Accelerating queries in decision tree with speculation

Jonathan Reymond

12.07.2022

Name of student
School Section/Faculty

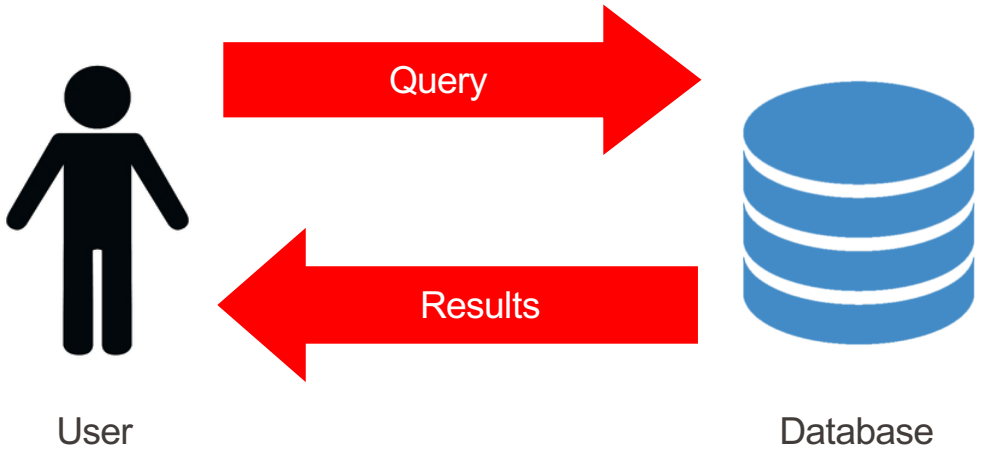
Plan

- Introduction
- Problem
- Techniques
- Experiment
- Conclusion

Jonathan Reymond
School of Computer and
Communication Sciences

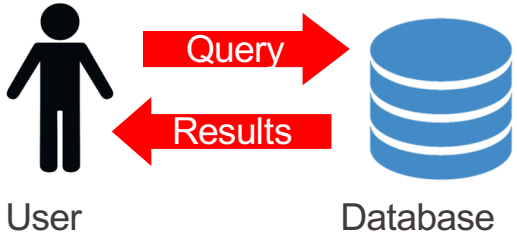
Introduction

- Importance of data science
- Data exploratory analysis

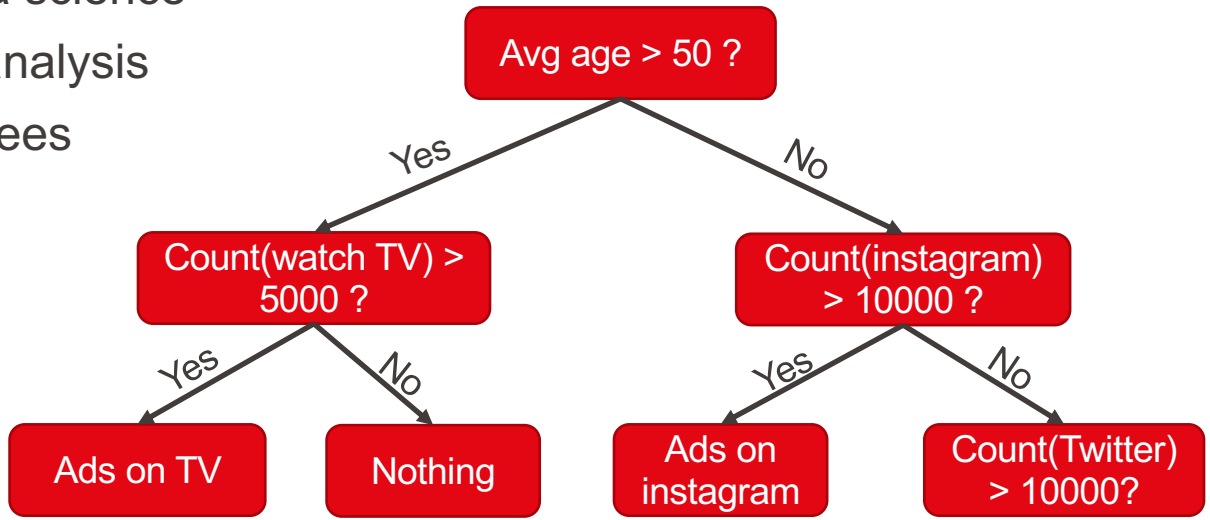


Jonathan Reymond
School of Computer and
Communication Sciences

Introduction

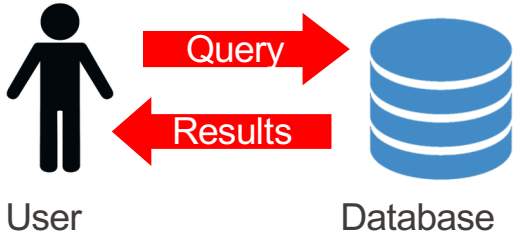


- Importance of data science
- Data exploratory analysis
- Involve decision trees

