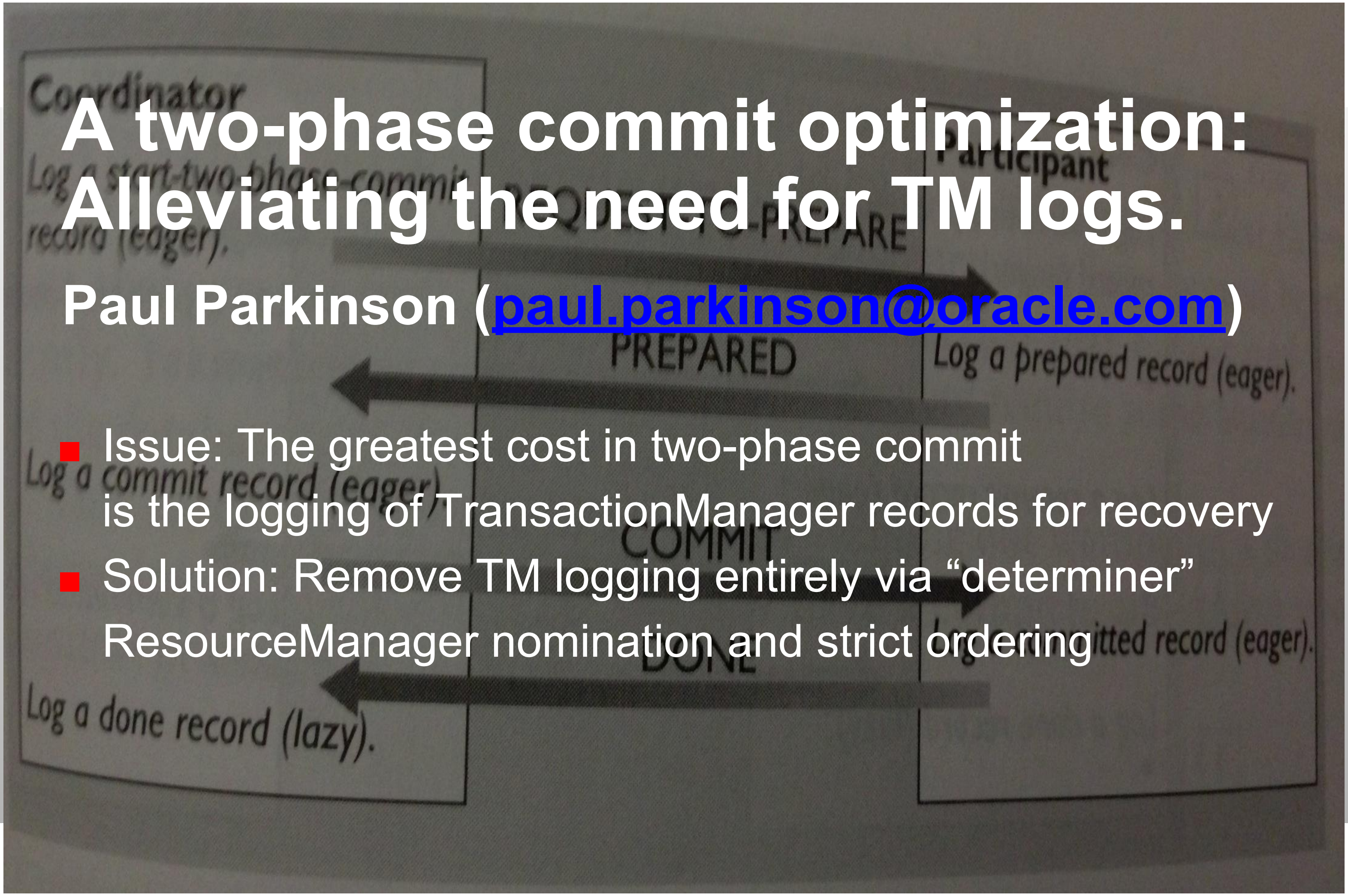
A two-phase commit optimization: Alleviating the need for TM logs.

Paul Parkinson (paul.parkinson@oracle.com)


Coordinator
Log a start-two-phase-commit record (eager).
PREPARE
Participant
PREPARED
Log a prepared record (eager).
Log a commit record (eager).
COMMIT
Log a committed record (eager).
DONE
Log a done record (lazy).

- Issue: The greatest cost in two-phase commit is the logging of TransactionManager records for recovery
- Solution: Remove TM logging entirely via “determiner” ResourceManager nomination and strict ordering

Traditional 2pc runtime and recovery

- TM issues prepare on all RMs (these calls can be done in parallel) and blocks for all prepare acks. **RMs persist gtrid.**
- **TM force writes a transaction log and blocks for ack. TM persists gtrid.**
- TM issues commit on all RMs (this can be done in parallel).
- Recovery: TM **compares in-doubt/prepared gtrids of RMs to gtrids in TM log.**

ORACLE

2pc using determiner nomination and strict ordering runtime and recovery

- TM issues prepare on all RMs except nominated “determiner” RM (this can be done in parallel) and blocks for all prepare acks. RMs persist gtrid. **Determiner RM is prepared last**
- **TM DOES NOT write a tx log**
- TM issues commit on all RMs except nominated “determiner” RM (this can be done in parallel) and blocks for all prepare acks. **Determiner RM is committed last**
- TM (RecoveryManager) **compares in-doubt/prepared tx ids of RMs to in-doubt/prepared tx ids of determiner RM**

Performance and other trade-offs

- ↓ Latency cost due to serial ordering/blocking of determiner resource.
- ↑ Benefits due to removal of TM tx logging:
  - ordered/blocking network and storage (replication, etc.) I/O latency removed
  - resource and/or batch blocking removed
  - memory consumption reduced
  - capacity requirements reduced
  - management and config (for tx store and HA, DR, etc.) requirements reduced
  - Up to 200% (ie 3xs) throughput improvement
  - even greater throughput for distributed txs that span TMs
  - even greater throughput if unknown heuristics are tolerable