Natural Language Processing with Intel® Quantum Simulator

Myles Doyle (ICHEC - myles.doyle@ichec.ie),
Lee J. O'Riordan (ICHEC),
Venkatesh Kannan (ICHEC)
Fabio Baruffa (Intel)

Project context

- Collaboration between Irish Centre for High-End Computing (ICHEC) and Intel®
- Project timeline: January 2019 - March 2020 (14 months)
- Funded by the Irish state (through Enterprise Ireland) and Intel®
- Funding amount: ~€150k

Project overview

- Using Intel® Quantum Simulator to implement quantum-enabled natural language processing (NLP) algorithms to determine sentence meanings

- Distributional Compositional Semantics (DisCo) model [1,2,3]
  - NLP algorithms to compute meanings of two sentences and decide if their meanings match
  - Incorporates grammatical structure of sentences in a language into the analysis algorithms
  - "Closest vector problem" quantum algorithm
  - "Sentence similarity" quantum algorithm

- Quantum advantage in memory requirement
  - Example: word-meaning space of a corpus, if based on 2000 most common words
    - One transitive verb: 1 GB in classical, 33 qubits in quantum
    - 10K transitive verbs: 10 TB in classical, 47 qubits in quantum

[1]. William Zeng and Bob Coecke, "Quantum Algorithms for Compositional Natural Language Processing", Proceedings of SLPCS, 2016.

[2]. Stephen Clark, Bob Coecke and Mehrnoosh Sadrzadeh, "A Compositional Distributional Model of Meaning", Proceedings of 2nd Quantum Interaction Symposium, 2008.

[3]. Bob Coecke, Mehrnoosh Sadrzadeh and Stephen Clark, "Mathematical Foundations of a Compositional Distributional Model of Meaning", Special issue of Linguistic Analysis, 2010.

Project overview

- DisCo model
  - Sentence meaning determined by combining word adjacency in text corpus + known meanings of component (basis) words
  - Sentences are represented as tensor products of each individual word (nouns, verbs, etc)
  - Use graphical notation from category theory to represent sentence meanings
  - Convert graphical notation directly to quantum mechanical representation

- Implement closest vector quantum algorithm between DisCo-modelled sentences
  - Use Hamming distance to measure closeness of vectors
  - Sentence similarity quantum algorithm naturally extends from above approach

DisCo

- Example sentence structure: noun-verb-noun
  "*John eats cake*", "*Mary walks outside*".

$$\text{John} \in \mathcal{N} \qquad \text{eats} \in \mathcal{N} \otimes \mathcal{S} \otimes \mathcal{N}$$

- Entire meaning space given by tensor product:

$$\mathcal{N} \otimes (\mathcal{N} \otimes \mathcal{S} \otimes \mathcal{N}) \otimes \mathcal{N}$$

Example representation

"John eats cake, Mary swims in water"

$$C^{-1} \left|\text{subject}\right\rangle \otimes \left|\text{verb}\right\rangle \otimes \left|\text{object}\right\rangle$$

$$C^{-1} \left(\sum_i |s_i\rangle \right) \otimes \left(\sum_j |v_j\rangle \right) \otimes \left(\sum_k |o_k\rangle \right)$$