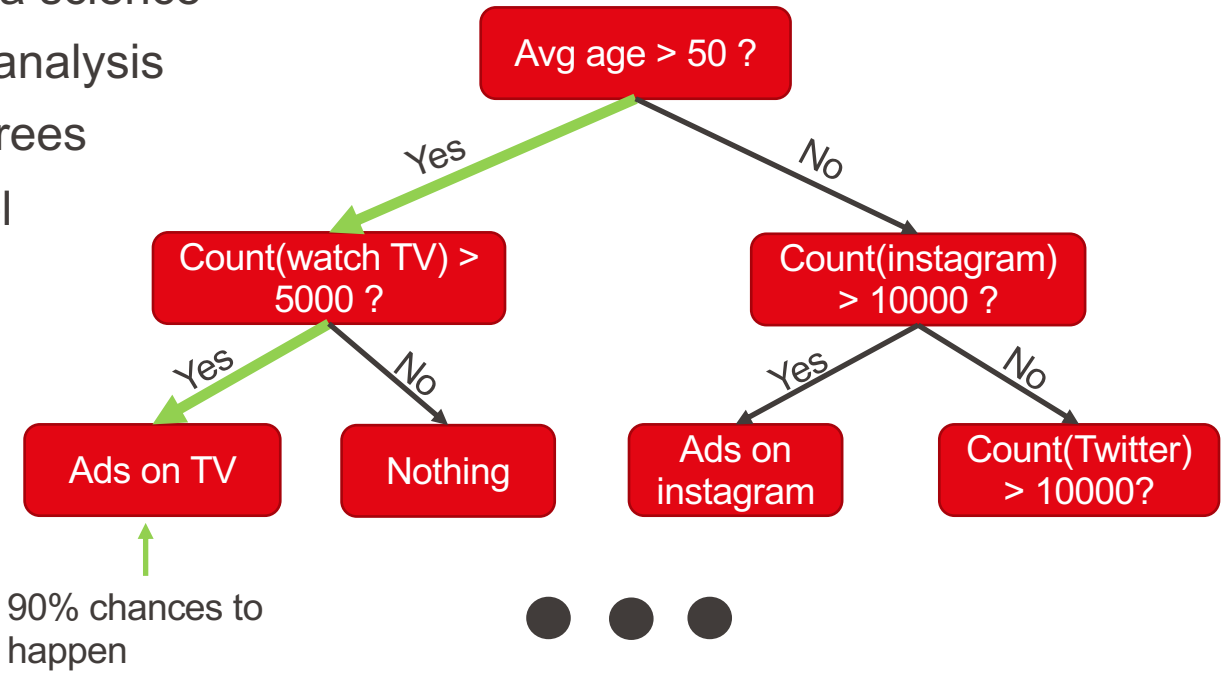


Jonathan Reymond
School of Computer and Communication Sciences

Introduction



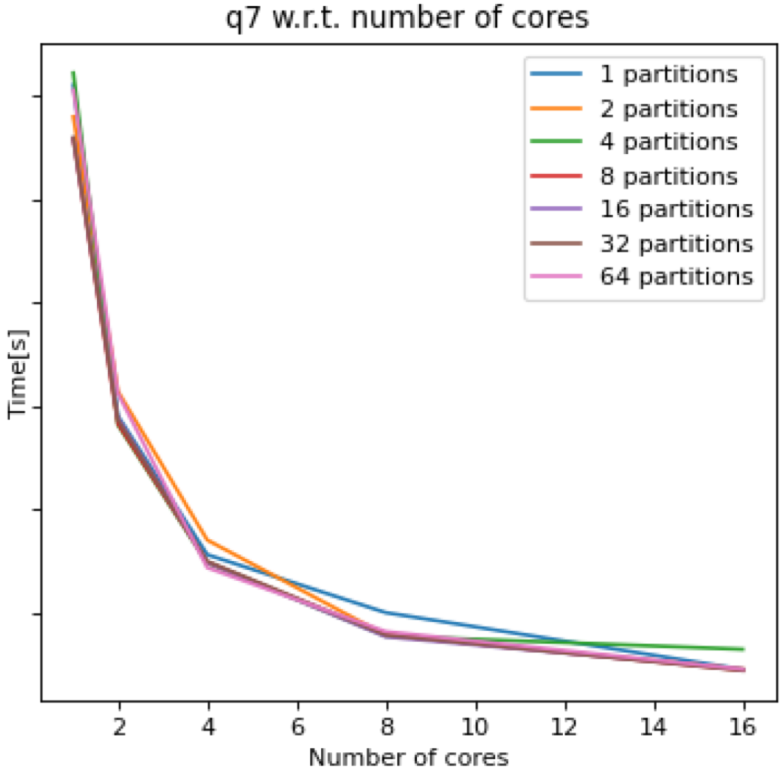
- Importance of data science
- Data exploratory analysis
- Involve decision trees
- Queries in parallel



Jonathan Reymond
School of Computer and Communication Sciences

Problem

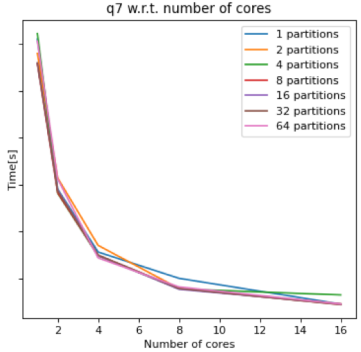
- Based on diminishing returns concept : better in parallel than sequentially

