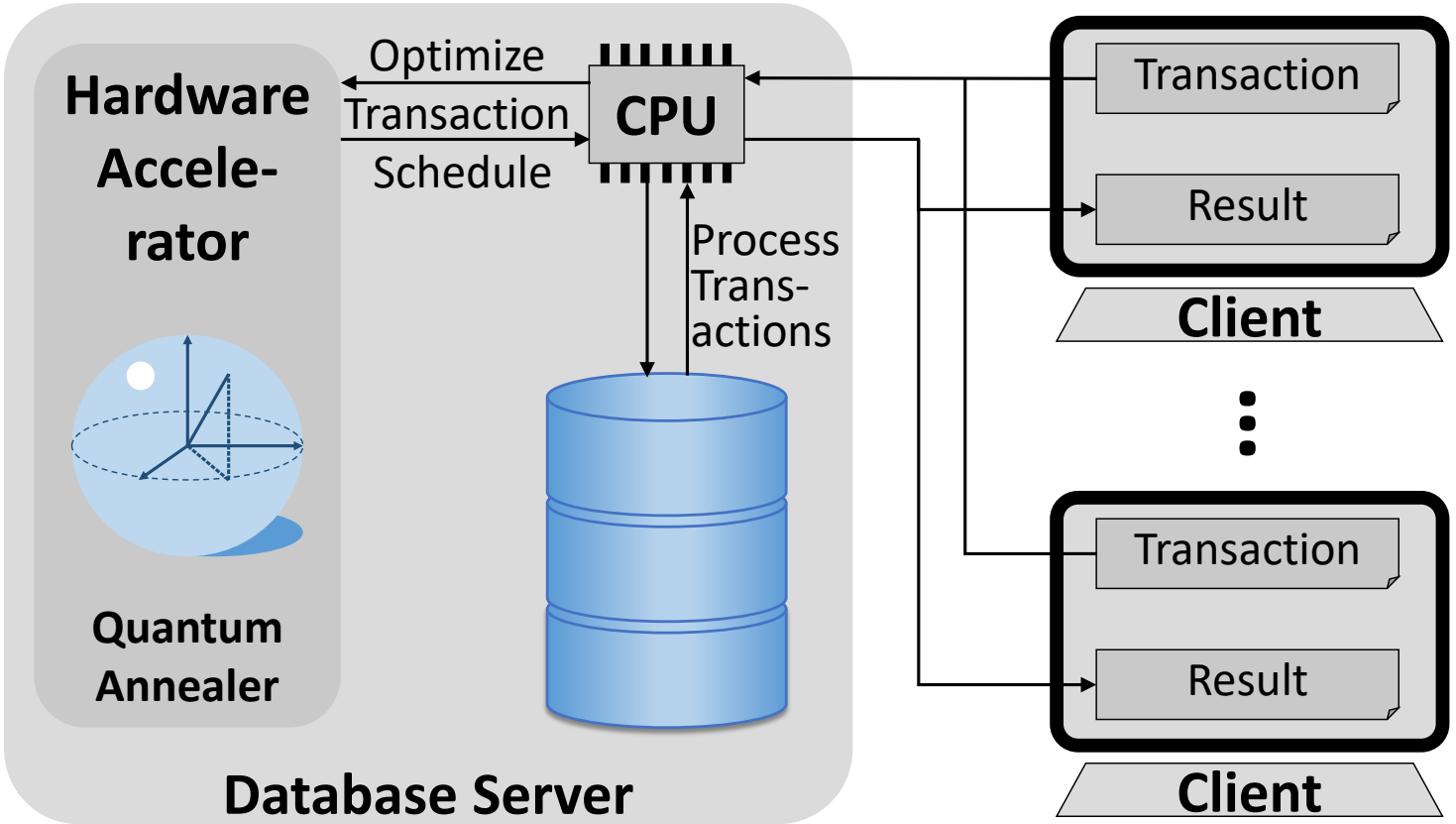
IDEAS 2020

August 12-18, 2020, On-line

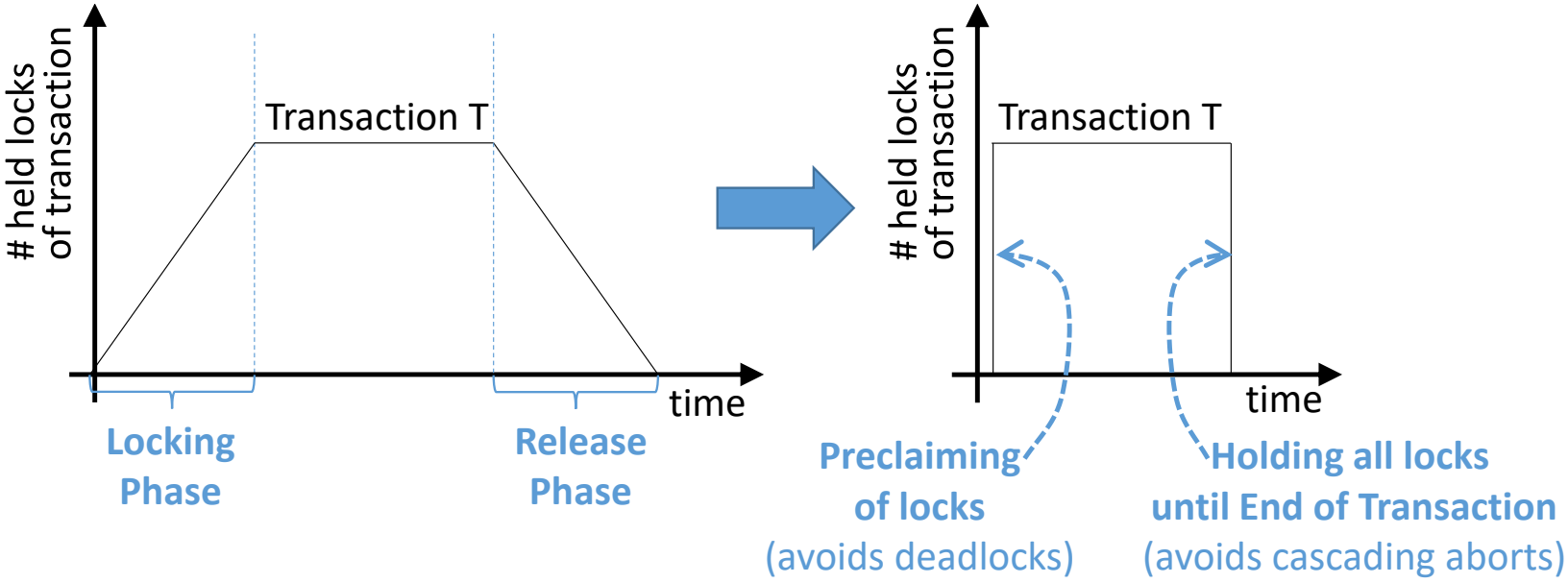
Avoiding Blocking by Scheduling Transactions using Quantum Annealing

Professor Dr. rer. nat. habil. Sven Groppe
https://www.ifis.uni-luebeck.de/index.php?id=groppe

Using Hardware Accelerator for optimizing Transaction Schedules



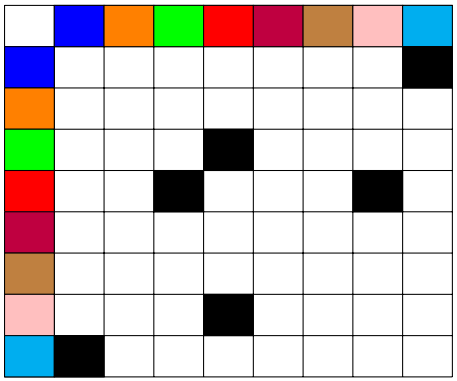
2 Phase Locking (2PL) versus Strict Conservative 2PL



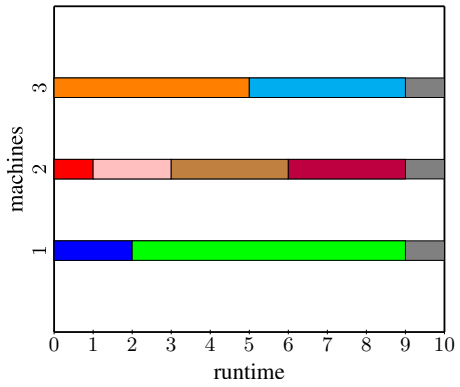
- required locks to be determined by
  - static analysis of transaction, or if static analysis is not possible:
  - an additional phase at runtime before transaction processing
    - A. Thomson et al., "Calvin: Fast distributed transactions for partitioned database systems", SIGMOD 2012.