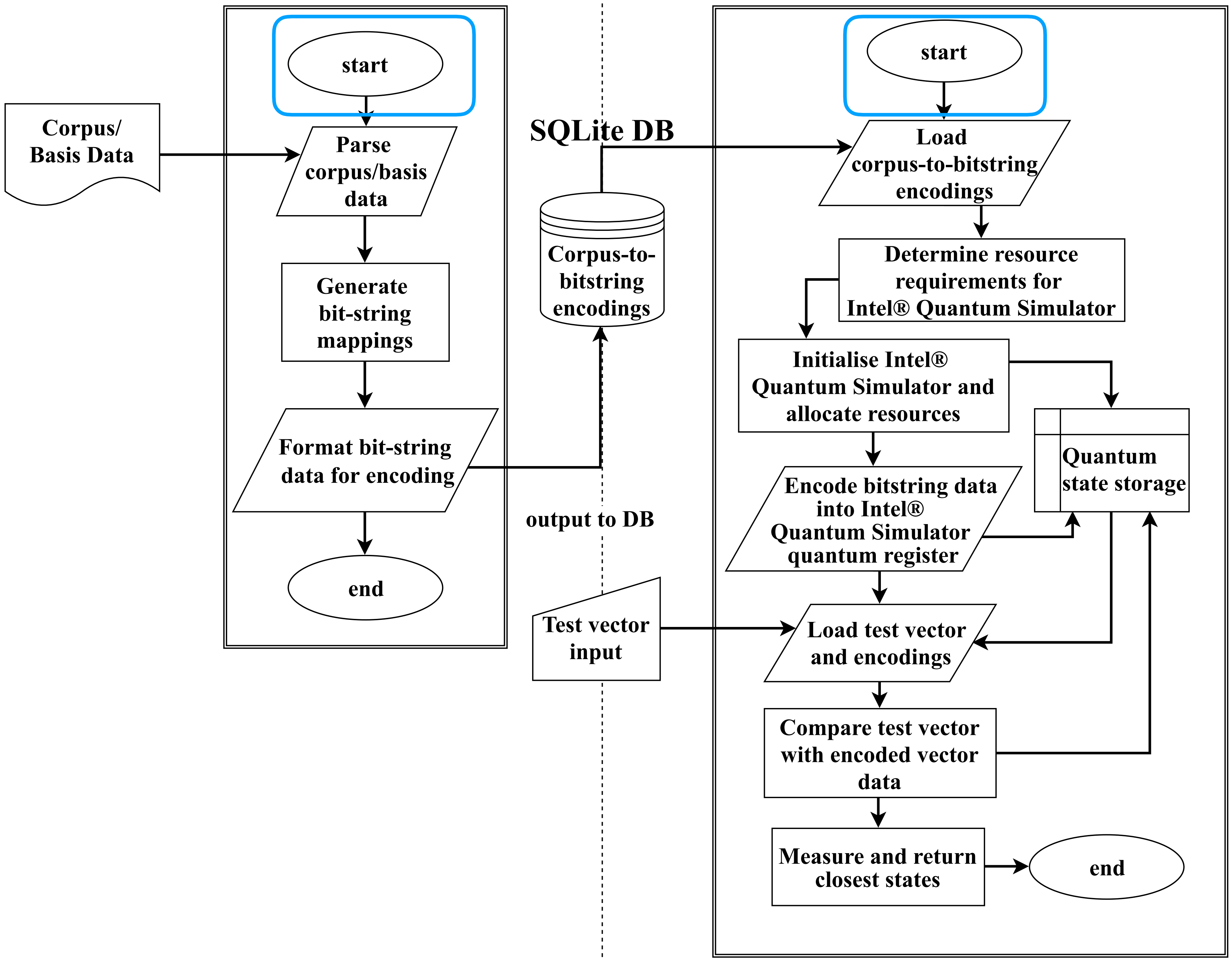
$\|000\rangle \rightarrow$	John eats water	$\|001\rangle \rightarrow$	John eats cake
$\|010\rangle \rightarrow$	John swims water	$\|011\rangle \rightarrow$	John swims cake
$\|100\rangle \rightarrow$	Mary eats water	$\|101\rangle \rightarrow$	Mary eats cake
$\|110\rangle \rightarrow$	Mary swims water	$\|111\rangle \rightarrow$	Mary swims cake

$$q_0 = \{\text{John}, \text{Mary}\}$$
$$q_1 = \{\text{eats}, \text{swims}\}$$
$$q_2 = \{\text{water}, \text{cake}\}$$

$$\begin{pmatrix} \text{John} \\ \text{Mary} \end{pmatrix} \otimes \begin{pmatrix} \text{eats} \\ \text{swims} \end{pmatrix} \otimes \begin{pmatrix} \text{water} \\ \text{cake} \end{pmatrix}$$