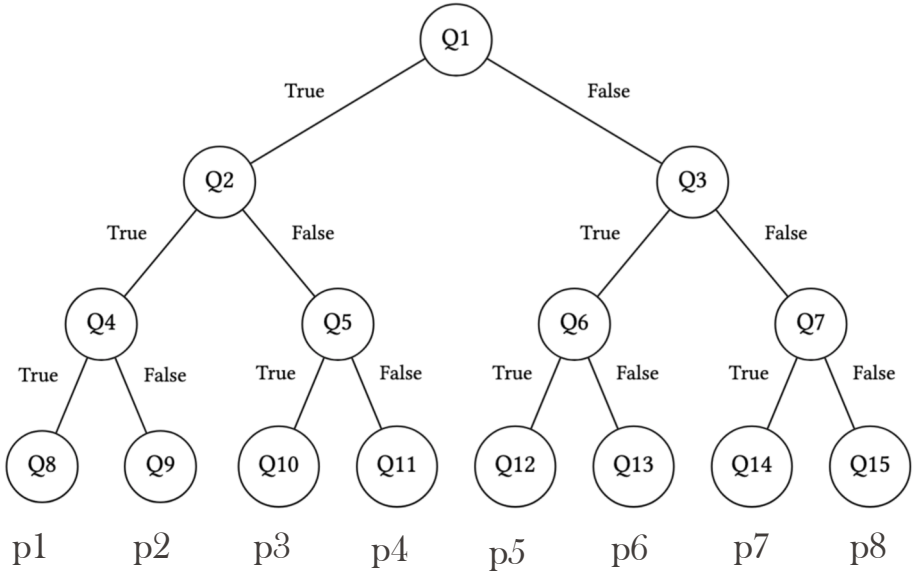


Jonathan Reymond
School of Computer and Communication Sciences

Problem

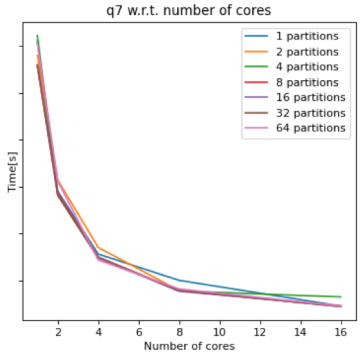
- Based on diminishing returns concept : better in parallel than sequentially
- Goal : Minimize running time most probable branch(es)





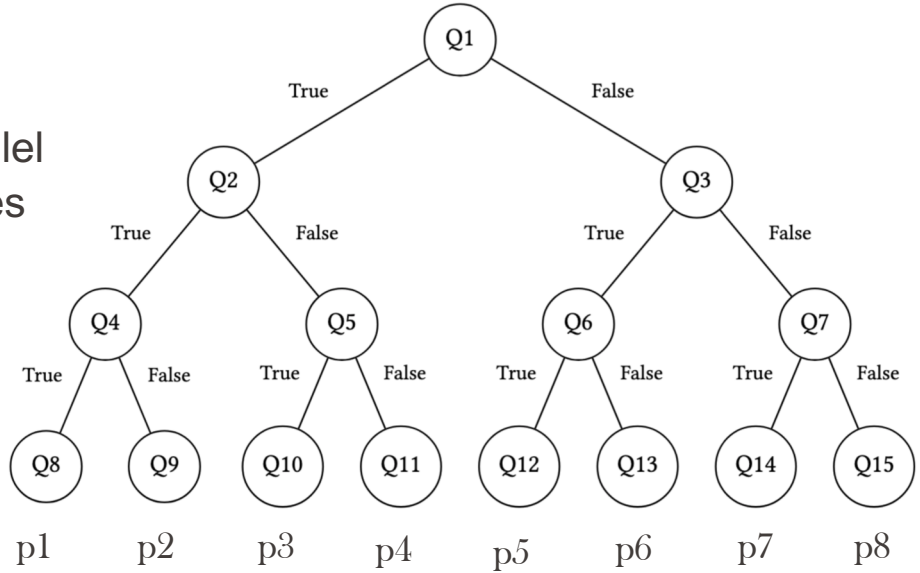
Jonathan Reymond
School of Computer and Communication Sciences

Problem



- Based on diminishing returns concept : better in parallel than sequentially
- Goal : Minimize running time most probable branch(es)
- Challenges :
  - Which queries to run in parallel
  - Allocate right number of cores
  - Mitigate time due to wrong speculation

Constraint programming



Jonathan Reymond
School of Computer and Communication Sciences

Techniques

- Assumptions:
  - Approximation of running time queries in decision tree w.r.t. number of cores
  - Probabilities of each branch to be executed

Jonathan Reymond
School of Computer and
Communication Sciences

Techniques

- Assumptions:
  - Approximation of running time queries in decision tree w.r.t. number of cores
  - Probabilities of each branch to be executed

- Constraint programming :
  - Formulation similar to mixed-integer linear programming
  - Solver based on logic (SAT) != MILP
  - Faster + more set of possible constraints
  - Give optimal result (for small instances)

