Some Recent Musings on Quantum Computing

Wolfgang Mauerer <wolfgang.mauerer@othr.de>
Technical University of Applies Sciences Regensburg

Based on joint work with M. Schönberger, M. Franz, L. Wolf, I. Sax, T. Krüger (OTH)
and M. Periyasamy, A. Plinge, D. D. Scherer, S. Scherzinger, Ch. Ufrecht, Ch. Mutschler

July 5th, 2022

GPG/PGP-ID 98356E1E, Fingerprint: 5920 9407 AB5C 8B28 3C7B 4F02 F16F 2523 9835 6E1E.

Quantum Computing Algorithms & Applications

QC: Research Necessities

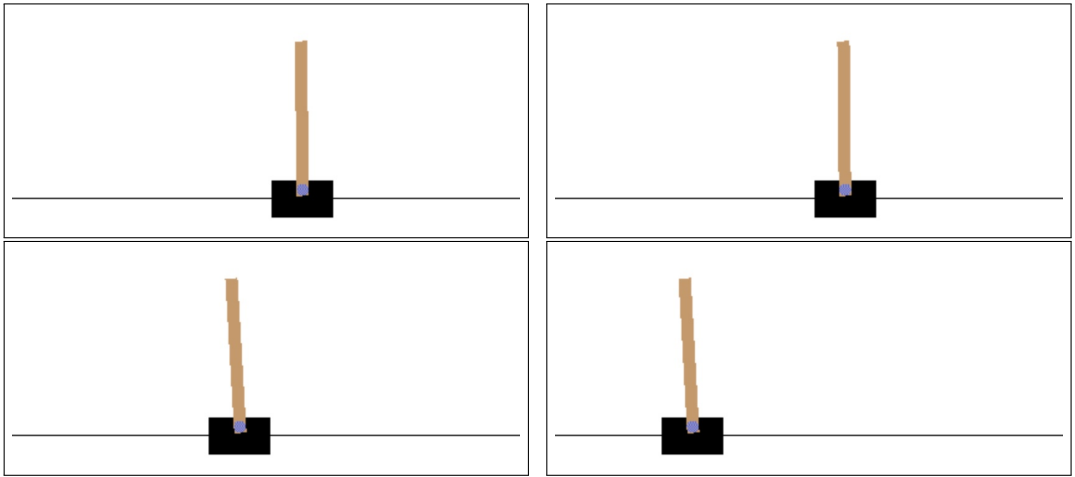
1. Explore precise limits (fundamental & technical)
2. **Consider problems holistically**
3. **Maximise utility**

Possible Applications

- Reinforcement Learning
- Optimisation
- Databases

Reinforcement Learning

Reinforcement Learning I



Ingredients

- State
- Action
- Reward/Punishment
- Tradeoff between current and future gains

Ingredients

- State
- Action
- Reward/Punishment
- Tradeoff between current and future gains

Bellman Optimality Equation

$$Q_*(s,a) = \mathbb{E}\Big[R_t + \gamma \max_{a'} Q_*(S_{t+1}, a') \mid S_t = s, A_t = a\Big]$$

Ingredients

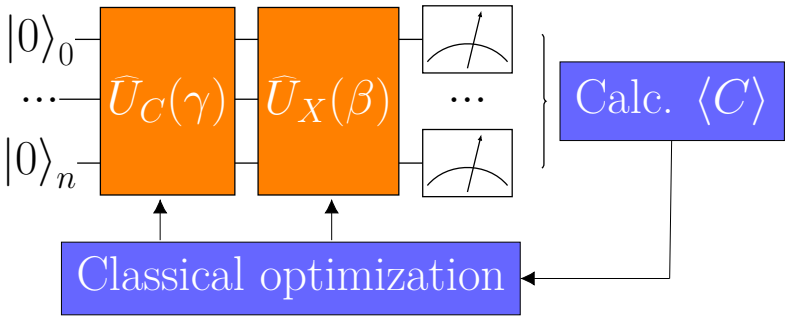
- State
- Action
- Reward/Punishment
- Tradeoff between current and future gains

