PATH PLANNING WITH LOIHI

Neuromorphic Computing Lab | Intel Labs

Nengo Summer School 2019

Rev. 0.5

LEGAL INFORMATION

This presentation contains the general insights and opinions of Intel Corporation (“Intel”). The information in this presentation is provided for information only and is not to be relied upon for any other purpose than educational. Intel makes no representations or warranties regarding the accuracy or completeness of the information in this presentation. Intel accepts no duty to update this presentation based on more current information. Intel is not liable for any damages, direct or indirect, consequential or otherwise, that may arise, directly or indirectly, from the use or misuse of the information in this presentation.

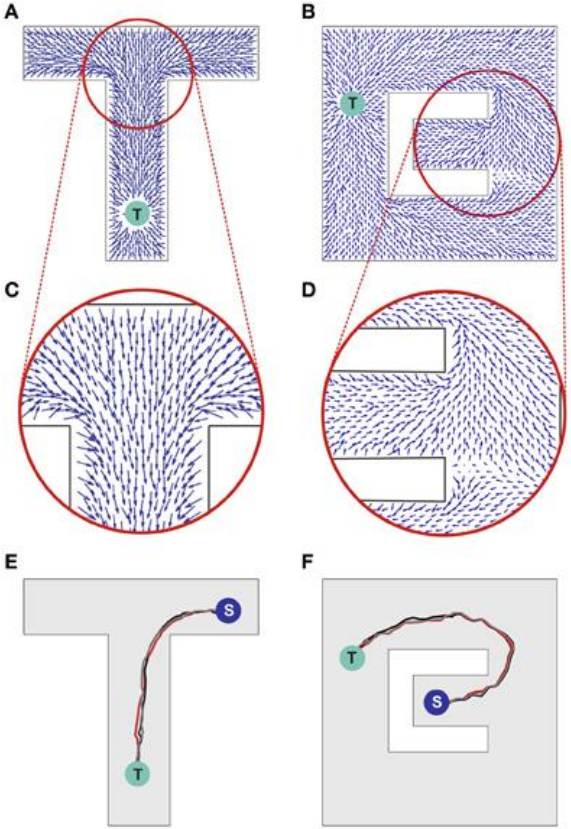
Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure. No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document. Intel, the Intel logo, Movidius, Core, and Xeon are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others

Copyright © 2019 Intel Corporation.

Path Planning with Spike Wavefronts



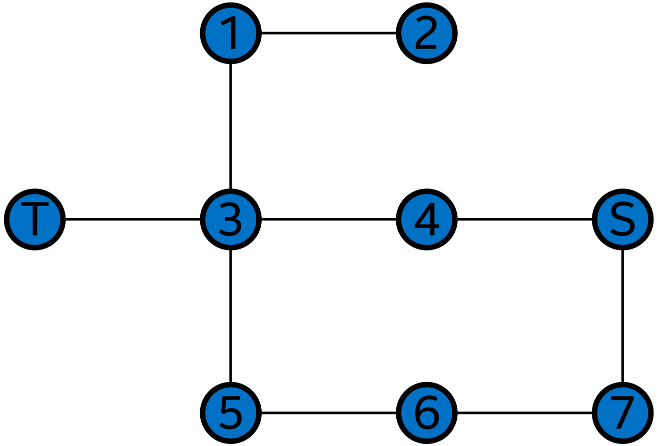
Hopfield[1] modeled how the brain might solve spatial navigation problems using parallel exploration of alternative routes through propagating waves of spiking activity

[1] Ponulak F., Hopfield J.J. Rapid, parallel path planning by propagating wavefronts of spiking neural activity. Front. Comput. Neurosci. 2013. V. 7. Article № e98.

Optimized (exact) spiking graph search algorithm (1)

Initial state:

- All edges have bidirectional synaptic connections



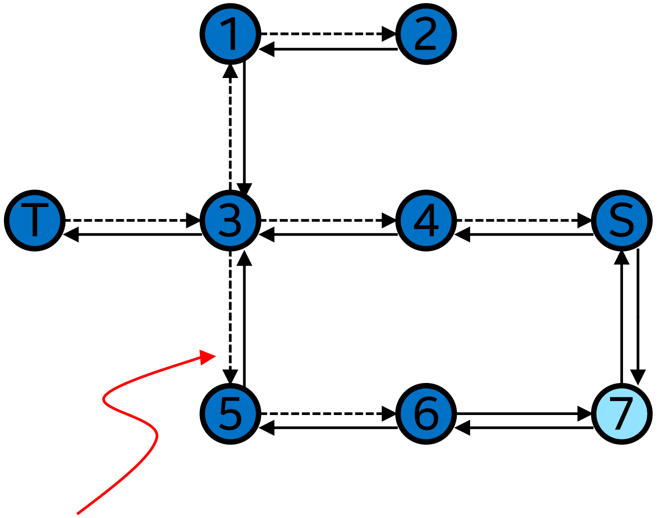
Optimized (exact) spiking graph search algorithm (2)

Phase 1 wavefront propagation:

- Spike wavefront propagates from Target (T) to Source (S)
- **First arriving spike** at each node causes **input weight to be zeroed**

End state:

- Zeroed edges form a spanning tree over all nodes between T and S within diam(S,T).
- Non-zeroed edges point in direction of shortest path back to T.



