**Tensor Processing Primitives in the Computational Sciences: Earthquake Simulations**

A. Breuer, E. Georganas (Intel), A. Heinicke (Intel), A. Noack, K. Voronin (Intel)

---

**Earthquake Simulations**

High-frequency Ground Motion Simulations: Broad challenge in computational seismology. Challenge is overcome by new approaches to computational seismology.

1. Higher frequencies require extended numerical models, making many simulations less efficient.
2. New approaches to computational seismology are needed to overcome this limitation.

---

**Tensor Processing Primitives**

Tensor Processing Primitives (TPPs) enable a wide array of advanced data-intensive applications in computational seismology. They are used to enhance the performance of seismological simulations.

1. Tensor Processing Primitives in the Computational Sciences: are the amplitudes of the horizontal particle velocities after seven seconds of simulated time. Source: [2].

2. Edge: may require a sixteen-fold increase in computational resources. Even if models could be kept unchanged, simply doubling the number of used cores may not be sufficient to achieve the desired speed-up.

3. The three stations: obtain an interactive version of the map by accessing the source software (https://dial3343.org).

---

**Performance Portability**

Space Matrix in 3D Tensor and 2D Tensor in a Space Matrix (rpm) can use two methods:

1. Single variant simulation: Multiple simulations, each with different parameters, are performed. The simulated/realized FPP performance is provided below, including the performance portability of the approach. The results are shown in Table 1.

2. Multi-variant simulation: Multiple simulations, each with different parameters, are performed. The simulated/realized FPP performance is provided below, including the performance portability of the approach. The results are shown in Table 2.

---

**Application Performance and Scalability**

We observe that the application time increases with the peak performance numbers. The benchmark results are shown in Table 3.

Notably, we also tested with codes without element-wise TPP on various platforms with different compiler (GCC, ICC, Intel) and measured performance results on all platforms. With our TPP-based approach those vanished while leaving 1.1 - 1.5x of each fastest compiler choice.

---

**References**