Advantage & Disadvantage

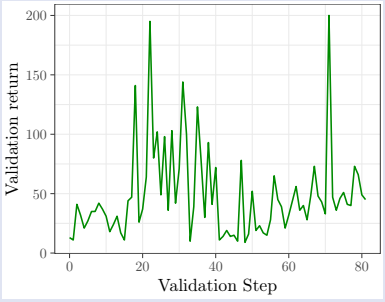
- Switch between *exploration* and *exploitation*
- Q function is hard to compute

Bellman Optimality Equation

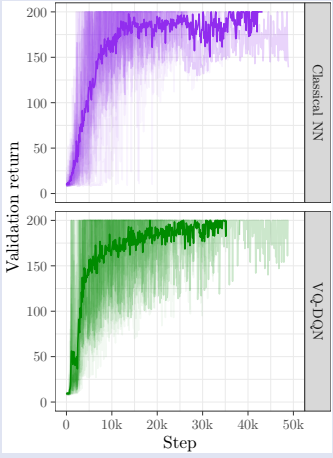
$$Q_*(s,a) = \mathbb{E}\Big[R_t + \gamma \max_{a'} Q_*(S_{t+1}, a') \mid S_t = s, A_t = a\Big]$$


$|0\rangle_0$
$\cdots$
$|0\rangle_n$
$\widehat{U}_C(\gamma)$
$\widehat{U}_X(\beta)$
$\cdots$
Calc. $\langle C \rangle$
Classical optimization

IBMQ performance



QRL: Classical vs. hybrid VQ-DQN



Optimisation


D:WAVE 2000Q
D:WAVE
D:WAVE 2000Q
D:WAVE

Boolean Satisfiability

$$f(\vec{x}) = (x_1 \vee \bar{x}_2 \vee \bar{x}_4) \wedge (x_2 \vee x_3 \vee x_5) \wedge (x_1 \vee x_4 \vee \bar{x}_6)$$

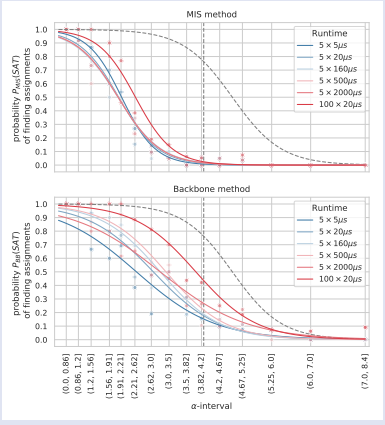
Annealers and QUBOs

$$\min_{\vec{x}} \left(\sum_i c_{ii} x_i + \sum_{i \neq j} c_{ij} x_i x_j \right)$$

Reductions

- Choi: k-SAT $\leq_{\mathrm{p}}$ MIS $\leq_{\mathrm{p}}$ QUBO
- Backbone

$\alpha = |C|/|V|$



Boolean Satisfiability

$$f(\vec{x}) = (x_1 \vee \bar{x}_2 \vee \bar{x}_4) \wedge (x_2 \vee x_3 \vee x_5) \wedge (x_1 \vee x_4 \vee \bar{x}_6)$$

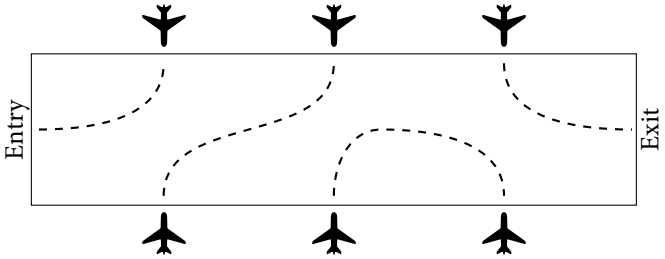
Annealers and QUBOs

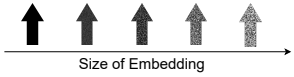
$$\min_{\vec{x}} \left(\sum_i c_{ii} x_i + \sum_{i \neq j} c_{ij} x_i x_j \right)$$

