UVCC Phase 6: Private, Verifiable, Fully-Parallel GPU Computation across Untrusted Domains

Alien Team
research@uvcc.example

Abstract—We present UVCC (Universal Verifiable Confidential Computing), a system that enables private and verifiable GPU computation across mutually distrustful domains. UVCC combines three-party replicated secret sharing (RSS) for confidentiality, a deterministic transcript-of-transcripts construction for verifiability, and native ML parallelism (DP/PP/TP) realized in a C++ GPU runtime with NCCL-based collectives and an exactly-once transport. We report a complete Phase 6 bring-up, diagnose and fix a PP deadlock, and demonstrate a large-scale run (R=8, S=4, T=2, M=32; 192 workers) with a determinism proof via matching global roots across reruns. We release a comprehensive, auditor-friendly log bundle along with scripts to derive metrics and figures directly from raw logs.

I. Introduction

Confidential and verifiable ML across providers requires both a privacy contract and a verifiable execution story without sacrificing parallel performance. UVCC delivers this by:

- Confidentiality via three-party RSS (honest-majority).
- Verifiability via deterministic transcripts, subsession/replica/global roots, and an audit bundle.
- Native parallelism: DP, PP, TP on GPUs with robust NCCL initialization across heterogeneous infrastructure.

II. Model and Guarantees

Adversary corrupts at most one party. Secrets include inputs, weights, gradients, optimizer state, and intermediate values. Verifiability binds all protocol events into deterministic roots with canonical ordering. The audit bundle encodes subsession, replica, and global roots and is verified against identities and policy.

III. System Overview

UVCC consists of:

- A reliable transport (exactly-once frame acceptance with conservative retries and long-polling).
- A transcript store (deterministic hashing, versioned leaf prefixes, Merkle roots).
- A native runtime integrating OPEN/LIFT engines and NCCL collectives for DP/PP/TP.

The Phase 6 worker executes a 1F1B-style schedule with careful PP/DP/NCCL ordering to avoid deadlocks.

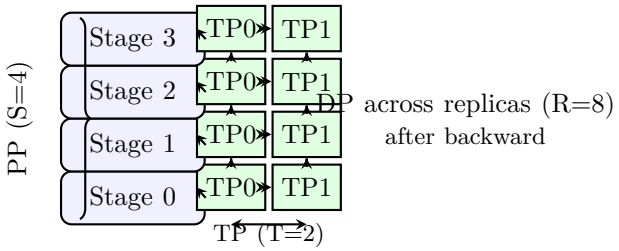

Fig. 1. Native parallelism: PP across stages, TP within stage, DP across replicas.

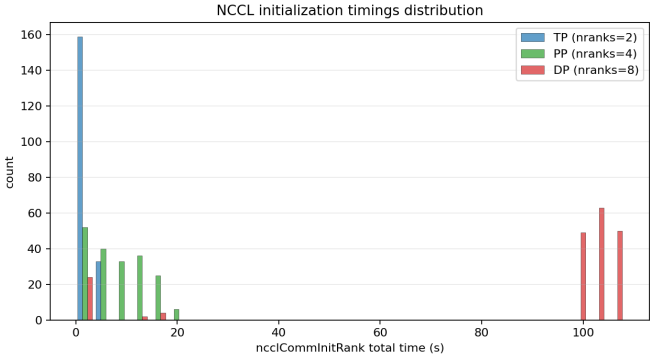

Fig. 2. NCCL init time distribution by nranks (TP=2, PP=4, DP=8).

IV. Implementation Highlights

Transport. Exactly-once receive with retransmit/backoff; long timeouts and long-polling reduce overload and tolerate party skew. NCCL. Robust UID distribution and timeout control via –phase6-timeout-s; NCCL Socket NET across VMs; deterministic group construction. Scheduler. PP gradient recvs posted per microbatch to avoid single-stream deadlock; DP init delayed until after backward to avoid OPEN skew.

V. Evaluation

A. Setup

We launch on 24 pods spanning providers. Final run topology: R=8, S=4, T=2, M=32 (192 workers). Logs are consolidated into a single explained file; our scripts derive metrics and figures from those logs for reproducibility.

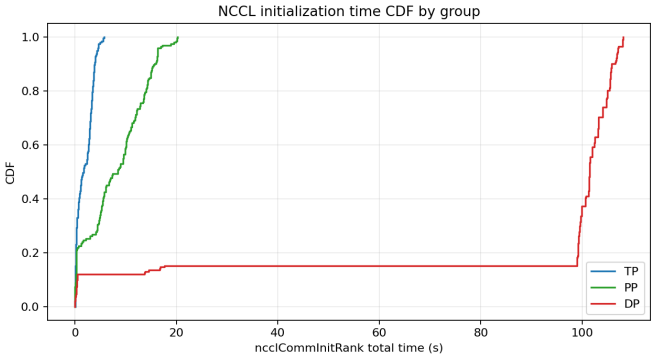

Fig. 3. NCCL init time CDF by group type (TP/PP/DP).