Optimizing Transaction Schedules

- Variant of job shop schedule problem (JSSP):
  - Multi-Core CPU
    - Process whole job (here transaction) on core X
  - Schedule: ∀ cores: Sequence of jobs to be processed
  - What is the optimal schedule for minimal overall processing time?
- Additionally to JSSP:
  Blocking transactions not to be processed in parallel
- Example:



Black: Blocking transactions

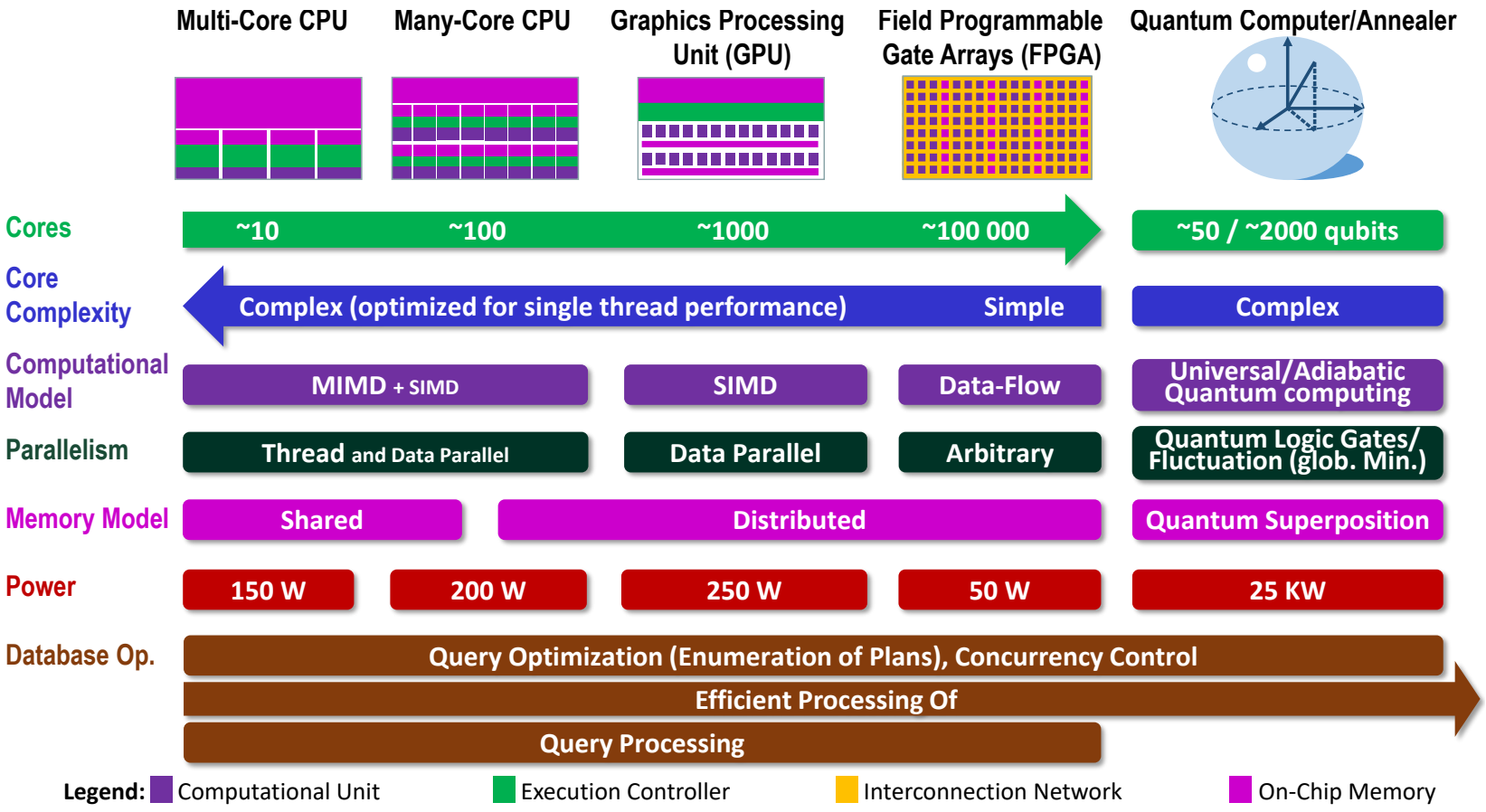


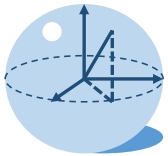
Transaction schedule

- JSSP is among the hardest combinatorial optimizing problems*
- ⇒ Hardware accelerating the optimization of transaction schedules