Jonathan Reymond
School of Computer and
Communication Sciences

Techniques

- Assumptions:
  - Approximation of running time queries in decision tree w.r.t. number of cores
  - Probabilities of each branch to be executed

- Constraint programming :
  - Formulation similar to mixed-integer linear programming
  - Solver based on logic (SAT) != MILP
  - Faster + more set of possible constraints
  - Give optimal result (for small instances)

- Process :
  - Solve problem without probability part
  - Include probabilities

Jonathan Reymond
School of Computer and
Communication Sciences

Techniques

- Without probabilities

$$
\begin{aligned}
& \min \ \textstyle\sum_r k_r \\
& \textstyle\sum_{q=1}^{Q} V_{q_r} \leqslant C && \forall r \in \{1, \ldots, R\} \\
& A\left[I_{q_j}\right] = V_{q_j} && \forall j \in \{0, \ldots, R-1\}, \ \forall q \\
& Distinct\left(I_q\right) && \forall q \in \{1, \ldots, Q\} \\
& k_r = \max_q \left(T_{q, V_{q_r}}\right) && \forall r \in \{1, \ldots, R\}
\end{aligned}
$$

Jonathan Reymond
School of Computer and
Communication Sciences

Techniques

- Without probabilities

$$\min \sum_r k_r$$

$$\sum_{q=1}^{Q} V_{q_r} \leqslant C \qquad \forall r \in \{1, \ldots, R\}$$

$$A\left[I_{q_j}\right] = V_{q_j} \qquad \forall j \in \{0, \ldots, R-1\},\ \forall q$$

$$Distinct\left(I_q\right) \qquad \forall q \in \{1, \ldots, Q\}$$

$$k_r = \max_q \left(T_{q,V_{q_r}}\right) \qquad \forall r \in \{1, \ldots, R\}$$

- With probabilities
  - Variables for each branch

$$\min \sum_{p \in \mathcal{P}} proba_p * pathTime_p$$

$$\sum_{q=1}^{Q} V_{q_r} \leqslant C \qquad \forall r \in \{1, \ldots, R\}$$

$$A\left[I_{q_j}\right] = V_{q_j} \qquad \forall j \in \{0, \ldots, R-1\},\ \forall q \in \{1, \ldots, Q\}$$

$$Distinct\left(I_q\right) \qquad \forall q \in \{1, \ldots, Q\}$$

$$k_r = \max_q \left(T_{q,V_{q_r}}\right) \qquad \forall r \in \{1, \ldots, R\}$$

$$indexRun_q = r \qquad \text{for } r \text{ s.t. } V_{q_r} > 0,\ \forall q$$

$$queryTime_q = \sum_{r=0}^{indexRun_q} k_r \qquad \forall q = 1, \ldots, Q$$

$$pathTime_p = \max_{q \in set_p} \left(queryTime_q\right) \qquad \forall p \in \mathcal{P}$$

Jonathan Reymond
School of Computer and Communication Sciences

Techniques

- Without probabilities

$$\min \sum_r k_r$$

$$\sum_{q=1}^{Q} V_{q_r} \leqslant C \qquad \forall r \in \{1, \ldots, R\}$$

$$A\left[I_{q_j}\right] = V_{q_j} \qquad \forall j \in \{0, \ldots, R-1\},\ \forall q$$

$$Distinct\left(I_q\right) \qquad \forall q \in \{1, \ldots, Q\}$$

$$k_r = \max_q \left(T_{q,V_{q_r}}\right) \qquad \forall r \in \{1, \ldots, R\}$$

- With probabilities
  - Variables for each branch

$$\min \sum_{p \in \mathcal{P}} proba_p * pathTime_p$$

$$\sum_{q=1}^{Q} V_{q_r} \leqslant C \qquad \forall r \in \{1, \ldots, R\}$$

$$A\left[I_{q_j}\right] = V_{q_j} \qquad \forall j \in \{0, \ldots, R-1\},\ \forall q \in \{1, \ldots, Q\}$$

$$Distinct\left(I_q\right) \qquad \forall q \in \{1, \ldots, Q\}$$

$$k_r = \max_q \left(T_{q,V_{q_r}}\right) \qquad \forall r \in \{1, \ldots, R\}$$

$$indexRun_q = r \qquad \text{for } r \text{ s.t. } V_{q_r} > 0,\ \forall q$$

$$queryTime_q = \sum_{r=0}^{indexRun_q} k_r \qquad \forall q = 1, \ldots, Q$$

$$pathTime_p = \max_{q \in set_p} \left(queryTime_q\right) \qquad \forall p \in \mathcal{P}$$

