_id` from `
                            }
                          ]
                        }
                      },
                      {
                        "join_preparation": {
                          "select#": 5,
                          "steps": [
                            {
                              "expanded_query": "/* select#5 */ select `participant`.`match_id` AS `match_id`,`par
                            }
                          ]
                        }
                      },
                      {
                        "derived": {
                          "table": "``.`` `dur_match`",
                          "select#": 4,
                          "merged": true
                        }
                      },
                      {
                        "derived": {
                          "table": "``.`` `par`",
                          "select#": 5,
                          "merged": true
                        }
                      },
                      {
```

```
                    "expanded_query": "/* select#3 */ select `participant`.`position` AS `position`,`participan
                  }
                ]
              }
            },
            {
              "join_preparation": {
                "select#": 6,
                "steps": [
                  {
                    "join_preparation": {
                      "select#": 7,
                      "steps": [
                        {
                          "join_preparation": {
                            "select#": 8,
                            "steps": [
                              {
                                "expanded_query": "/* select#8 */ select `match_info`.`match_id` AS `match_id`
                              }
                            ]
                          }
                        },
                        {
                          "join_preparation": {
                            "select#": 9,
                            "steps": [
                              {
                                "expanded_query": "/* select#9 */ select `participant`.`match_id` AS `match_id`
                              }
                            ]
                          }
                        },
                        {
                          "derived": {
                            "table": "``.`` `dur_match`",
                            "select#": 8,
                            "merged": true
                          }
                        },
                        {
                          "derived": {
                            "table": "``.`` `par`",
                            "select#": 9,
                            "merged": true
                          }
                        },
                        {
                          "expanded_query": "/* select#7 */ select `participant`.`position` AS `position`,`part
                        }
                      ]
                    }
                  },
                  {
                    "derived": {
                      "table": " `grp_pos_champ`",
                      "select#": 7,
                      "materialized": true
                    }
                  },
                  {
                    "expanded_query": "/* select#6 */ select `grp_pos_champ`.`position` AS `position`,max(`grp_
                  }
                ]
              }
            },
            {
              "derived": {
                "table": " `list`",
                "select#": 3,
                "materialized": true
              }
            },
            {
              "derived": {
                "table": " `tmp`",
                "select#": 6,
                "materialized": true
              }
            },
            {
              "expanded_query": "/* select#2 */ select `list`.`position` AS `position`,`list`.`champion_id` AS
            }
          ]
        }
      },
      {
        "derived": {
          "table": "``.`` `final`",
          "select#": 2,
          "merged": true
        }
      },
      {
        "expanded_query": "/* select#1 */ select `list`.`position` AS `position`,`champ`.`champion_name` AS `ch
      },
      {
        "transformations_to_nested_joins": {
          "transformations": [
            "JOIN_condition_to_WHERE",
```

```
            "parenthesis_removal"
          ],
          "expanded_query": "/* select#7 */ select `participant`.`position` AS `position`,`participant`.`champi
        }
      },
      {
        "transformations_to_nested_joins": {
          "transformations": [
            "JOIN_condition_to_WHERE",
            "parenthesis_removal"
          ],
          "expanded_query": "/* select#3 */ select `participant`.`position` AS `position`,`participant`.`champi
        }
      },
      {
        "transformations_to_nested_joins": {
          "transformations": [
            "JOIN_condition_to_WHERE",
            "parenthesis_removal"
          ],
          "expanded_query": "/* select#1 */ select `list`.`position` AS `position`,`champ`.`champion_name` AS
```


可以看到其實 select #4,5,8,9 也都重複兩次 寫法跟我的很像
所以得出來的時間才會差不多

Yao:

```
mysql> source yao/6.sql
+-------------+----------------------------+
| position    | ANY_VALUE(c.champion_name) |
+-------------+----------------------------+
| DUO_CARRY   | Annie                      |
| DUO_SUPPORT | Annie                      |
| JUNGLE      | Annie                      |
| MID         | Annie                      |
| TOP         | Annie                      |
+-------------+----------------------------+
5 rows in set (5.77 sec)
```

時間只需 1/3
但答案在 Ver 14.14 Distrib 5.7.30, for Linux (x86_64) using 是錯的
應該是 any_value()的問題
Query 也不能在 Ver 14.14 Distrib 5.6.21, for osx10.8 (x86_64)環境下跑