Wavefront arrives first from 3->5, so weight is zeroed

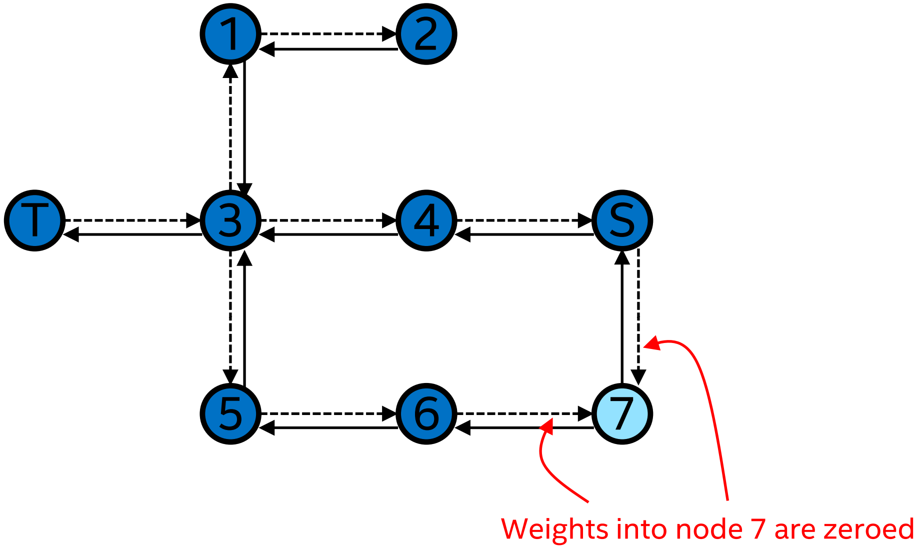
Optimized (exact) spiking graph search algorithm (3)

Phase 1 cleanup:

- After S fires, **weights into unfired nodes are zeroed**

End state:

- S only has one nonzero fanout edge

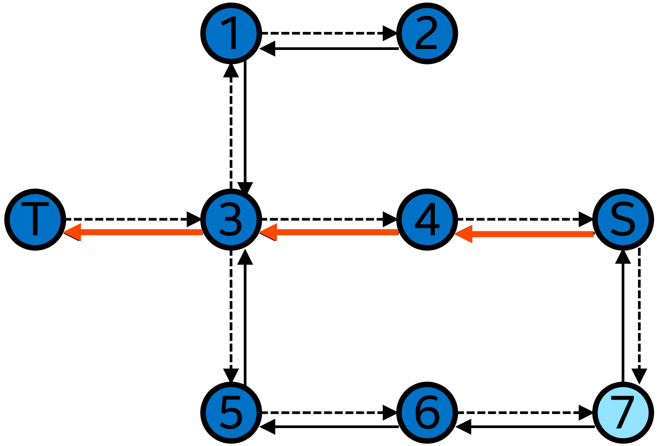


Optimized (exact) spiking graph search algorithm (4)

Phase 2:

- Trace path with non-zero weights from S->T to read out shortest path

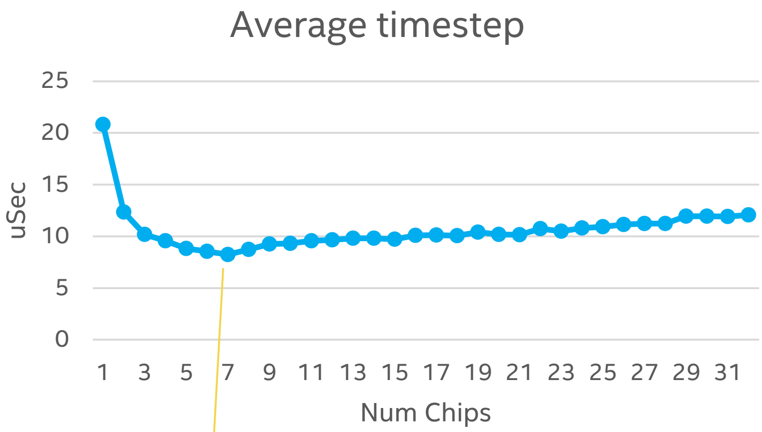
(∃ only *one path* S->T due to graph's spanning tree structure after phase 1)