* M.R. Garey, D.S. Johnson and R. Sethi, The Complexity of Flowshop and Jobshop Scheduling, 1(2):117-129, 1976

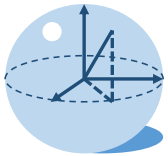
Architectures of Emergent Hardware



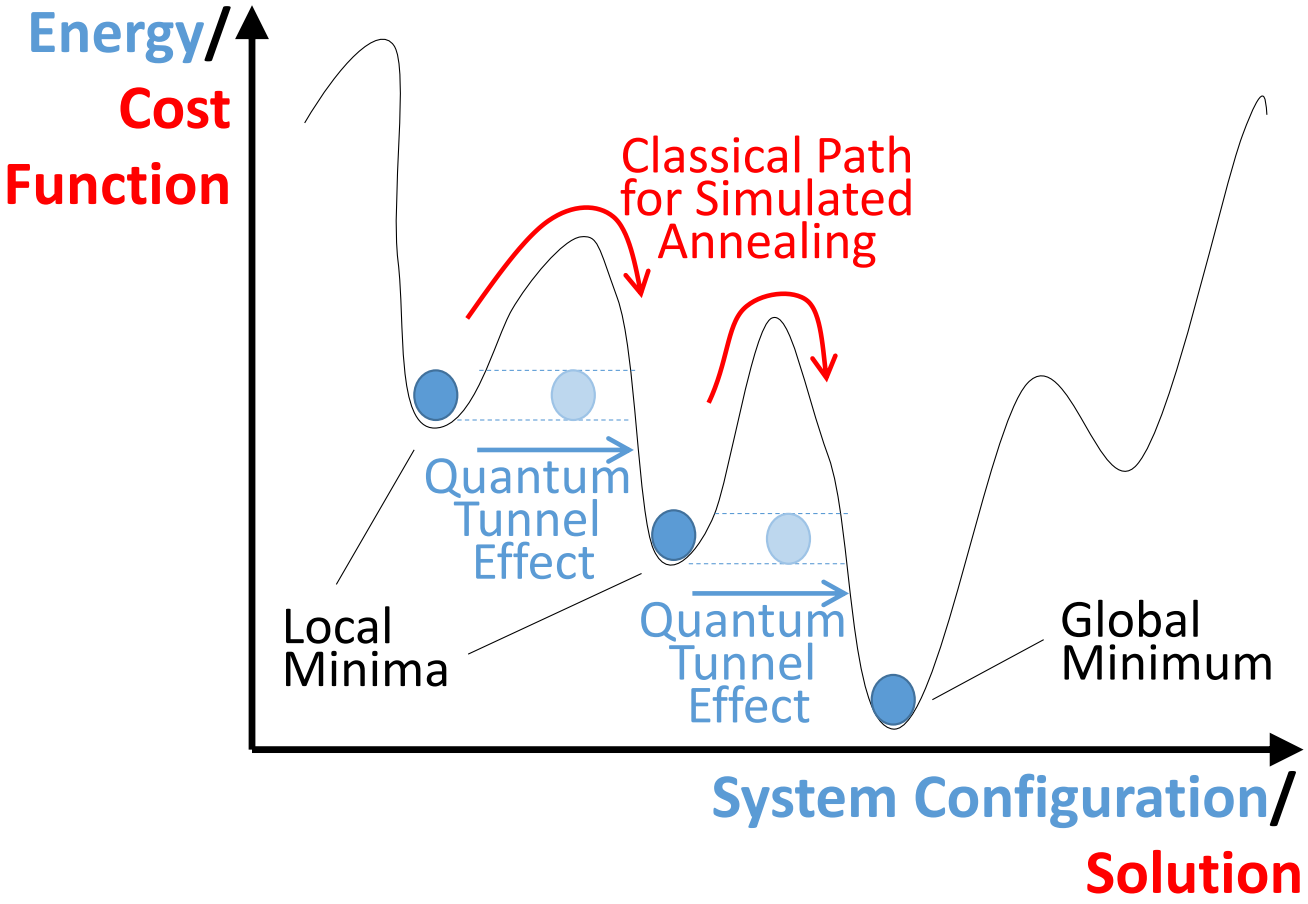


Quantum Computer

- use of quantum-mechanical phenomena such as superposition and entanglement to perform computation
- Different types of quantum computer, e.g.
  - Digital Quantum Computer
    - uses quantum logic gates to do computation
    - measurement (sometimes called observation) assigns the observed variable to a single value
  - Quantum Annealing
    - metaheuristic for finding the global minimum of a given objective function over a given set of candidate solutions
    - i.e., some way to solve a special type of mathematical optimization problem



Quantum versus Simulated Annealing



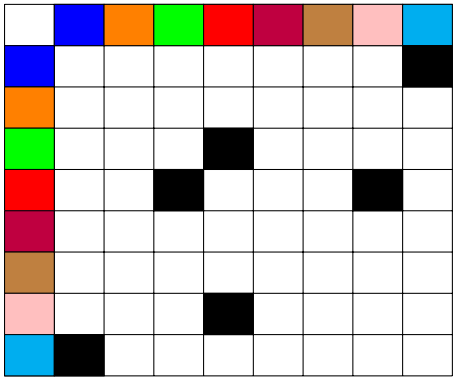
Optimizing Transaction Schedules via Quantum Annealing

- Transaction Model
  - T: set of transactions with |T| = n
  - M: set of machines with |M| = k
  - $O \subseteq T \times T$: set of blocking transactions
  - l_i: length of transaction i
  - R: maximum execution time
  - upper bound $r_i = R - l_i$ for start time of transaction i
- Example
  - T = {t_1, t_2, t_3}, n=3