Reductions

- Choi: k-SAT $\leq_p$ MIS $\leq_p$ QUBO
- Backbone

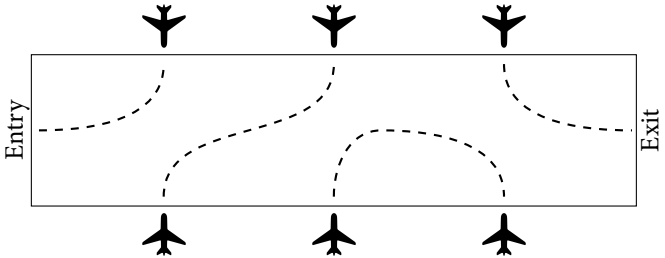
Maximise Utility

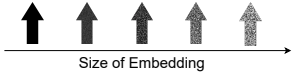




$$\min_{\vec{x}} \left(\sum_{i \in I} c_{ii} x_i + \sum_{(i,j) \in K} c_{ij} x_i x_j \right)$$

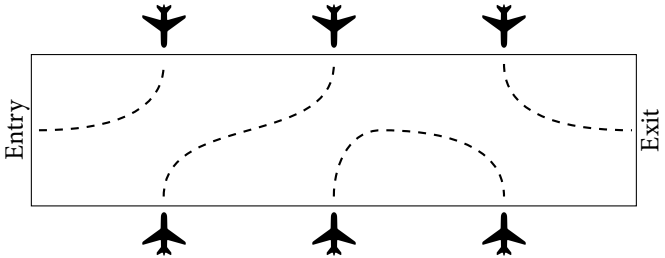
Maximise Utility

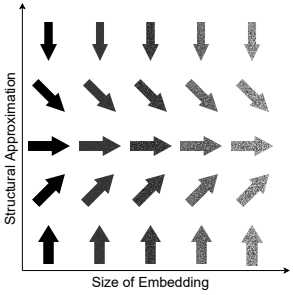




$$\min_{\vec{x}} \Bigg(\underbrace{\sum_{i \in I} c_{ii} x_i + \sum_{(i,j) \in K} c_{ij} x_i x_j}_{\text{Correctness}} + \underbrace{\sum_{i \in I'} c_{ii} x_i + \sum_{(i,j) \in K'} c_{ij} x_i x_j}_{\text{Optimisation}} \Bigg)$$

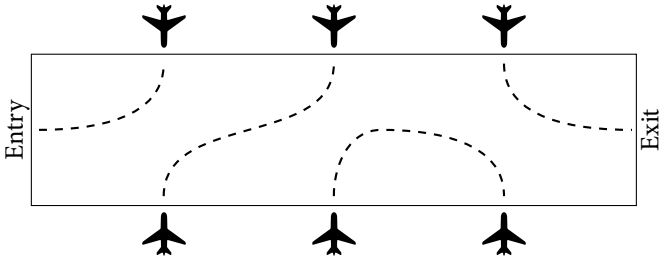
Maximise Utility

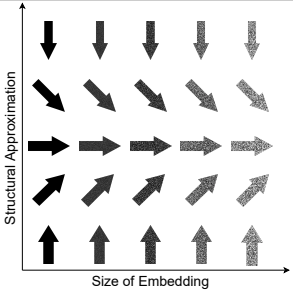


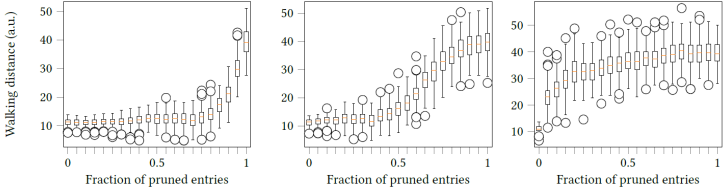


$$\min_{\vec{x}} \Big(\underbrace{\sum_{i \in I} c_{ii} x_i + \sum_{(i,j) \in K} c_{ij} x_i x_j}_{\text{Correctness}} + \underbrace{\sum_{i \in I'} c_{ii} x_i + \sum_{(i,j) \in K'} c_{ij} x_i x_j}_{\text{Optimisation}} \Big)$$

Maximise Utility

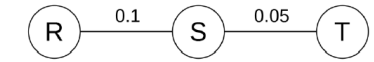
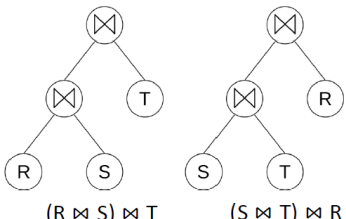