Python layer

C++ (Intel® Quantum Simulator) Layer



Comparing sentences

Basis representation

Dataset	Word	Bin. Index
n_s	adult	00
n_s	child	01
n_s	smith	10
n_s	surgeon	11
v	stand	00
v	move	01
v	sit	10
v	sleep	11
n_o	inside	0
n_o	outside	1

Corpus quantum state

Dataset	Word	State
n_s	John	$(\|00\rangle + \|10\rangle)/\sqrt{2}$
n_s	Mary	$(\|01\rangle + \|11\rangle)/\sqrt{2}$
v	walk	$(\|00\rangle + \|01\rangle)/\sqrt{2}$
v	rest	$(\|10\rangle + \|11\rangle)/\sqrt{2}$
n_o	inside	$\|0\rangle$
n_o	outside	$\|1\rangle$

$$\text{Data} = \text{"John rests outside, and Mary walks inside"}$$

$$\text{Test} = \text{"Adults stand inside"}$$

$$\textbf{John rests outside} \rightarrow \frac{1}{2}(|00100\rangle + |00110\rangle + |10100\rangle + |10110\rangle) \rightarrow |J\rangle$$

$$\textbf{Mary walks inside} \rightarrow \frac{1}{2}(|01001\rangle + |01011\rangle + |11001\rangle + |11011\rangle) \rightarrow |M\rangle$$

Comparing sentences

Basis representation

Dataset	Word	Bin. Index
n_s	adult	00
n_s	child	01
n_s	smith	10
n_s	surgeon	11
v	stand	00
v	move	01
v	sit	10
v	sleep	11
n_o	inside	0
n_o	outside	1

Data = "John rests outside, and Mary walks inside"

Test = "Adults stand inside"

$$|Q\rangle = \frac{1}{\sqrt{8}}(|00000\rangle|00100\rangle + |00000\rangle|00110\rangle + |00000\rangle|10100\rangle + |00000\rangle|10110\rangle + |00000\rangle|01001\rangle + |00000\rangle|01011\rangle + |00000\rangle|11001\rangle + |00000\rangle|11011\rangle)$$

$$|Q\rangle = a_1|00000\rangle|00100\rangle + a_2|00000\rangle|00110\rangle + a_3|00000\rangle|10100\rangle + a_4|00000\rangle|10110\rangle + a_5|00000\rangle|01001\rangle + a_6|00000\rangle|01011\rangle + a_7|00000\rangle|11001\rangle + a_8|00000\rangle|11011\rangle$$

Corpus quantum state

Dataset	Word	State
n_s	John	$(\vert 00\rangle + \vert 10\rangle)/\sqrt{2}$
n_s	Mary	$(\vert 01\rangle + \vert 11\rangle)/\sqrt{2}$
v	walk	$(\vert 00\rangle + \vert 01\rangle)/\sqrt{2}$
v	rest	$(\vert 10\rangle + \vert 11\rangle)/\sqrt{2}$
n_o	inside	$\vert 0\rangle$
n_o	outside	$\vert 1\rangle$

"Adults stand inside" $\longrightarrow$ "Adult(s) sit inside"