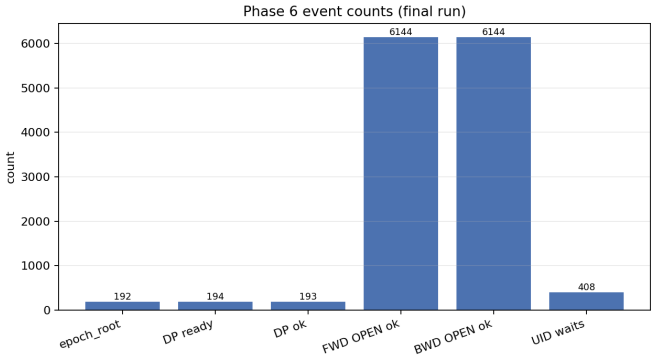

Fig. 4. Phase 6 event counts (epoch roots, OPEN completions, DP readiness).

B. NCCL Initialization Metrics

C. Protocol Event Coverage

D. Robustness Indicators

E. Determinism Proof

We ran the full job twice with fixed sid_job_hex; the resulting global_root_hex values matched exactly, demonstrating determinism under cross-provider skew.

VI. Related Work

We build on MPC, verifiable computation, and large-scale ML parallelism; UVCC's novelty is an auditor-grade verifiable MPC runtime with native DP/PP/TP across domains.

VII. Conclusion

UVCC Phase 6 demonstrates that private, verifiable, fully-parallel GPU computation across untrusted domains is practical and deterministic at scale. Future work: GPU-accelerated preprocessing (TCF/W-VOLE), production SKS, and larger model pipelines.

Artifacts and Reproducibility

Figures are generated by research/uvcc_native/scripts/make_phase6_figs.py from the consolidated log: research/uvcc_native/out-.../all_logs_explained.md.

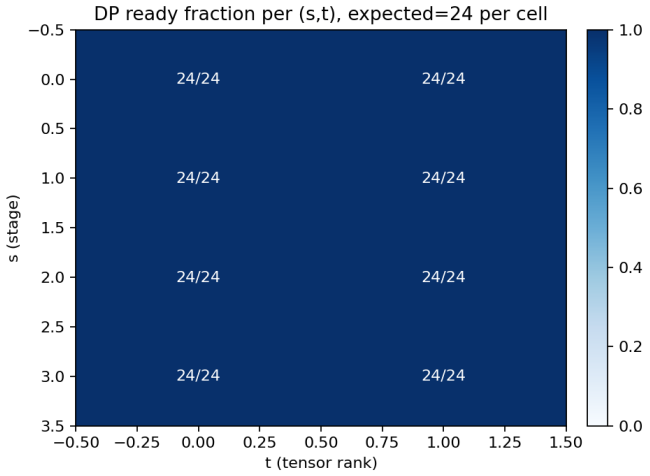

Fig. 5. DP readiness fraction per (stage s, tensor rank t), annotated as ready/expected. Uniform color indicates complete DP readiness.

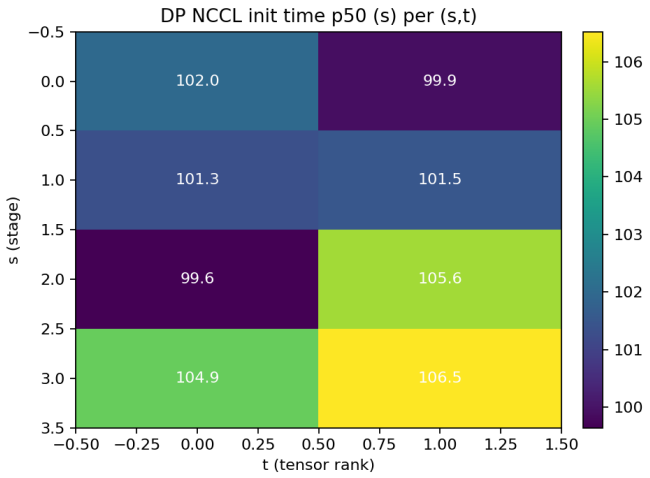

Fig. 6. DP NCCL init latency p50 per (stage s, tensor rank t).

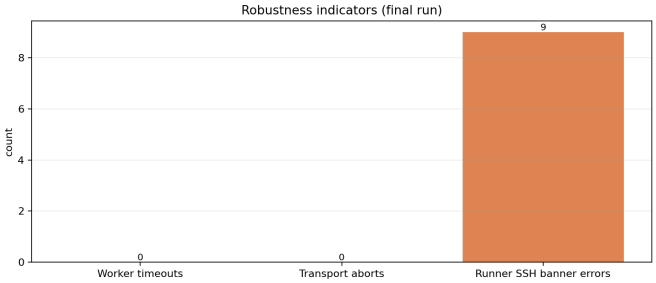

Fig. 7. Robustness indicators (final run): worker timeouts and transport aborts are zero; runner SSH banner errors occurred during artifact collection and were tolerated.

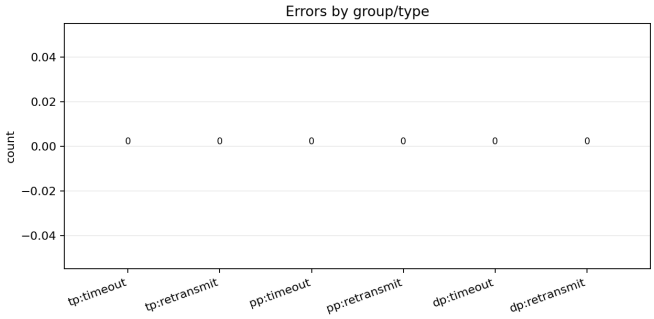

Fig. 8. Errors partitioned by group (TP/PP/DP) and type.