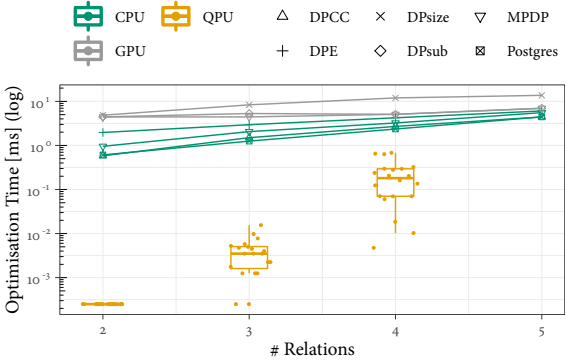
Databases

Join Ordering Overview

Input			The query graph with the relations **R**, **S** and **T** and two predicates.
Solutions			**Two possible left-deep join trees** representing different join permutations.
Resulting cost	51,000	100,000	Vastly different costs.

D-Wave Quantum Annealing

- Comparison to classical DP
- Speedups for small queries[3]
- But: Limits quickly reached



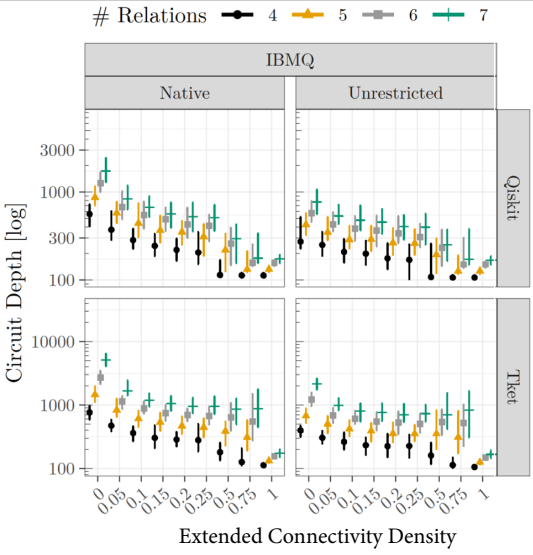
[3]Note that we consider simplified queries for the QPU, to avoid discretisation issues.

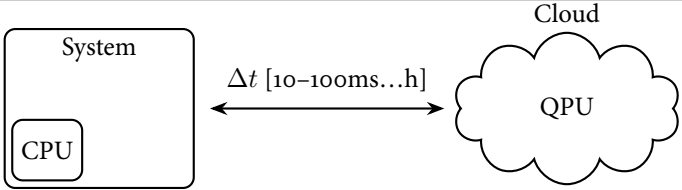
Extrapolating QPUs

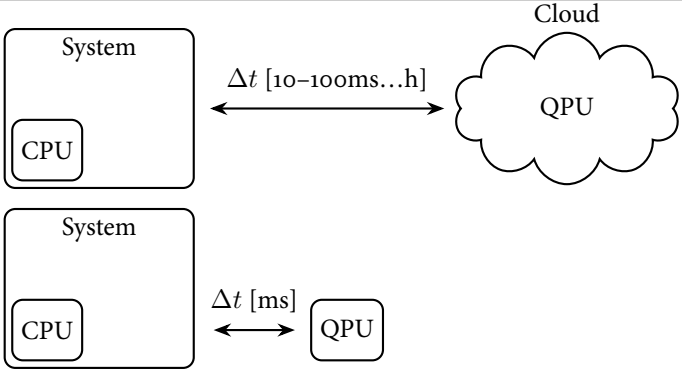
- Criteria:
  - Problem size
  - Connectivity
  - Gate sets
- Topologies:
  - IBM Q
  - Rigetti
  - IonQ