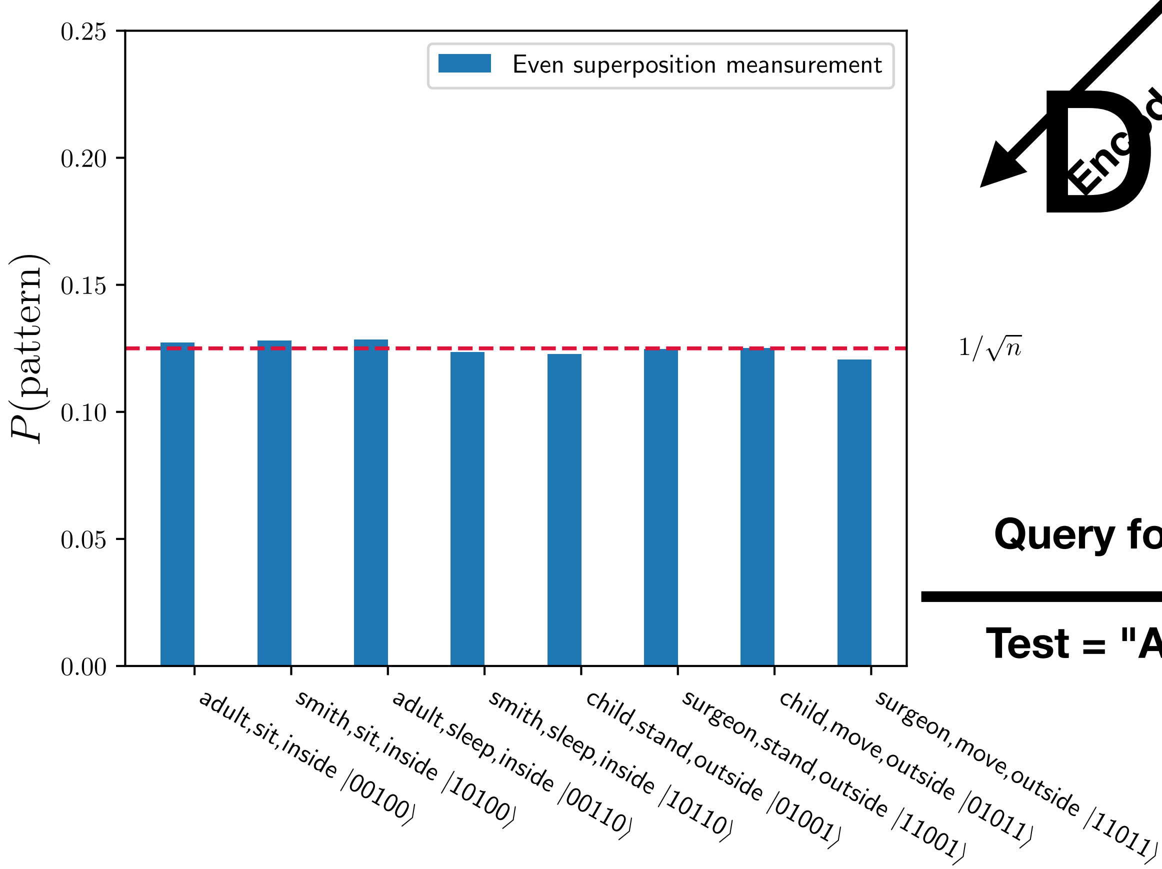
adult, sit, inside
smith , sit, inside
adult, sleep, inside
.
.
.

Create encodings →

$|00100\rangle$
$|10100\rangle$
$|00110\rangle$
.
.
.