```
explain
SELECT info.position, ANY_VALUE(c.champion_name)
FROM
(
    SELECT p.champion_id, p.position, COUNT(*) as appear
    FROM participant as p, match_info as m
    WHERE p.position != 'NONE'
    AND p.position != 'DUO'
    AND p.position != 'BOT'
    AND p.position != 'SOLO'
    AND m.duration BETWEEN 2400 AND 3000
    AND m.match_id = p.match_id
    GROUP BY p.champion_id, p.position
    ORDER BY appear DESC
) as info, champ as c
WHERE c.champion_id = info.champion_id
GROUP BY info.position | {
  "steps": [
    {
      "join_preparation": {
        "select#": 1,
        "steps": [
          {
            "join_preparation": {
              "select#": 2,
              "steps": [
                {
                  "expanded_query": "/* select#2 */ select `p`.`champion_id` AS `champion_id`,`p`.`position` AS `po
                }
              ]
            }
          },
          {
            "derived": {
              "table": " `info`",
              "select#": 2,
              "materialized": true
            }
          },
          {
            "expanded_query": "/* select#1 */ select `info`.`position` AS `position`,any_value(`c`.`champion_name`)
          }
        ]
      }
    },
    {
      "join_optimization": {
        "select#": 1,
        "steps": [
          {
            "join_optimization": {
              "select#": 2,
              "steps": [
                {
                  "condition_processing": {
                    "condition": "WHERE",
                    "original_condition": "((`p`.`position` <> 'NONE') and (`p`.`position` <> 'DUO') and (`p`.`posi
                    "steps": [
                      {
                        "transformation": "equality_propagation",
                        "resulting_condition": "((`p`.`position` <> 'NONE') and (`p`.`position` <> 'DUO') and (`p`.
                      },
                      {
                        "transformation": "constant_propagation",
                        "resulting_condition": "((`p`.`position` <> 'NONE') and (`p`.`position` <> 'DUO') and (`p`.
                      },
                      {
                        "transformation": "trivial_condition_removal",
                        "resulting_condition": "((`p`.`position` <> 'NONE') and (`p`.`position` <> 'DUO') and (`p`.
                      }
                    ]
                  }
                },
                {
                  "substitute_generated_columns": {
                  }
                },
                {
                  "table_dependencies": [
                    {
                      "table": "`participant` `p`",
                      "row_may_be_null": false,
                      "map_bit": 0,
                      "depends_on_map_bits": [
                      ]
                    },
                    {
                      "table": "`match_info` `m`",
                      "row_may_be_null": false,
                      "map_bit": 1,
                      "depends_on_map_bits": [
                      ]
                    }
                  ]
                },
                {
                  "ref_optimizer_key_uses": [
                    {
                      "table": "`participant` `p`",
                      "field": "match_id",
```

```
            "equals": "`m`.`match_id`",
            "null_rejecting": false
          },
          {
            "table": "`match_info` `m`",
            "field": "match_id",
            "equals": "`p`.`match_id`",
            "null_rejecting": false
          }
        ]
      },
      {
        "rows_estimation": [
          {
            "table": "`participant` `p`",
            "table_scan": {
              "rows": 1817575,
              "cost": 6376
            }
          },
          {
            "table": "`match_info` `m`",
            "table_scan": {
              "rows": 182186,
              "cost": 545
            }
          }
        ]
      },
      {
        "considered_execution_plans": [
          {
            "plan_prefix": [
            ],
            "table": "`match_info` `m`",
            "best_access_path": {
              "considered_access_paths": [
                {
                  "access_type": "ref",
                  "index": "PRIMARY",
                  "usable": false,
                  "chosen": false
                },
                {
                  "rows_to_scan": 182186,
                  "access_type": "scan",
                  "resulting_rows": 20241,
                  "cost": 36982,
                  "chosen": true
                }
              ]
            },
            "condition_filtering_pct": 100,
            "rows_for_plan": 20241,
            "cost_for_plan": 36982,
            "rest_of_plan": [
              {
                "plan_prefix": [
                  "`match_info` `m`"
                ],
                "table": "`participant` `p`",
                "best_access_path": {
                  "considered_access_paths": [
                    {
                      "access_type": "ref",
                      "index": "match_id",
                      "rows": 9.953,
                      "cost": 241750,
                      "chosen": true
                    },
                    {
                      "rows_to_scan": 1817575,
                      "access_type": "scan",
                      "using_join_cache": true,
                      "buffers_needed": 1,
                      "resulting_rows": 1.19e6,
                      "cost": 4.83e9,
                      "chosen": false
                    }
                  ]
                },
                "condition_filtering_pct": 65.61,
                "rows_for_plan": 132177,
                "cost_for_plan": 278732,
                "chosen": true
              }
            ]
          },
          {
            "plan_prefix": [
            ],
            "table": "`participant` `p`",
            "best_access_path": {
              "considered_access_paths": [
                {
                  "access_type": "ref",
                  "index": "match_id",
                  "usable": false,
                  "chosen": false
                },
```