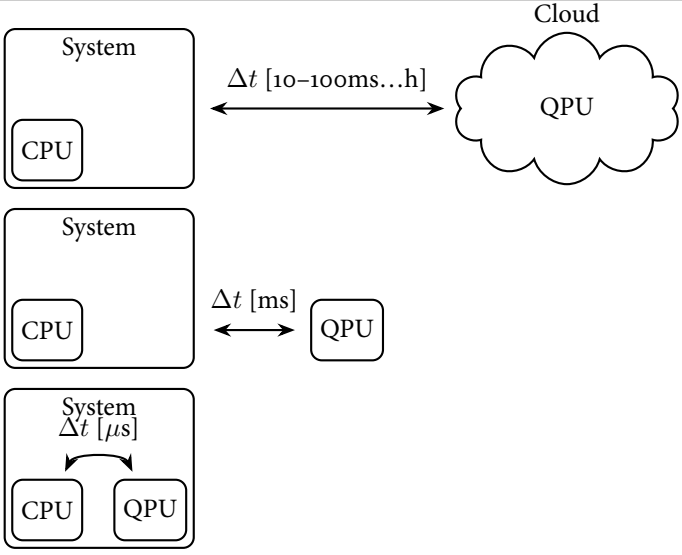
Impact on IBM Q

- Higher connectivity: Drastic impact
- Gate sets: Moderate impact

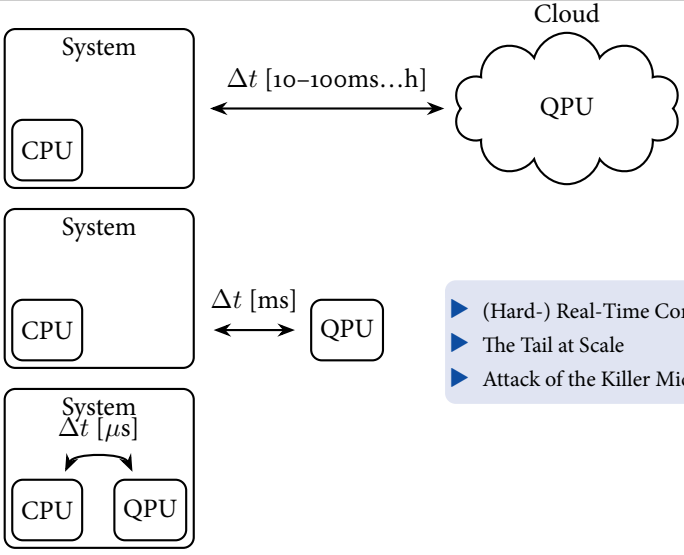



System
CPU
Δt [10–100ms…h]
Cloud
QPU

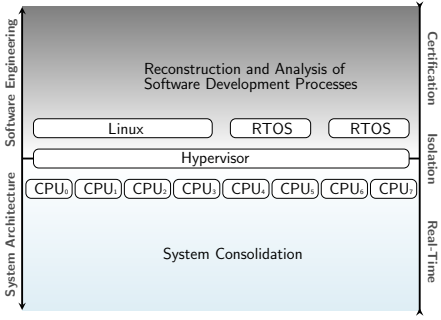

Cloud
System
CPU
Δt [10–100ms…h]
QPU
System
CPU
Δt [ms]
QPU


Cloud
System
CPU
Δt [10–100ms…h]
QPU
System
CPU
Δt [ms]
QPU
System
Δt [μs]
CPU
QPU

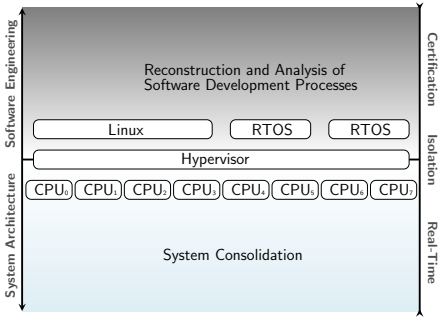
Latency and Jitter and Integration

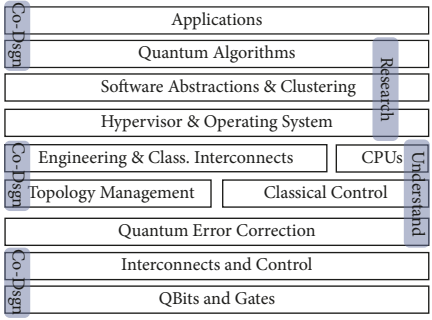


- (Hard-) Real-Time Computing
- The Tail at Scale
- Attack of the Killer Microseconds



