A practical introduction to programming Tenstorrent accelerators

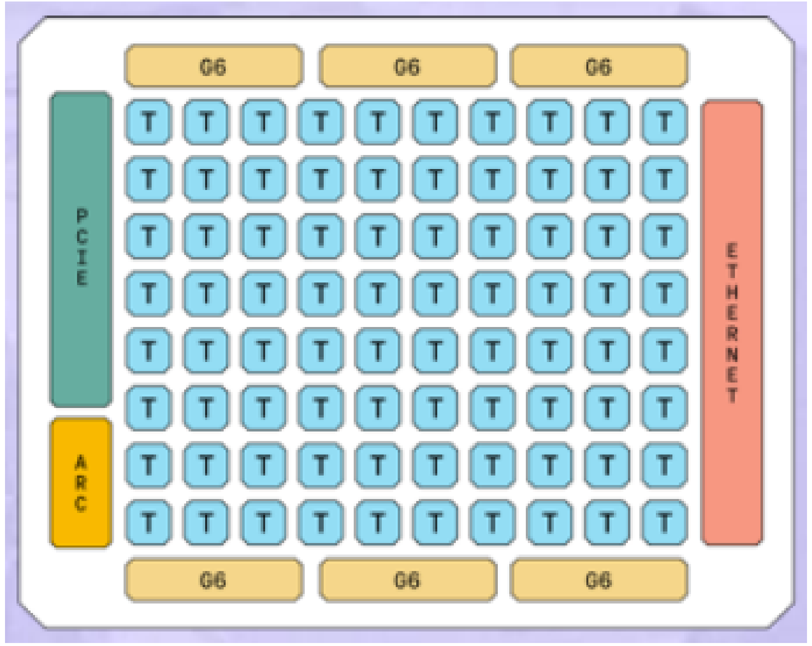
Nick Brown
EPCC University of Edinburgh

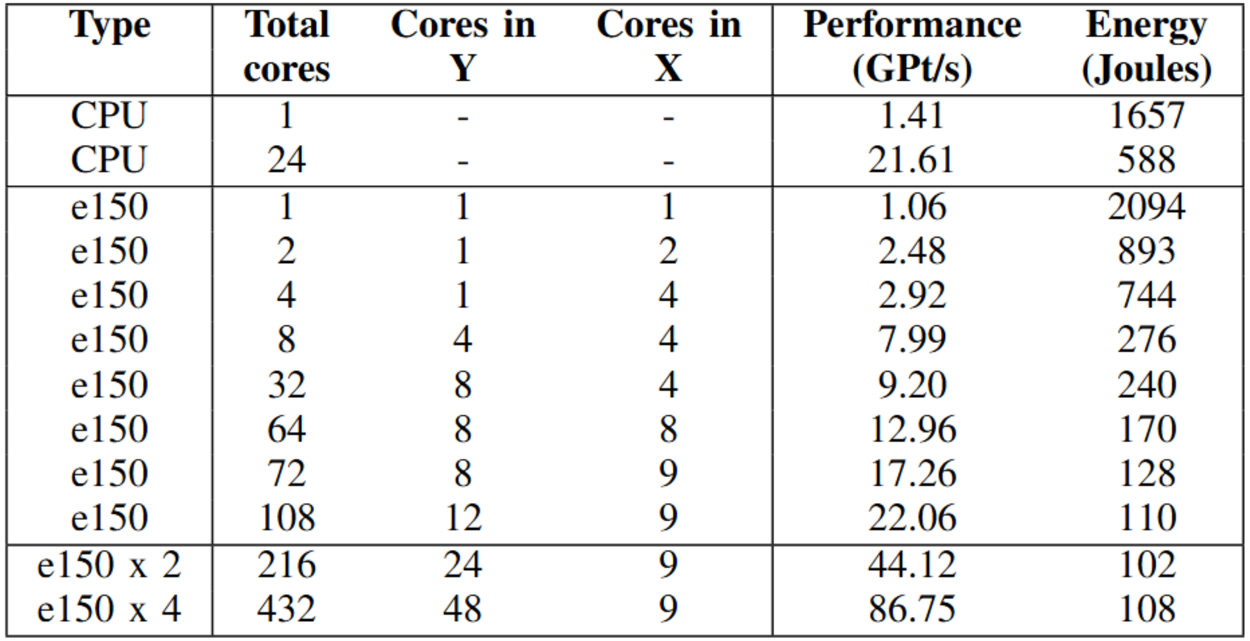
Felix Le Clair
Tenstorrent

Jake Davies
EPCC University of Edinburgh

Motivation

- There is increased focus on moving towards more energy efficient accelerator technologies in HPC whilst maintaining performance
  - Numerous accelerators for ML are being proposed, and some of these (such as Tenstorrent) are being made available for more general workloads





Type	Total cores	Cores in Y	Cores in X	Performance (GPt/s)	Energy (Joules)
CPU	1	-	-	1.41	1657
CPU	24	-	-	21.61	588
e150	1	1	1	1.06	2094
e150	2	1	2	2.48	893
e150	4	1	4	2.92	744
e150	8	4	4	7.99	276
e150	32	8	4	9.20	240
e150	64	8	8	12.96	170
e150	72	8	9	17.26	128
e150	108	12	9	22.06	110
e150 x 2	216	24	9	44.12	102
e150 x 4	432	48	9	86.75	108

- A lot of what you need in ML is also beneficial for HPC!
- Tenstorrent decouples the movement of data from compute, potentially helping us with memory bound workloads
- To the left is a stencil code on the Grayskull compared to a 24-core Xeon Platinum
  - Comparable performance, but five times less energy usage

We focus on the Wormhole

- The first generation was the Grayskull
  - This has been End Of Lived now
- The current generation is the Wormhole
- The next generation is the Blackhole
  - We have both Wormhole and Blackhole, using Wormhole today

- All built using the Tensix architecture

Tutorial learning objectives

- This tutorial is open to everybody, regardless of experience with HPC and accelerators
  - Is practically driven, where we will walk-through key concepts on the machine itself, and then you can explore the concepts more independently via a series of walk-throughs