```
                    {
                      "rows_to_scan": 1817575,
                      "access_type": "scan",
                      "resulting_rows": 1.19e6,
                      "cost": 369891,
                      "chosen": true
                    }
                  ]
                },
                "condition_filtering_pct": 100,
                "rows_for_plan": 1.19e6,
                "cost_for_plan": 369891,
                "pruned_by_cost": true
              }
            ]
          },
          {
            "attaching_conditions_to_tables": {
              "original_condition": "((`p`.`match_id` = `m`.`match_id`) and (`p`.`position` <> 'NONE')
              "attached_conditions_computation": [
              ],
              "attached_conditions_summary": [
                {
                  "table": "`match_info` `m`",
                  "attached": "(`m`.`duration` between 2400 and 3000)"
                },
                {
                  "table": "`participant` `p`",
                  "attached": "((`p`.`position` <> 'NONE') and (`p`.`position` <> 'DUO') and (`p`.`posi
                }
              ]
            }
          },
          {
            "clause_processing": {
              "clause": "ORDER BY",
              "original_clause": "`appear` desc",
              "items": [
                {
                  "item": "count(0)"
                }
              ],
              "resulting_clause_is_simple": false,
              "resulting_clause": "`appear` desc"
            }
          },
          {
            "clause_processing": {
              "clause": "GROUP BY",
              "original_clause": "`p`.`champion_id`,`p`.`position`",
              "items": [
                {
                  "item": "`p`.`champion_id`"
                },
                {
                  "item": "`p`.`position`"
                }
              ],
              "resulting_clause_is_simple": false,
              "resulting_clause": "`p`.`champion_id`,`p`.`position`"
            }
          },
          {
            "refine_plan": [
              {
                "table": "`match_info` `m`"
              },
              {
                "table": "`participant` `p`"
              }
            ]
          }
        ]
      }
    },
    {
      "condition_processing": {
        "condition": "WHERE",
        "original_condition": "(`c`.`champion_id` = `info`.`champion_id`)",
        "steps": [
          {
            "transformation": "equality_propagation",
            "resulting_condition": "multiple equal(`c`.`champion_id`, `info`.`champion_id`)"
          },
          {
            "transformation": "constant_propagation",
            "resulting_condition": "multiple equal(`c`.`champion_id`, `info`.`champion_id`)"
          },
          {
            "transformation": "trivial_condition_removal",
            "resulting_condition": "multiple equal(`c`.`champion_id`, `info`.`champion_id`)"
          }
        ]
      }
    },
    {
      "substitute_generated_columns": {
      }
    },
    {
```