Jonathan Reymond
School of Computer and Communication Sciences

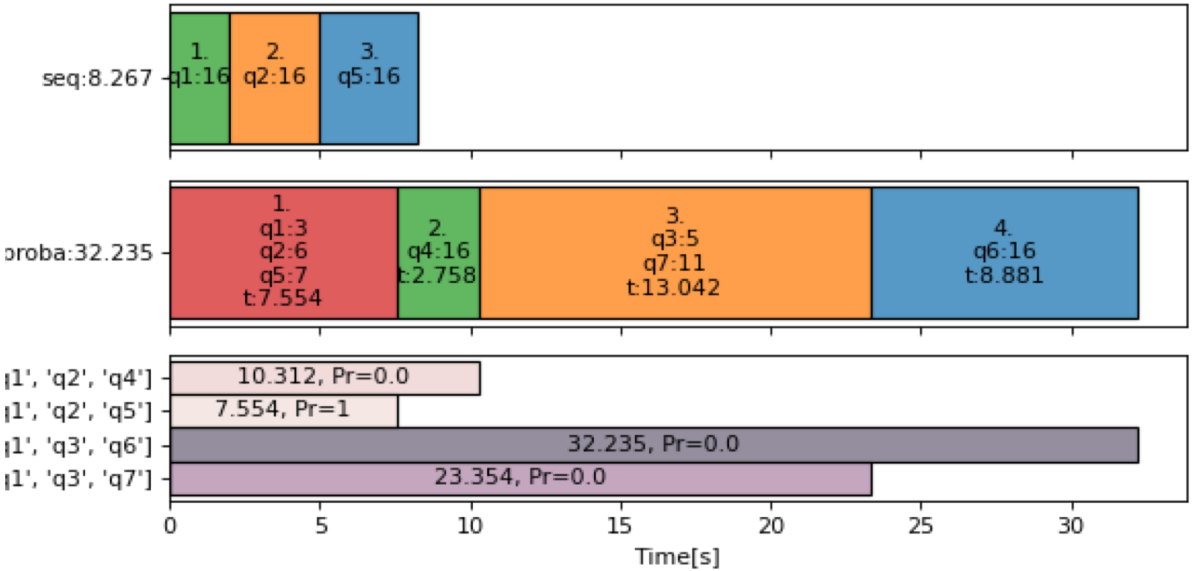
Experiment : Setup

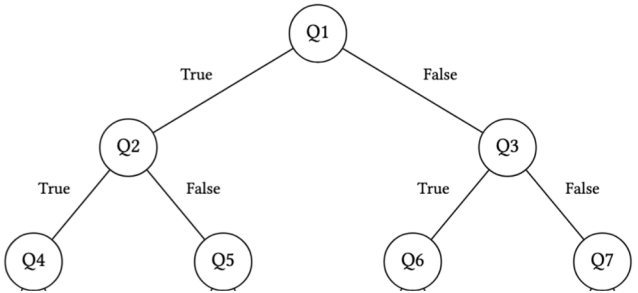
- IMDB dataset : large, multiple tables
- Sql queries in Spark (join-order benchmark from Viktor Leis and al.)
- Running time computation :
  - Linux machine kernel version 4.15., 16 cores of type Intel Xeon E5-2640 v2 at 2GHz, 8 cores per socket, 256 GB of main memory.
- Study results from decision trees of height 2 and 4