  - M = {m_1, m_2}, k=2
  - O = {(t_2, t_3)}
  - $l_1 = 2$, $l_2 = 1$, $l_3 = 1$
  - R = 2
  - $r_1 = 0$, $r_2 = 1$, $r_3 = 1$
- Quadratic unconstrained binary optimization (QUBO) problems (solving is NP-hard)
  - A QUBO-problem is defined by N weighted binary variables $X_1, ..., X_N \in 0, 1$, either as linear or quadratic term to be minimized:

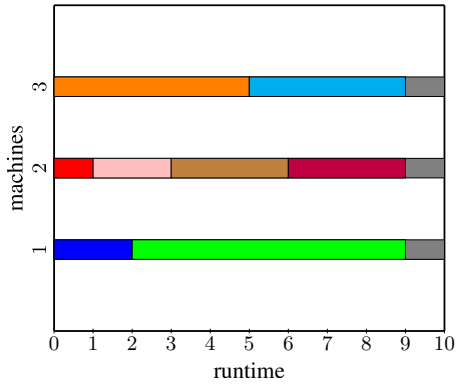
$$\sum_{0<i\leq N} w_i X_i + \sum_{i\leq j\leq N} w_{ij} X_i X_j, \text{ where } w_i, w_{ij} \in \mathbb{R}$$

Optimizing Transaction Schedules via Quantum Annealing

- Multi-Core CPU
  - Process whole transaction on core X
- Solution formulated as set of binary variables
  - $X_{i,j,s}$ is 1 iff transaction t_i is started at time s on machine m_j, otherwise 0
- Example:



Black: Blocking transactions



Transaction schedule

- Solution:
  $X_{1,1,0}$, $X_{3,1,2}$, $X_{4,2,0}$, $X_{7,2,1}$, $X_{6,2,3}$, $X_{5,2,6}$, $X_{2,3,0}$, $X_{8,3,5}$

Optimizing Transaction Schedules via Quantum Annealing

- Valid Solution
  - A: each transaction starts exactly once

$$A = \underbrace{\sum_{i=1}^{n}}_{\text{transactions}} \Big(\underbrace{\sum_{j=1}^{k}}_{\text{machines}} \underbrace{\sum_{s=0}^{r_i}}_{\text{start times}} X_{i,j,s} - 1 \Big)^2$$