1. Understand the Tenstorrent architecture & core concepts
   - We will explore the hardware, how it is designed the and key terminology
2. Get started with the Tenstorrent tt-metal SDK
   - Exploring key concepts for writing codes for the Tenstorrent architecture and understanding how to build these
3. Optimising codes on Tenstorrent by using the matrix engine and vector unit
   - Throughout we will be running on real Tenstorrent hardware
4. An awareness of RISC-V and how it underlies technologies such as this

We are also happy to discuss your own applications and how these might be ported to the architecture

Session plan

Time	Title	Type
14:00 – 14:05	Introduction, welcome and objectives	Presentation
14:05 – 14:30	An Overview of the Tenstorrent architecture	Presentation
14:30 – 14:40	Logging onto the RISC-V testbed for Tenstorrent hardware	Practical
14:40 – 15:30	Introduction to the SDK & data movement (lecture and two practicals)	Presentation and practicals
15:30 – 16:00	Break	
16:00 – 16:05	Welcome back and overview of second part	Presentation
16:05 – 16:25	Overview of compute SDK	Presentation
16:25 – 17:25	Practicals three, four and five	Practicals
17:25 – 17:30	Conclusions and audience next steps to continue working with the technologies	Presentation

Materials and the Tenstorrent community

- We will remind people as we progress through the session

- All materials for this tutorial are open source and can be found at
  - https://github.com/RISCVtestbed/tt-tutorial

- More generally if you wish to continue exploring this after the tutorial finishes
  - https://docs.tenstorrent.com/

- There is a Tenstorrent developer community
  - https://tenstorrent.com/developers
  - Discord at https://discord.com/invite/tenstorrent