ElasWave: An Elastic-Native System for Scalable Hybrid-Parallel Training

Xueze Kang[1†], Guangyu Xiang[1†], Yuxin Wang[4*], Hao Zhang[4], Yuchu Fang[4], Yuhang Zhou[3], Zhenheng Tang[2], Youhui Lv[5], Eliran Maman[7], Mark Wasserman[7], Alon Zameret[7], Zhipeng Bian[6], Shushu Chen[5], Zhiyou Yu[5], Jin Wang[5], Xiaoyu Wu[4], Yang Zheng[4], Chen Tian[3], Xiaowen Chu[1*]

[1]*HKUST(GZ)* [2]*HKUST* [3]*NJU*

Huawei [4]*TTE Lab* [5]*ICT BG* [6]*Cloud* [7]*TRC Team*

Large-scale LLM pretraining is rapidly moving to the 10^5–10^6 accelerator scale, where failures are no longer exceptional events but part of normal operation. In this setting, efficient and elastic recovery is not just a backup feature; it is becoming a core systems requirement at production-scale. In this talk, we will briefly discuss the path toward an industrially deployable elastic-native training stack. We will focus on the design intuition behind ElasWave , the key mechanisms that make per-step elastic recovery practical, and the current results from our prototype.

We start building ElasWave from a simple observation: practical elastic training must jointly preserve parameter consistency and computation consistency, while also keeping recovery time (MTTR) low and sustaining high throughput after topology changes. Existing systems typically optimize only a subset of these goals. ElasWave addresses these challenges with a per-step fault-tolerant design that treats elasticity as multi-dimensional scheduling across graph, dataflow, DVFS, and RNG. It preserves training semantics while adapting execution online through micro-batch/pipeline resharding, asynchronous parameter migration, optimizer-partitioned recovery, DVFS-based bubble absorption, and RNG resharding for computation consistency, supported by a dynamic communicator and per-step in-memory snapshots for in-place group edits, verification, and redistribution.

Preliminary results on 96 NPUs show that ElasWave is promising relative to state-of-the-art baselines, with improvements in throughput, recovery efficiency, and training stability. As a work in progress, we are improving the system towards an industrially deployable stack.

[†]Equal contribution.

[*]Corresponding authors: yuxin.wang11@huawei.com, xwchu@hkust-gz.edu.cn.