Jonathan Reymond
School of Computer and
Communication Sciences

Experiment : Results

- Tree of height 2

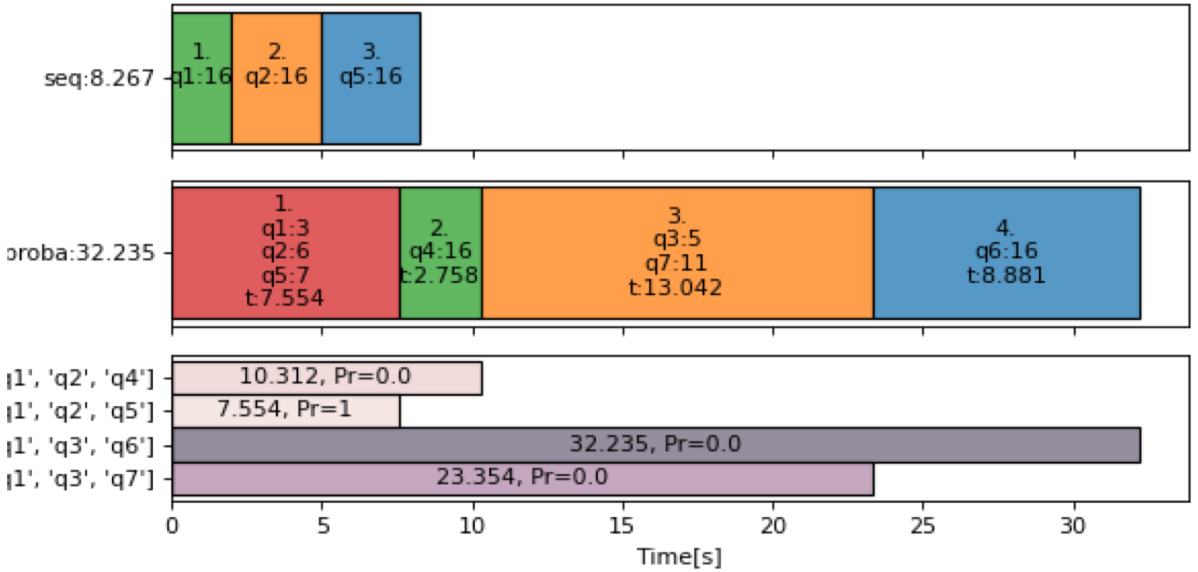




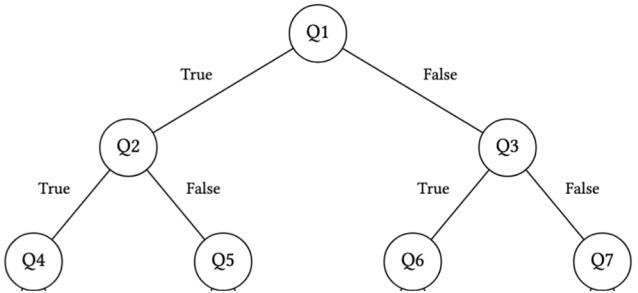
Jonathan Reymond
School of Computer and Communication Sciences

Experiment : Results

- Tree of height 2



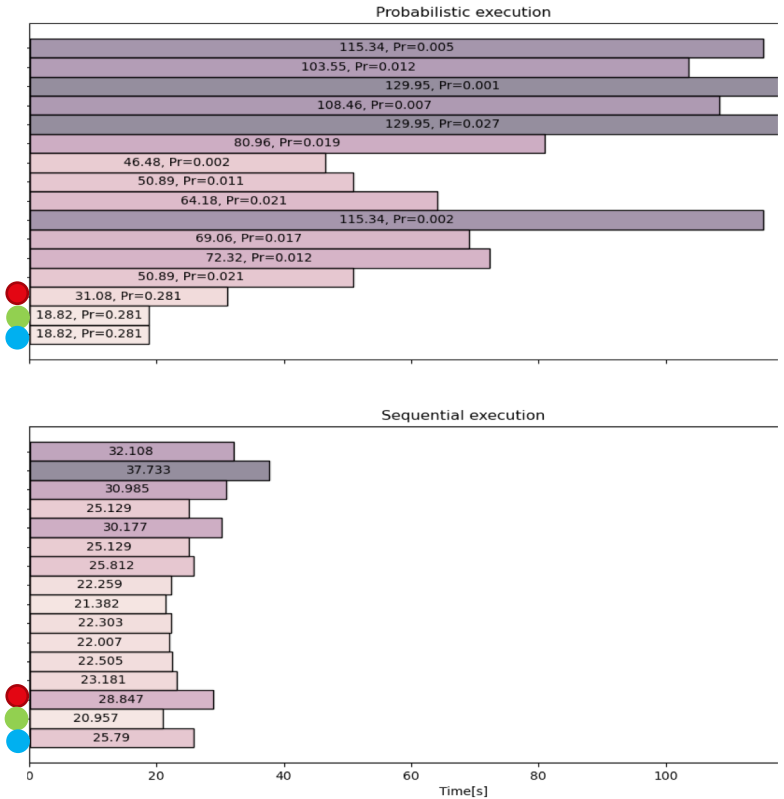
- Can skip some batches
- speed-up ~ 10%



Jonathan Reymond
School of Computer and Communication Sciences

Experiment : Results

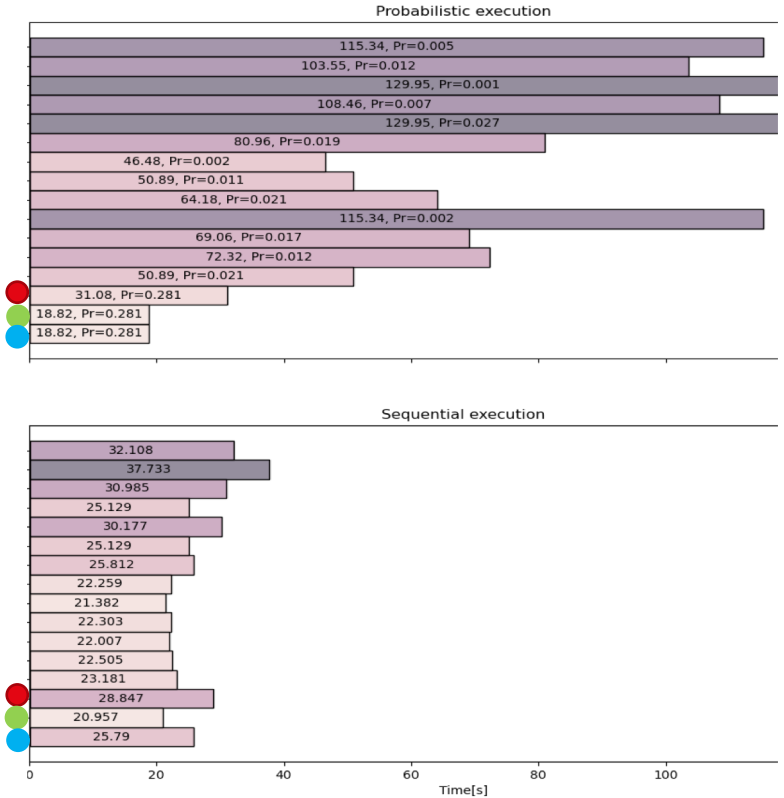
- 1) High probability 3 left branches
  - Speed-up :
    - 2 of them (blue: 37%, green:11%)
    - small cost for red (< 2 sec)