```
      {
        "table_dependencies": [
          {
            "table": " `info`",
            "row_may_be_null": false,
            "map_bit": 0,
            "depends_on_map_bits": [
            ]
          },
          {
            "table": "`champ` `c`",
            "row_may_be_null": false,
            "map_bit": 1,
            "depends_on_map_bits": [
            ]
          }
        ]
      },
      {
        "ref_optimizer_key_uses": [
          {
            "table": " `info`",
            "field": "champion_id",
            "equals": "`c`.`champion_id`",
            "null_rejecting": false
          },
          {
            "table": " `info`",
            "field": "champion_id",
            "equals": "`c`.`champion_id`",
            "null_rejecting": false
          },
          {
            "table": "`champ` `c`",
            "field": "champion_id",
            "equals": "`info`.`champion_id`",
            "null_rejecting": false
          }
        ]
      },
      {
        "rows_estimation": [
          {
            "table": " `info`",
            "table_scan": {
              "rows": 132176,
              "cost": 6618
            }
          },
          {
            "table": "`champ` `c`",
            "table_scan": {
              "rows": 138,
              "cost": 1
            }
          }
        ]
      },
      {
        "considered_execution_plans": [
          {
            "plan_prefix": [
            ],
            "table": "`champ` `c`",
            "best_access_path": {
              "considered_access_paths": [
                {
                  "access_type": "ref",
                  "index": "PRIMARY",
                  "usable": false,
                  "chosen": false
                },
                {
                  "rows_to_scan": 138,
                  "access_type": "scan",
                  "resulting_rows": 138,
                  "cost": 28.6,
                  "chosen": true
                }
              ]
            },
            "condition_filtering_pct": 100,
            "rows_for_plan": 138,
            "cost_for_plan": 28.6,
            "rest_of_plan": [
              {
                "plan_prefix": [
                  "`champ` `c`"
                ],
                "table": " `info`",
                "best_access_path": {
                  "considered_access_paths": [
                    {
                      "access_type": "ref",
                      "index": "<auto_key0>",
                      "rows": 957.8,
                      "cost": 158611,
                      "chosen": true
                    },
                    {
                      "access_type": "ref",
```

2-2-1

Phase 1 growing phase 在 18~68 行

Phase 2 shrinking phase 在 70~78 行

在 main 內讀入 write target, read target，用 vector<int> regs 一次存起在，write target index 放在 regs[0]。

再把 regs 與整行 line 讀入 threadfunc(reg, str).

Threadfunc 是用 VSmutex(type 為 shared_mutex) 的 vector 進行上鎖。
並在函數(threadfunc)內用 unique_lock 與 shared_lock 分別鎖住 write,read target
算完後，assign 給 write target，最後進入 shrinking phase
我沒有用 semaphore，可能就是因為這樣 data2 才會出錯吧

```cpp
#include<iostream>
#include<thread>
#include<fstream>
#include<cstring>
#include<stdio.h>
#include<cstdlib>
#include<vector>
#include <unistd.h>
#include<semaphore.h>
#include<shared_mutex>
using namespace std;
vector<shared_mutex*> VSshared_mutex;
vector<shared_mutex*> VXshared_mutex;
vector<int> a;
sem_t semaphore;
void threadfunc(vector<int> regs,string str){
    //cout<<regs.size()<<endl;
    //phase 1
    unique_lock<shared_mutex> Xlock(*VSshared_mutex[regs[0]]);
    //VXshared_mutex[regs[0]]->lock();
    //VSshared_mutex[regs[0]]->lock();
    //cout<<"locking X S $"<<regs[0]<<endl;
    for(int i=1;i<regs.size();i++){
        unique_lock<shared_mutex> Slock(*VSshared_mutex[regs[i]]);
        //VSshared_mutex[regs[i]]->lock();
        //cout<<"locking S $"<<regs[i]<<endl;
    }
    int ans = 0;
    string tmp="";
    int i=0;
    for(;str[i]!=' ';i++){
        tmp.push_back(str[i]);
    }
    tmp.erase(tmp.begin());
    tmp = "";
    bool sign=0;
    i+=3;
    for(;i<str.size();i++){
        //cout<<"i="<<i<<endl;
        if(str[i]!=' ')tmp.push_back(str[i]);
```

```
        else{
            //cout<<tmp<<" ";
            if(tmp[0] == '$'){
                tmp.erase(tmp.begin());
                //cout<<"("<<a[stoi(tmp)]<<")"<<" ";
                if(sign==0) ans += a[stoi(tmp)];
                else ans-= a[stoi(tmp)];
            }else if(tmp[0]>='0' and tmp[0]<='9'){
                if(sign==0) ans += stoi(tmp);
                else ans -= stoi(tmp);
            }else if(tmp[0]=='+' or tmp[0]=='-'){
                if(tmp[0]=='+')sign=0;
                else sign=1;
            }
            tmp="";
        }
    }