  - B: transactions cannot be executed at the same time on the same machine

$$B = \underbrace{\sum_{j=1}^{k}}_{\text{machines}} \overbrace{\sum_{i_1=1}^{n-1}}^{\text{transactions without } t_n} \underbrace{\sum_{s_1=0}^{r_{i_1}}}_{\text{start times}} \overbrace{\sum_{i_2=i_1+1}^{n}}^{\text{remaining transactions}} \underbrace{\sum_{s_2=q}^{p}}_{\text{invalid start times}} X_{i_1,j,s_1} X_{i_2,j,s_2} \text{ for } q = \max\{0, s_1 - l_{i_2} + 1\}, p = \min\{s_1 + l_{i_1}, r_{i_2}\}$$

  - C: transactions that block each other cannot be executed at the same time

$$C = \underbrace{\sum_{\{t_{i_1}, t_{i_2}\} \in O}}_{\text{blocking transactions}} \overbrace{\sum_{j_1=1}^{k}}^{\text{machines}} \underbrace{\sum_{s_1=0}^{r_{i_1}}}_{\text{start times}} \overbrace{\sum_{j_2 \in J}}^{\text{remaining machines}} \underbrace{\sum_{s_2=q}^{p}}_{\text{invalid start times}} X_{i_1,j_1,s_1} X_{i_2,j_2,s_2} \text{ for } J = \{1, ..., k\} \setminus \{j_1\}, q = \max\{0, s_1 - l_{i_2} + 1\}, p = \min\{s_1 + l_{i_1}, r_{i_2}\}$$

Optimizing Transaction Schedules via Quantum Annealing

- Optimal Solution
  - D: minimizing the maximum execution time

$$D = \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{s=0}^{r_i} w_{s+l_i} X_{i,j,s}, \text{ where } w_{s+l_i} = \frac{(k+1)^{s+l_i-1}}{(k+1)^R} < 1$$

    - Increasing weights: Weight of step n is larger than of all preceding steps 1 to n-1 $\Rightarrow$ preferring transactions ending earlier
    - Weigths in A, B and C ≥ 1
      $\Rightarrow$ first priority is validity, second priority is optimality
- Overall Solution
  - Minimize $P = A + B + C + D$

Optimizing Transaction Schedules via Quantum Annealing

- Experiments on real Quantum Annealer (D-Wave 2000Q cloud service)
  - first minute free (afterwards too much for our budget)
- Versus Simulated Annealing on CPU
- Preprocessing time/Number of QuBits: $O((n \cdot k \cdot R)^2)$

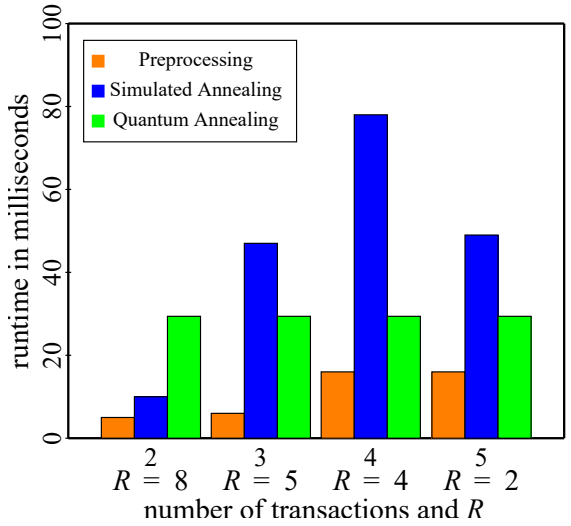


Fig.	k	n	R	O	$l_1, \ldots, l_n$	$r_1, \ldots, r_n$	req. var.
11	2	2	8	$\{\}$	8, 4	0, 4	8
		3	5	$\{(t_1, t_3)\}$	4, 5, 1	1, 0, 4	10
		4	4	$\{(t_2, t_4)\}$	3, 2, 1, 2	1, 2, 3, 2	16
		5	2	$\{(t_1, t_2), (t_4, t_5)\}$	1, 1, 1, 1, 1	1, 1, 1, 1, 1	10

Summary & Conclusions

- Scheduling transactions as jobshop problem with additionally considering blocking transactions
  - Hard combinatorial optimization problem
    $\Rightarrow$ hardware acceleration
- Formulating transaction schedule problem as quadratic unconstrained binary optimization (QUBO) problem
- Hardware acceleration via quantum annealing
  - Constant execution time in contrast to simulated annealing on classical computers
  - Preprocessing time increasing with larger problem sizes
- Future Work
  - Caching of formulas to minimize preprocessing time
  - Quantum annealing of other database problems