Jonathan Reymond
School of Computer and
Communication Sciences

Experiment : Results

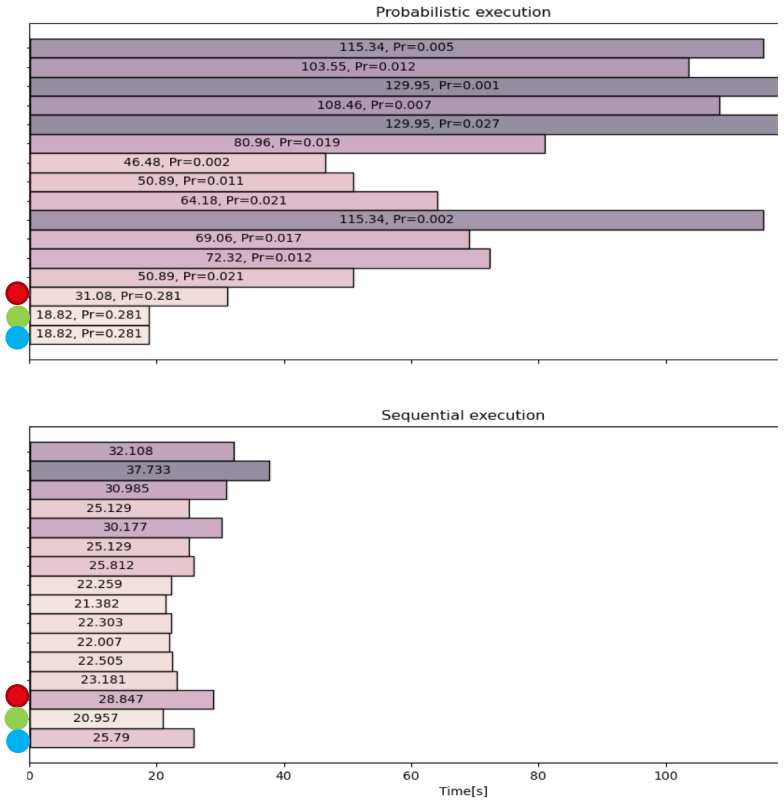
- 1) High probability 3 left branches
  - Speed-up :
    - 2 of them (blue: 37%, green:11%)
    - small cost for red (< 2 sec)
- Cost dependent on branch location
- 2) High probability only to left-most and right-most branches
  - Right : 18.1[s] instead of 25.8[s]
  - Left : 40.4[s] instead of 32.1[s]



Jonathan Reymond
School of Computer and
Communication Sciences

Experiment : Results

- 1) High probability 3 left branches
  - Speed-up :
    - 2 of them (blue: 37%, green:11%)
    - small cost for red (< 2 sec)
- Cost dependent on branch location
- 2) High probability only to left-most and right-most branches
  - Right : 18.1[s] instead of 25.8[s]
  - Left : 40.4[s] instead of 32.1[s]
- Cost speculation higher as tree grows
- Gains dependent on probabilities

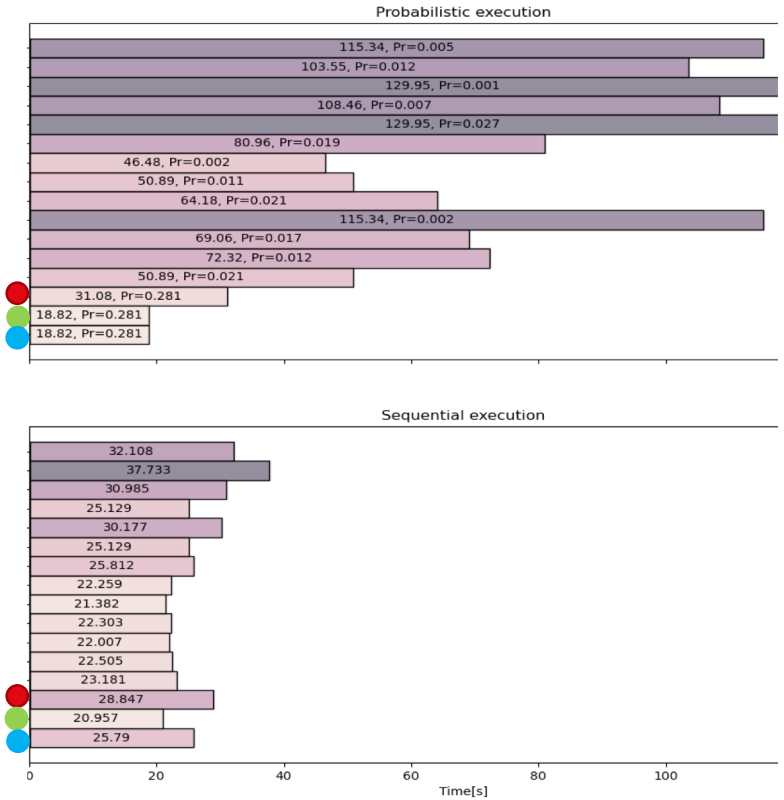


Jonathan Reymond
School of Computer and Communication Sciences

Experiment : Results

- 1) High probability 3 left branches
  - Speed-up :
    - 2 of them (blue: 37%, green:11%)
    - small cost for red (< 2 sec)
- Cost dependent on branch location
- 2) High probability only to left-most and right-most branches
  - Right : 18.1[s] instead of 25.8[s]
  - Left : 40.4[s] instead of 32.1[s]