    if(tmp[0] == '$'){
        tmp.erase(tmp.begin());
        if(sign==0) ans += a[stoi(tmp)];
        else ans-= a[stoi(tmp)];
    }else if(tmp[0]>='0' and tmp[0]<='9'){
        if(sign==0) ans += stoi(tmp);
        else ans -= stoi(tmp);
    }

    a[regs[0]] = ans ;
    //phase 2
    //VXshared_mutex[regs[0]]->unlock();
    VSshared_mutex[regs[0]]->unlock();
    //cout<<"unlocking X S $"<<regs[0]<<endl;
    for(int i=1;i<regs.size();i++){
        VSshared_mutex[regs[i]]->unlock();
        //cout<<"locking S $"<<regs[i]<<endl;
    }

}

```

```
80  int main(int argc,char *argv[])
81  {
82      //fstream file;
83      fstream file_out;
84      //thread Thread[stoi(argv[0])];
85      //file.open("data/data3",ios::in);
86      file_out.open(argv[2],ios::out);
87      int n;cin>>n;
88      vector<thread *> Threadvector(stoi(argv[1]));
89
90      int in;
91      for(int i=0;i<n;i++){
92          cin>>in;
93          a.push_back(in);
94          shared_mutex *SMutex = new(shared_mutex);
95          VSshared_mutex.push_back(SMutex);
96          shared_mutex *XMutex = new(shared_mutex);
97          VXshared_mutex.push_back(XMutex);
98      }
99      string str;
100     vector<vector<int> > ReadWrite;
101     int counter = 0;
102     getline(cin,str);
103     while(getline(cin,str)){
104
105         //cout<<"str="<<str<<endl;
106         if(str=="")break;
107         vector<int> regs;
108         string tmp="";
109         int flag=0;
110         for(int i=0;i<str.length();i++){
111             if(str[i]!=' ')tmp.push_back(str[i]);
112             else{
113                 if(tmp == "=" )flag=1;
114                 else if(tmp[0] == '$'){
115                     tmp.erase(tmp.begin());
116                     int r = stoi(tmp);
117                     bool flag=0;
```

```
                    bool flag=0;
                    for(int i=0;i<regs.size();i++){
                        if(regs[i]==r)flag=1;
                    }
                    if(flag==0){
                        regs.push_back(stoi(tmp));
                        //cout<<tmp<<" ";
                    }
                }
                tmp="";
            }
        }
        if(tmp[0] == '$'){
            tmp.erase(tmp.begin());
            regs.push_back(stoi(tmp));
        }
        Threadvector[counter++%stoi(argv[1])] = new thread(threadfunc,regs,str);// 這裡出問題
        //threadfunc(regs,str);
        ReadWrite.push_back(regs);
        //cout<<"\n";

    }
    for(int i=0;i<stoi(argv[1]);i++){
        Threadvector[i]->join();
    }
    for(int i=0;i<n;i++){
        file_out<<"$"<<i<<" = "<<a[i]<<endl;
    }