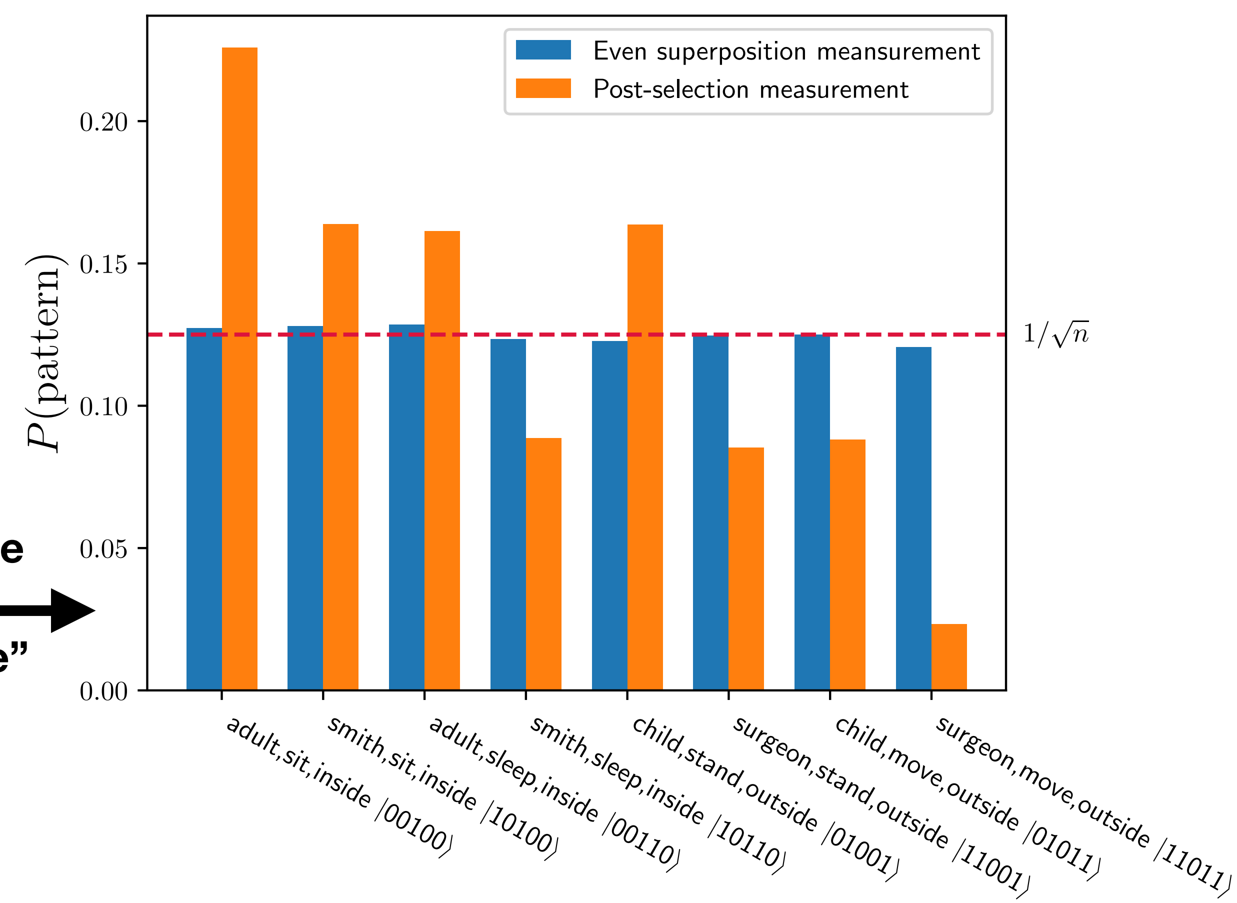
Encode into quantum state

Demo



Query for closest sentence →

Test = "Adults stand inside"
$|00000\rangle$



Irish National Supercomputer: Kay

- 336 node cluster
- 13,440 CPU cores, 63 TB distributed memory
- Dual-socket 20-core Intel® Xeon Gold (Skylake) 6148 at 2.4 GHz with 192 GB memory
- 400 GB local SSD scratch
- 100 GB Intel® OmniPath network
- Additional partitions
    - Dual NVIDIA Tesla V100
    - Intel Xeon Phi (Knights Landing architecture)
    - High-memory 1.5TB RAM with 1TB local SSD scratch