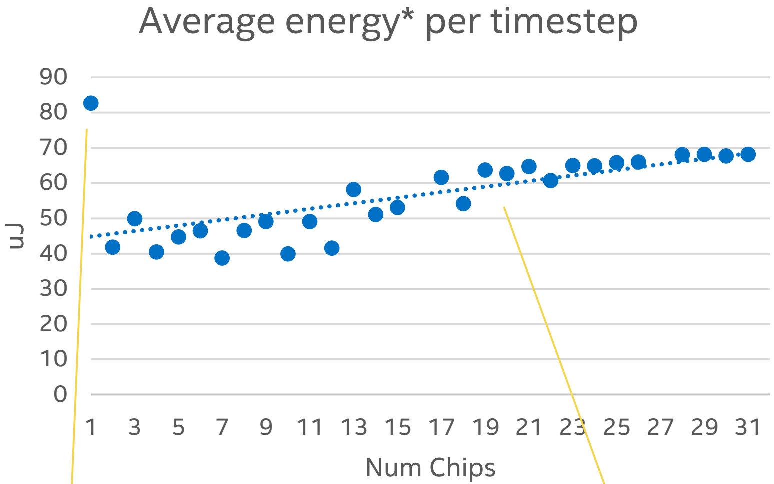
Tutorial: Path Planning

*

Distributing the 50x50x50 lattice over Nahuku



Search performance improves from 1 to 7 chips, degrading thereafter as chip-to-chip synchronization and spike communication overhead dominate.

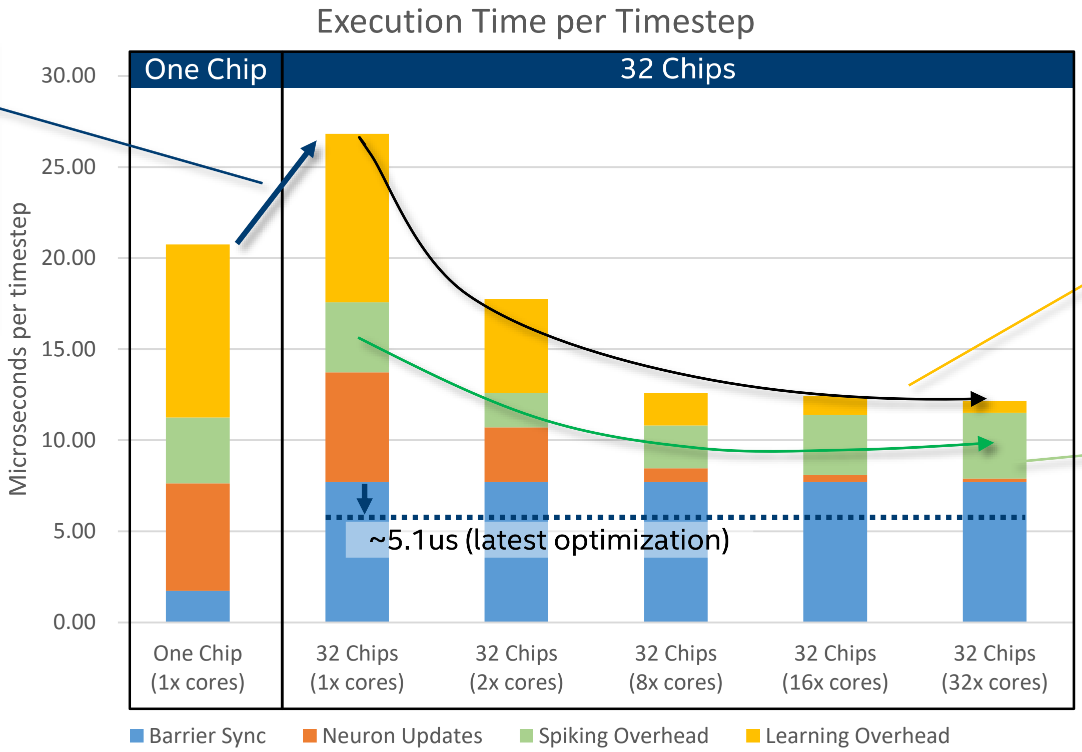


Single chip case has *worst* energy efficiency due to leakage

Energy efficiency degrades with increasing parallelism due to increasing chip-to-chip communication cost

*Includes energy due to board-level leakage/idle power

Increasing **core parallelism** with **fixed chip count**



Fixed 128-way core parallelism. Slowdown due to **increased barrier sync time** over 32 chips vs 1 chip

Learning overhead **decreases** with increasing core parallelism

Spike overhead decreases, then **increases** with increasing core parallelism