}

```

2-2-2

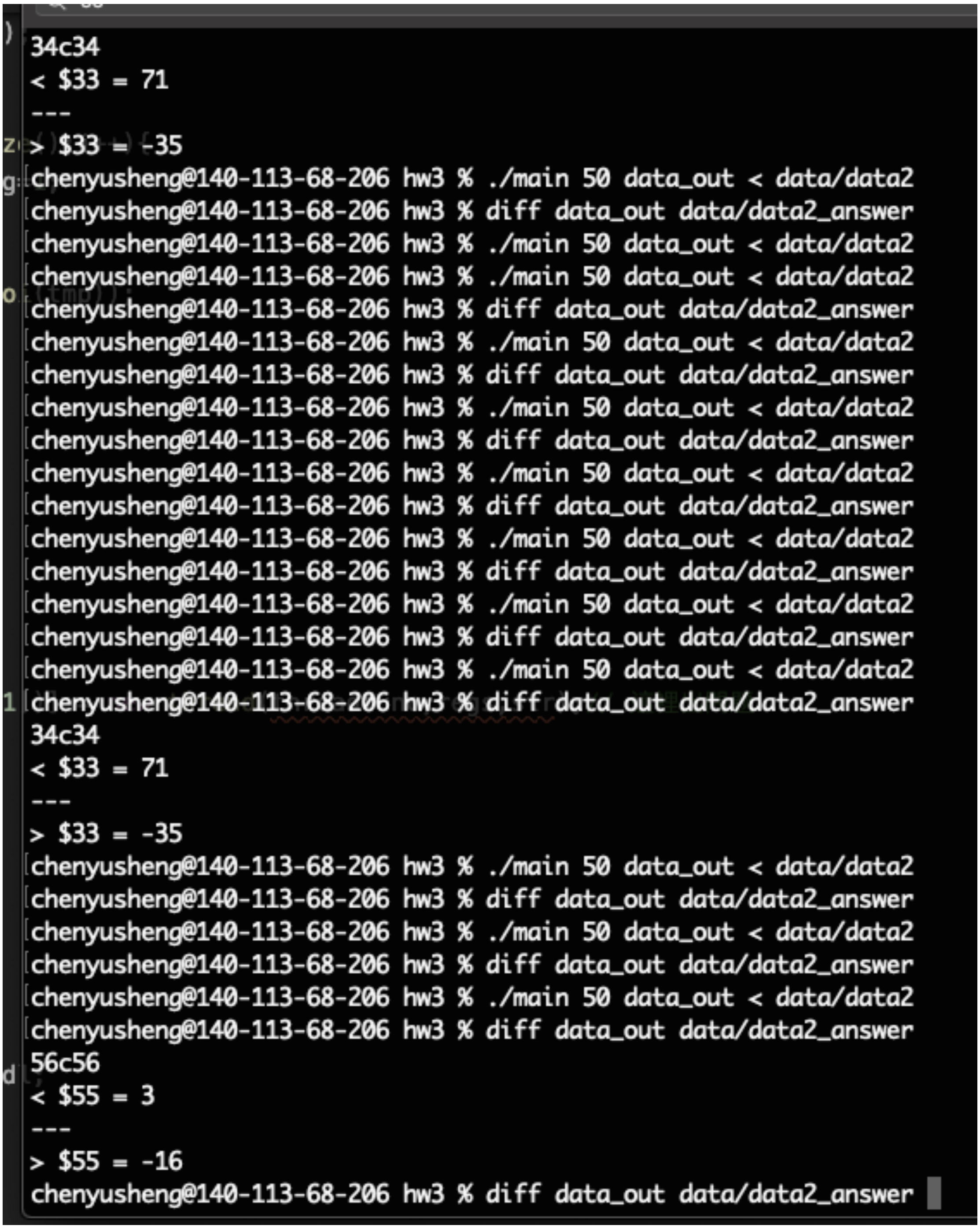
不同的 Thread 數時間都差不多

```
chenyusheng@140-113-68-206 hw3 % ./main 1 data_out < data/data2
totalTime = 2.66387
chenyusheng@140-113-68-206 hw3 % ./main 2 data_out < data/data2
totalTime = 2.68209
chenyusheng@140-113-68-206 hw3 % ./main 3 data_out < data/data2
totalTime = 3.2215
chenyusheng@140-113-68-206 hw3 % ./main 4 data_out < data/data2
totalTime = 3.42754
chenyusheng@140-113-68-206 hw3 % ./main 5 data_out < data/data2
totalTime = 3.06511
chenyusheng@140-113-68-206 hw3 % ./main 6 data_out < data/data2
totalTime = 3.35763
chenyusheng@140-113-68-206 hw3 % ./main 7 data_out < data/data2
totalTime = 3.16709
chenyusheng@140-113-68-206 hw3 % ./main 8 data_out < data/data2
totalTime = 3.34566
chenyusheng@140-113-68-206 hw3 % ./main 9 data_out < data/data2
totalTime = 2.85088
chenyusheng@140-113-68-206 hw3 % ./main 10 data_out < data/data2
totalTime = 2.94897
chenyusheng@140-113-68-206 hw3 % ./main 20 data_out < data/data2
totalTime = 3.36398
chenyusheng@140-113-68-206 hw3 % ./main 30 data_out < data/data2
totalTime = 3.43145
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 40 data_out < data/data2
totalTime = 2.90746
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 50 data_out < data/data2
totalTime = 3.99136
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 60 data_out < data/data2
totalTime = 3.0197
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 70 data_out < data/data2
totalTime = 3.1512
chenyusheng@140-113-68-206 hw3 % ./main 80 data_out < data/data2
totalTime = 3.8153
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 90 data_out < data/data2
totalTime = 3.46677
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 %
```