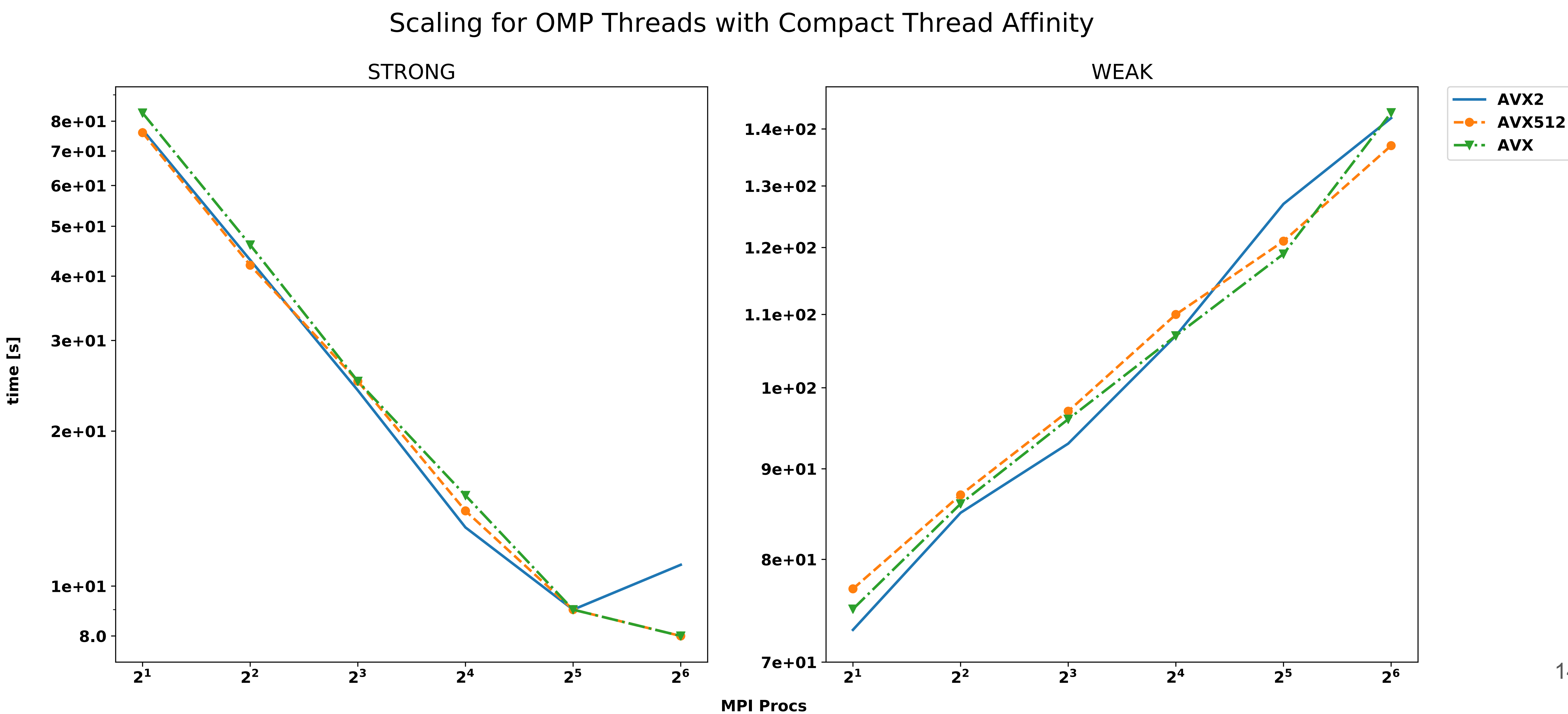
Intel® Quantum Simulator

- High performance quantum simulator
- Single and two qubit quantum gates
    - Rotation, Hadamard, Pauli operators, Square root of Pauli, Toffoli, SWAP, Square root of SWAP
- Controlled gates
- Single and double precision qubit registers
- Measurement routines
- Observe amplitudes of each state

