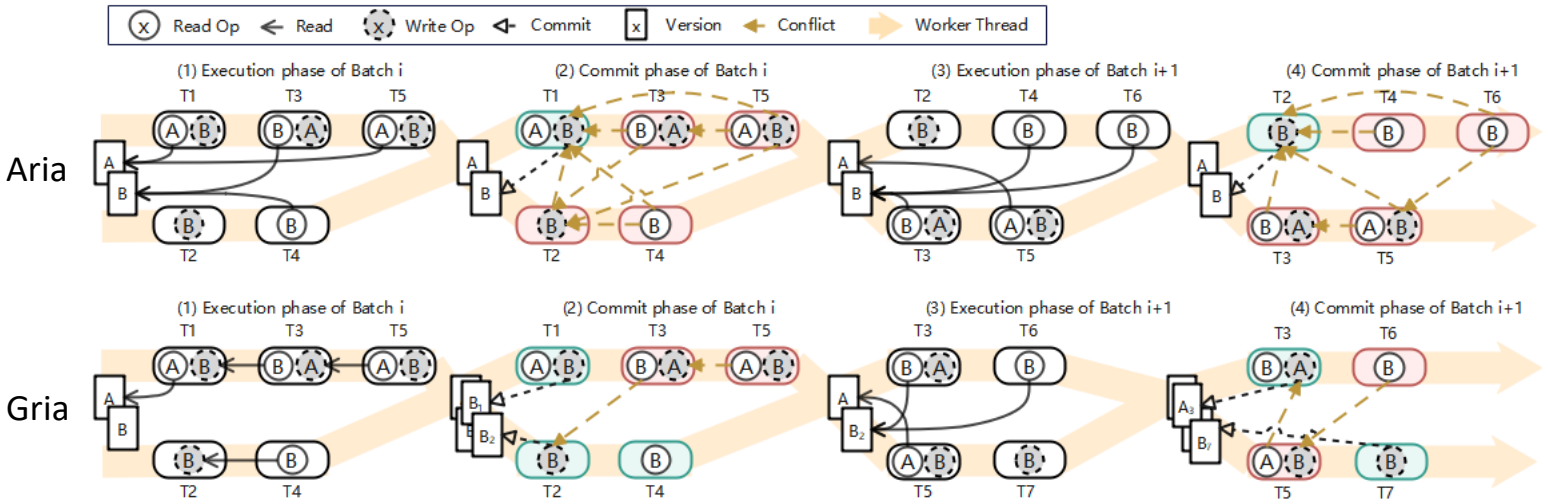
Gria: An Efficient Deterministic Concurrency Control Protocol

Xinyuan WANG, Yun PENG, Hejiao HUANG

Frontiers of Computer Science, DOI: 10.1007/s11704-023-2605-z

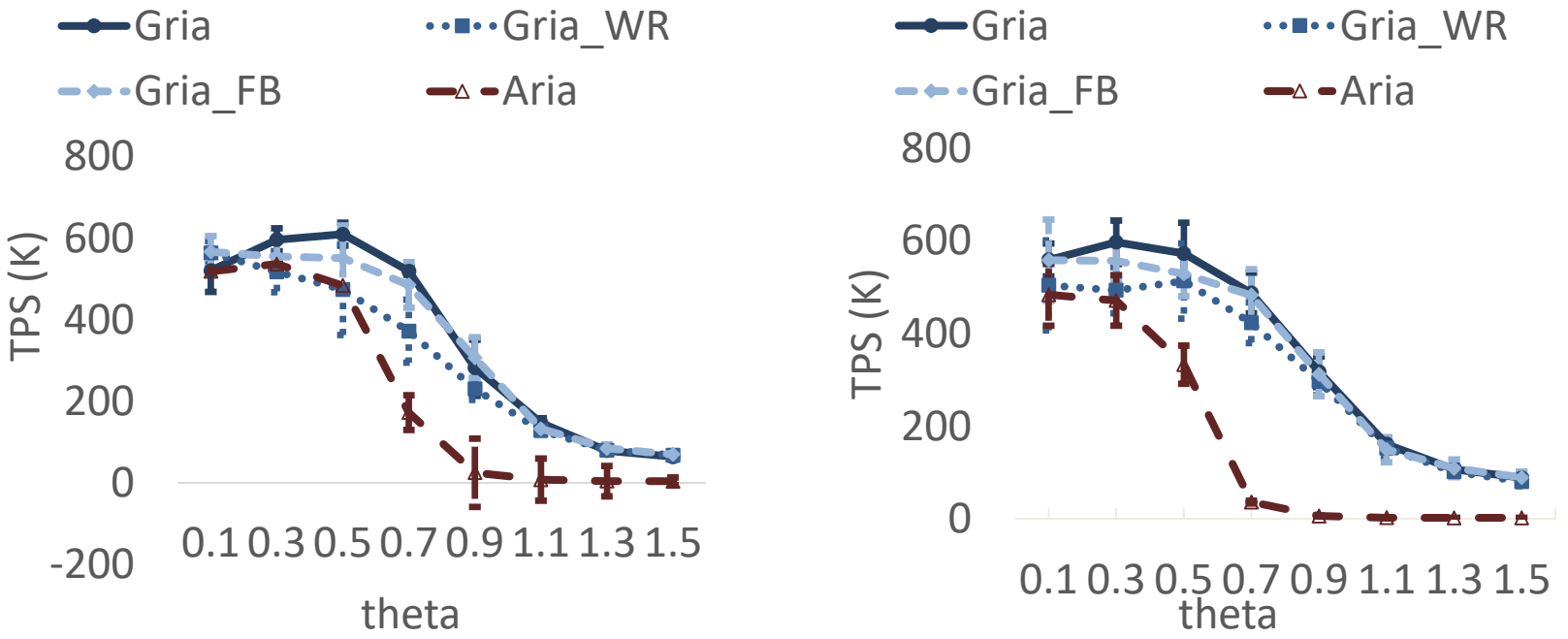
Problems & Ideas



- Problems of the state-of-the-art deterministic concurrency control Aria:
  - it is impractical to configure a suitable batch size for Aria when the read-write set is unknown.
  - Aria running in low-concurrency scenarios, e.g., a single-thread scenario, suffers from the same conflicts as running in high-concurrency scenarios.
  - the single-version schema brings write-after-write conflicts.
- Idea of Gria:
  - the batch size of Gria is auto-scaling based on the number of committed transactions in the previous batch.
  - Each transaction read the versions written by the transaction running by the same thread.
  - Gria builds on the multi-version structure.

Main Contributions

- Contributions:
  - We designed a deterministic concurrency control protocol, Gria. It does not rely on prior knowledge of the read-write set. Besides, Gria is efficient by leveraging the auto-scaling batch size, grouping, and multi-version technique.
  - We present a deterministic reordering mechanism performed in the commit phase to reduce the conflict.
  - We introduce a rechecking strategy performed in the extended phase to avoid false-positive conflicts.



The YCSB results under the read- (left) and write- (right) intensive workloads with varying contentions. (WR: without reordering; RC: with rechecking strategy).