```
chenyusheng@140-113-68-206 hw3 % ./main 4 data_out < data/data2
totalTime = 3.42754
chenyusheng@140-113-68-206 hw3 % ./main 5 data_out < data/data2
totalTime = 3.06511
chenyusheng@140-113-68-206 hw3 % ./main 6 data_out < data/data2
totalTime = 3.35763
chenyusheng@140-113-68-206 hw3 % ./main 7 data_out < data/data2
totalTime = 3.16709
chenyusheng@140-113-68-206 hw3 % ./main 8 data_out < data/data2
totalTime = 3.34566
chenyusheng@140-113-68-206 hw3 % ./main 9 data_out < data/data2
totalTime = 2.85088
chenyusheng@140-113-68-206 hw3 % ./main 10 data_out < data/data2
totalTime = 2.94897
chenyusheng@140-113-68-206 hw3 % ./main 20 data_out < data/data2
totalTime = 3.36398
chenyusheng@140-113-68-206 hw3 % ./main 30 data_out < data/data2
totalTime = 3.43145
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 40 data_out < data/data2
totalTime = 2.90746
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 50 data_out < data/data2
totalTime = 3.99136
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 60 data_out < data/data2
totalTime = 3.0197
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 70 data_out < data/data2
totalTime = 3.1512
chenyusheng@140-113-68-206 hw3 % ./main 80 data_out < data/data2
totalTime = 3.8153
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 90 data_out < data/data2
totalTime = 3.46677
chenyusheng@140-113-68-206 hw3 % diff data_out data/data2_answer
chenyusheng@140-113-68-206 hw3 % ./main 90 data_out < data/data3
totalTime = 7418.89
chenyusheng@140-113-68-206 hw3 % ./main 80 data_out < data/data3
totalTime = 7384.83
chenyusheng@140-113-68-206 hw3 % ./main 50 data_out < data/data3
totalTime = 7411.9
chenyusheng@140-113-68-206 hw3 % ./main 40 data_out < data/data3
```

```
[chenyusheng@140-113-68-206 hw3 % ./main 4 data_out < data/data3
totalTime = 7877.12
time = 12064204
[chenyusheng@140-113-68-206 hw3 % diff data_out data/data3_answer
[chenyusheng@140-113-68-206 hw3 % ./main 5 data_out < data/data3
totalTime = 7733.43
time = 11930896
[chenyusheng@140-113-68-206 hw3 %
[chenyusheng@140-113-68-206 hw3 % ./main 6 data_out < data/data3
totalTime = 7755.07
time = 11965980
[chenyusheng@140-113-68-206 hw3 %
[chenyusheng@140-113-68-206 hw3 % ./main 7 data_out < data/data3
totalTime = 7805.87
time = 12027632
[chenyusheng@140-113-68-206 hw3 % ./main 8 data_out < data/data3
totalTime = 7837.76
time = 12126766
[chenyusheng@140-113-68-206 hw3 % ./main 9 data_out < data/data3
totalTime = 8028.17
time = 12218750
[chenyusheng@140-113-68-206 hw3 % ./main 10 data_out < data/data3
totalTime = 7746.67
time = 11982474
[chenyusheng@140-113-68-206 hw3 % ./main 11 data_out < data/data3
totalTime = 7734.91
time = 11951977
[chenyusheng@140-113-68-206 hw3 % ./main 12 data_out < data/data3
totalTime = 7737.52
time = 11937252
chenyusheng@140-113-68-206 hw3 %
```

2-2-3



Data2 有非常低的機率大概(10 次出現一次)會出現一兩行不一樣，但只要不快速連續執行就比較沒有錯誤，推測是前面的行數的 read 被卡住了，被後面行數的 write 搶先 X lock，導致 data 錯誤。

到 deadline 前我都找不出有效解法，可能必須要用旗標吧，但因為寫的很多了，不想重寫。

以上都是在 OS X 跑，如果用 linux 跑連 data1 都有可能錯，我不知道為什麼。