- Cost speculation higher as tree grows
- Gains dependent on probabilities



Consider only subtree

Jonathan Reymond
School of Computer and Communication Sciences

Conclusion

- Optimizer minimizing the running time of the most probable branch(es)

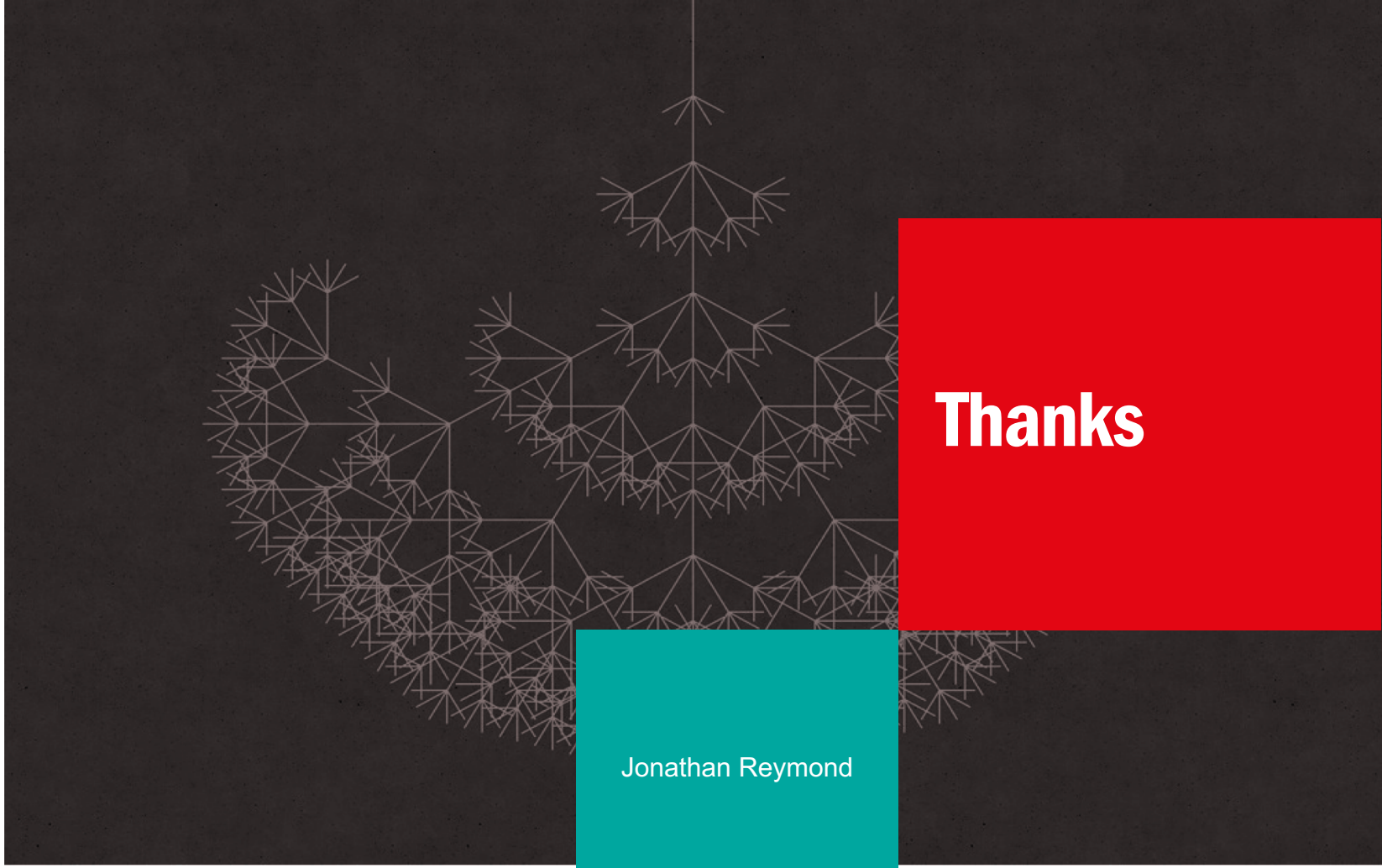
- Speed-up of this approach
  - Limitations (probabilities, height, …)
  - Solutions (subtree, …)

Jonathan Reymond
School of Computer and
Communication Sciences

Conclusion

- Optimizer minimizing the running time of the most probable branch(es)
- Speed-up of this approach
  - Limitations (probabilities, height, …)
  - Solutions (subtree, objective function)
- Improvements:
  - Consider Sql query properties during analysis (tables, what can be shared)
  - More resources => more diminishing returns => more speed-up

Jonathan Reymond
School of Computer and
Communication Sciences



Thanks

Jonathan Reymond

Jonathan Reymond
School of Computer and
Communication Sciences