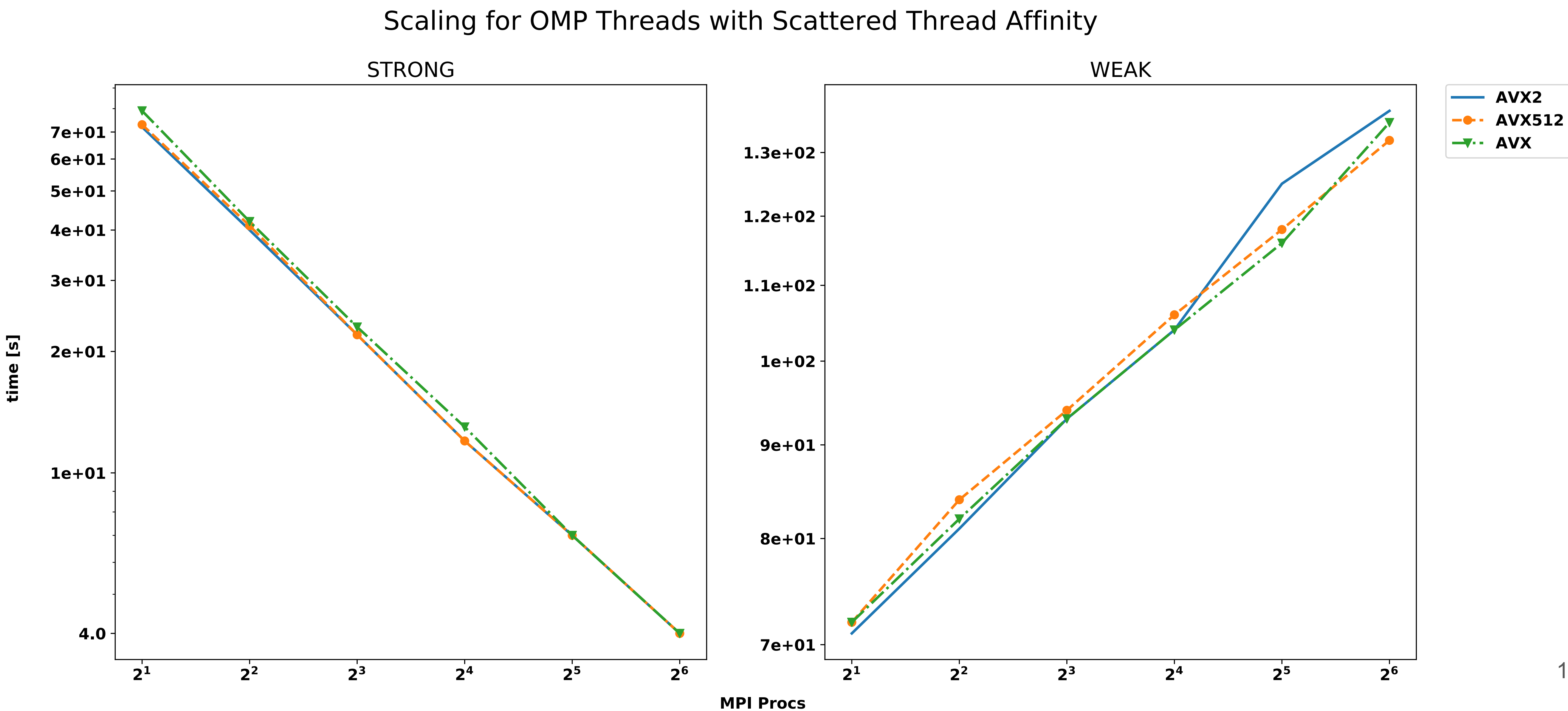
Intel® Quantum Simulator: Optimisations

- Vectorisation
  - Loops to implement single and controlled gate operations are data parallel
  - SIMD execution of gate operation loops using 4-wide AVX-512 instruction set
- Multi-threading
  - Loops to implement single and controlled gate operations are 2-level nested
  - Dynamic check of workload to decide nesting level at which to parallelise using multi-threading
- Communication
  - Gate operations reuse temporary storage for qubit simulation to avoid performance impact from paging
- Gate Fusion
  - Limited computation by single and controlled qubit operations
  - Huge performance loss due to memory bandwidth when leaving Last Level Cache (LLC)
  - Gate fusion used to block computations in LLC for single and controlled qubit operations

