Scalability in Calvin

Discussant: Yeounoh chung
yeounoh_chung at brown
Calvin is amazing!

- Calvin transforms a non-transactional storage system into a shared-nothing, *near-linearly scalable* database system with *full distributed ACID transactions*.

Magic trick: removes synchronization overhead from the distributed system! How? Use deterministic global transaction ordering.
Linear scalability at what cost?

- High scalability is achieved by removing the concurrency control overhead. Is this OK?
  - First, assumption: very high-contention workload [1].
  - Now, all conflicting execution is in serial order [1, 2].
    - Long transaction is a problem!
    - Dependent transaction is a headache!

![Diagram of Calvin and batched txn req](image)

Figure 6: Slowdown for 100% multipartition workloads, varying contention index.

Conclusion

• Calvin is amazing if the workload is *not embarrassingly partitionable* and with *high contention*.

• Otherwise, Calvin can add overheads.
  – Long transactions, dependent transactions.
  – Most real workload in which concurrency control is not the big impediment for the scalability.