Performance Experiments with PSSI

Stephen Revilak, Patrick O'Neil, and Elizabeth O'Neil

HPTS 2011

Oct. 24, 2011

Precisely Serializable Snapshot Isolation (PSSI)

- PSSI is an extension of Snapshot Isolation (SI) that provides Serializability.
- Published in ICDE 2011.

Basic ideas behind PSSI:

- Keep track of read-write, write-read, write-write dependencies between pairs of transactions.
- Abort any T_i that would cause a dependency cycle to form.
- Like SI, PSSI enforces First Updater Wins
- Like SI, reads to not block writes, and writes do not block reads.

Experiments in Improving PSSI's Performance

For this (short) presentation, we're going to look at one data point, and how it changes through several experiments.

These results come from our PSSI prototype, built with MySQL 5.1.31's InnoDB storage engine.

Base Case

Our base case:

- A workload where each transaction reads three rows, and updates one.
- Durable group commit
- Late lock release (release T_i's locks and assign T_i's commit timestamp after logs are flushed to disk).
- Logs written to a desktop-grade 7200 RPM SATA hard disk drive
- 80 MPL (MPL = Multi-programming Level)

Base Case Performance: 756 CTPS

Experiment #1: Use a Faster Disk for Logging

- Replace 7200 RPM HDD with an Intel X25-E SSD.
- SSD reduces group commit duration from $\approx$ 78 ms to 2–3 ms
- Table data resides on a separate HDD. Only use the SSD for logging.

Results:

Disk	CTPS
HDD	756
SSD	3936

- SSD significantly reduces transaction duration
- 420% performance improvement.

Experiment #2: Early Lock Release

- **Early Lock Release**: Release locks and assign T_i's commit timestamp *before* logs are flushed to disk.
- Okay for update transactions, but not read-only transactions.
  - Making early lock release safe for read-only transactions will be a topic for future work.
  - Besides, we have no read-only transactions :)

Disk	Late Lock Release	Early Lock Release
HDD	756	811
SSD	3936	3951

- Reduces the number of active transactions w/in DBMS
- Modest improvement: 7.3% for HDD, $<1\%$ for SSD (Seems like SSD should have done better. Hmmm...)

Experiment #3: Change the Threading Model

- **Old model**: 80 client threads = 80 MPL
- **New model**:
```
active++;
BEGIN;
...
active--;
COMMIT;
```
- Background thread tries to keep `active` ≈ target MPL

Disk	LLR	ELR	ELR + New Thread Model
HDD	756	811	4264
SSD	3936	3951	4477

- Keeps database busy during log flushes
- 425% improvement for HDD. 13% improvement for SSD.
- HDD achieves ≈ 95% of SSD throughput.

Summary

Three Experiments:

- One with hardware
- One with DBMS internals
- One with the DBMS client's threading strategy

Findings:

- SSD's are like fairy dust.
- Early Lock Release (ELR): a big win for S2PL, less so for optimistic methods like PSSI.
- Fast disks always help, but sometimes you can get good mileage out of slow disks.

Thank You