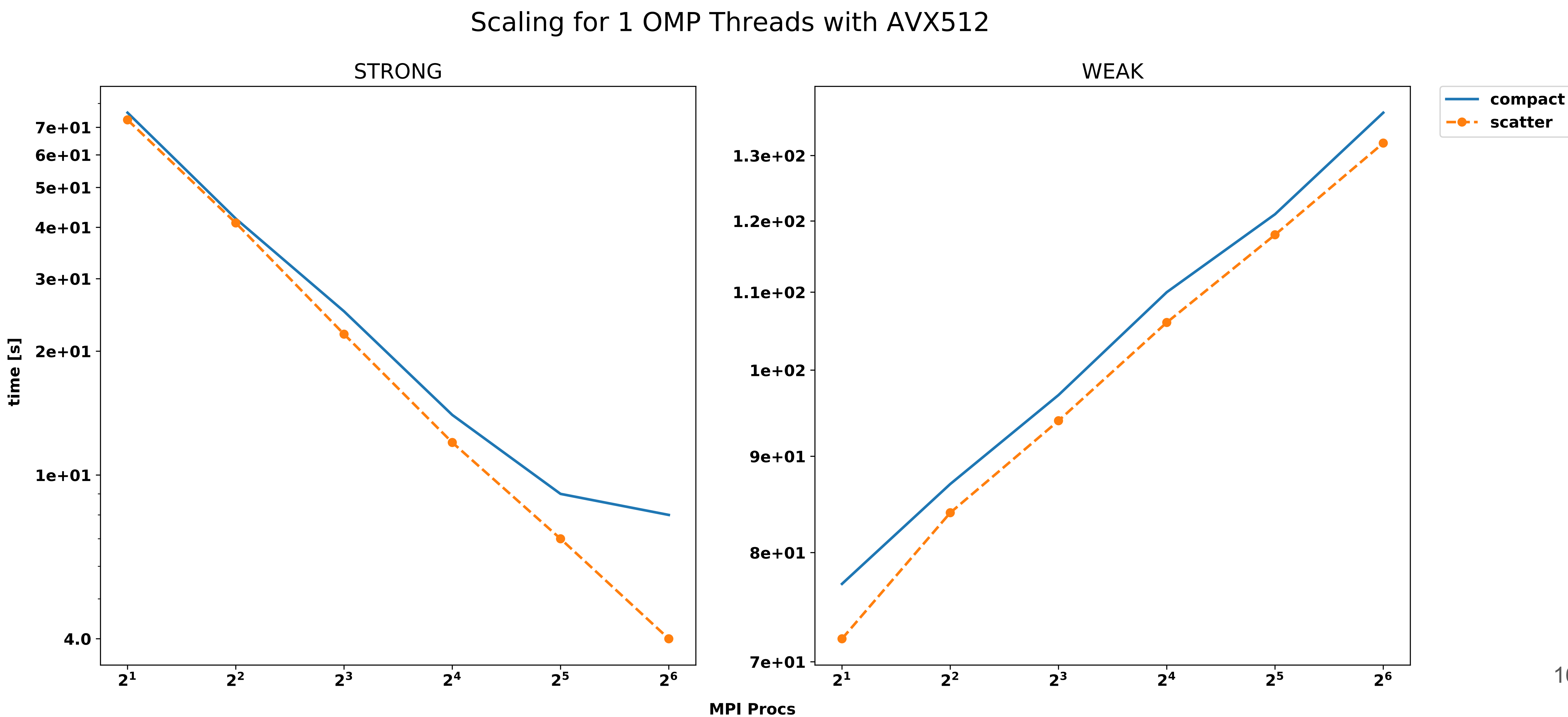
Performance: Vectorisation



Performance: Vectorisation



Performance: Thread Affinity



Summary

- Comparing sentences using DisCo algorithm
- Natural extension to quantum states
- Promising candidate for NLP methods
  on quantum computers
- Implemented on top of Intel® Quantum Simulator
- Intel® Quantum Simulator
- Performance on Kay
- **Questions?**

~