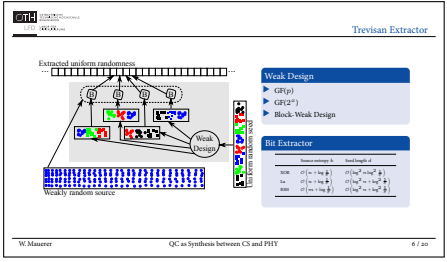
OTH Regensburg: > 300 Commits to Jailhouse@GitHub
R. Ramsauer, D. Lohmann, WM: *Look mum, no VM exits! (almost)*, Proc. 13th Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT) (2017)

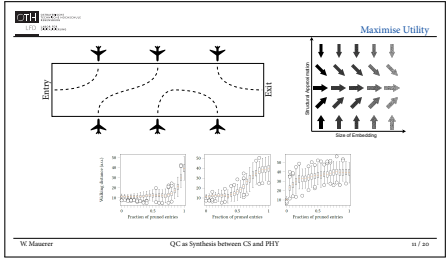


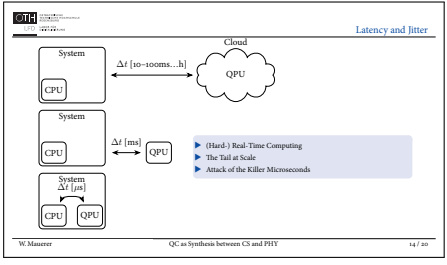


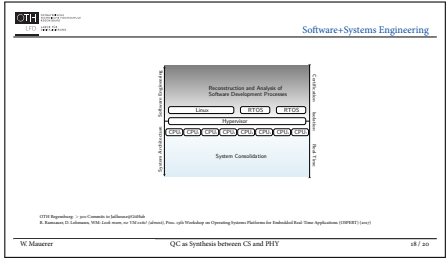
OTH Regensburg: > 300 Commits to Jailhouse@GitHub
R. Ramsauer, D. Lohmann, WM: *Look mum, no VM exits! (almost)*, Proc. 13th Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT) (2017)

Thank you!

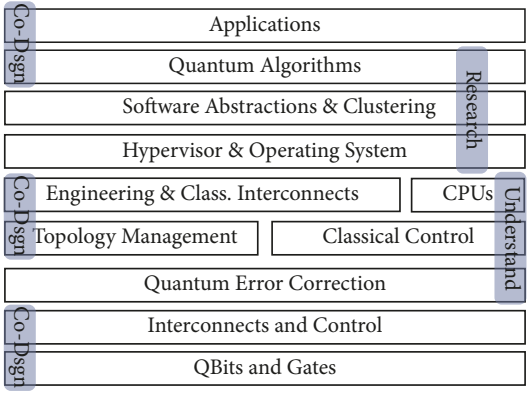

Trevisan Extractor
Extracted uniform randomness
Weak Design
Weakly random source
Uniform random word
Weak Design
GF(p)
GF(2^n)
Block-Weak Design
Bit Extractor
W. Maurerer
QC as Synthesis between CS and PHY
8 / 20

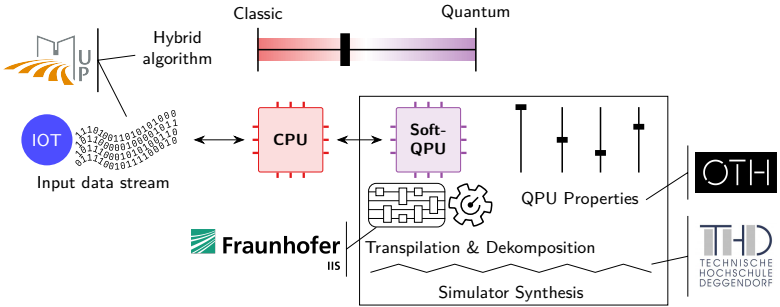

Maximise Utility
Entry
Exit
Structural Separation
Size of Embedding
Fraction of pruned entries
W. Maurerer
QC as Synthesis between CS and PHY
11 / 20


Latency and Jitter
Cloud
System
CPU
QPU
Δt [10–100ms…h]
Δt [ms]
(Hard-) Real-Time Computing
The Tail at Scale
Attack of the Killer Microseconds
System Δt [μs]
W. Maurerer
QC as Synthesis between CS and PHY
14 / 20

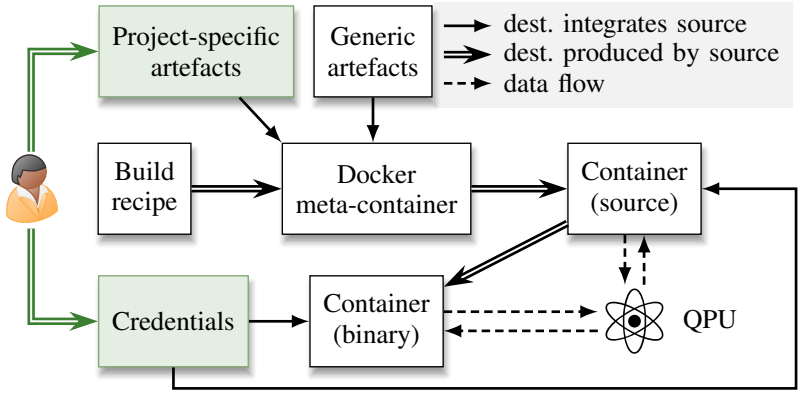

Software+Systems Engineering
Reconstruction and Analysis of Software Development Processes
Linux
RTOS
RTOS
Hypervisor
CPU0
CPU1
CPU2
CPU3
CPU4
CPU5
CPU6
CPU7
System Consolidation
Software Engineering
System Architecture
Config-ware
Isolation
Real-Time
W. Maurerer
QC as Synthesis between CS and PHY
18 / 20

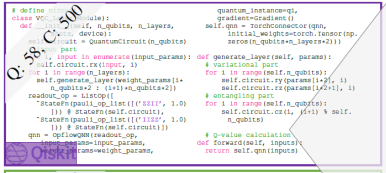
Appendix/Backup


Co-Dsgn
Applications
Co-Dsgn
Quantum Algorithms
Research
Software Abstractions & Clustering
Hypervisor & Operating System
Co-Dsgn
Engineering & Class. Interconnects
CPUs
Understand
Co-Dsgn
Topology Management
Classical Control
Understand
Quantum Error Correction
Co-Dsgn
Interconnects and Control
Co-Dsgn
QBits and Gates


Classic
Quantum
Hybrid algorithm
IOT
Input data stream
CPU
Soft-QPU
QPU Properties
Fraunhofer IIS
Transpilation & Dekomposition
Simulator Synthesis
OTH
THD TECHNISCHE HOCHSCHULE DEGGENDORF

Quantum Reproducibility





Quantum Deep Q-Learning

Init replay buf $\mathbb{D}$, (target) VQC $\theta^{(-)}$
for $s \leftarrow 0$, steps do
  Sample $\mathbb{B} =$
    $(s_t, a_t, s_{t+1}, r_{t+1})\backslash\mathbb{D}$
  for all $b_i \in \mathbb{B}$ **do**
    ▷ Target
    $y_i \leftarrow \gamma \max_{a'} Q(s_{i+1}, a'; \theta^{-})$
    $q_i \leftarrow Q(s_i, a_i; \theta)$ ▷ Q-Value
  end for
  $L(\theta) \leftarrow (y - q)^2$ ▷ Loss
  Update θ param shift rule [36], [47])
  if $s \bmod \text{update} = 0$ **then**
    $\theta^{-} \leftarrow \theta$
  end if
end for

Multi Query Optimization

Init MQO QUBO qm, circ. params β, γ
$c_{\text{lin}}, c_{\text{qdr}} \leftarrow \text{IsingCoeffs}(qm)$
$hl \leftarrow \{\}$
for $k \leftarrow 0, p$ **do**
  $hl \leftarrow hl \frown \text{CostHam}(c_{\text{lin}}, c_{\text{qdr}})$
  $hl \leftarrow hl \frown \text{MixerHam}()$
end for
$qc \leftarrow \text{buildQCircuit}(hl, \beta, \gamma)$
Initialize classical optimizer opt
while ¬converged **do**
  $\beta, \gamma \leftarrow opt.\text{step}(qc, \beta, \gamma)$
end while
$r \leftarrow \text{sample}(qc, \beta, \gamma)$