View Reviews

Paper ID
723

Paper Title
Improving OCC Performance Through Transaction Batching and Operation Reordering

Track Name
Research -> March 2018

REVIEWER #1

REVIEW QUESTIONS

1. Overall Rating
Weak Accept

2. Relevant for PVLDB
Yes

3. Are there specific revisions that could raise your overall rating?
Yes

4. Summary of the paper (what is being proposed and in what context) and a brief justification of your overall recommendation. One paragraph
The paper introduces a system that uses batching and transaction reordering to improve the performance of database systems using optimistic concurrency control (OCC). These techniques are used at different levels. At the storage level, batching and reordering of read/write requests are used to apply writes before reads, thus reducing the possibility of stale reads (and thereby, conflicts). At the validator level, these techniques are used to select which transactions to abort in order to optimize for a given objective, e.g., reducing the number of aborted transactions, or reducing average transaction latency. The proposed techniques effectively improve throughput and reduce average and tail latencies.

5. Three (or more) strong points about the paper (Please be precise and explicit; clearly explain the value and nature of the contribution).
S1. The idea of batching and reordering in the validator is novel and interesting.
S2. The algorithm used in the validator can be used to optimize different targets.
S3. The proposed techniques are versatile. They can be extended to any isolation level, and can still improve performance even when implemented on top of an existing system, instead of inside the system.

6. Three (or more) weak points about the paper (Please indicate clearly whether the paper has any mistakes, missing related work, or results that cannot be considered a contribution; write it so that the authors can understand what is seen as negative)
W1. The parallelism technique used in the transaction reordering component of the validator does not really eliminate the bottleneck of the system. The pre-validation is basically reducing the batch size for this component. However, it could have been used in the batch preparation component. In this case, inter-batch conflicts can be further reduced, but the transaction reordering component would have to deal with a larger batch size.

W2. The disadvantage of batching is that transactions have to wait longer before a batch can be accumulated and the actual processing can start. The experiment scenario, however, avoids this by allowing clients to continuously generate new transactions. In a scenario with many concurrent clients where each each only sends a new transaction when the previous one is finished, batching will have a heavier penalty. How would the system perform in this kind of scenario? Also the paper does not mention the concurrency level in their DMBS-X experiments. Does each connection still have a concurrency level of 300?

W3. As shown in the experiment result, the performance of the system is heavily affected by the policy chosen. However, finding a sophisticated policy might not be an easy job. For example, although the mc targets to maximize the throughput of the system, the rct and rdeg policies are capable of achieving the same throughput with smaller tail latency even though they do not specifically target throughput. Another example is the difference between rct and rdeg.

---

7. Novelty (Please give a high novelty ranking to papers on new topics, opening new fields, or proposing truly new ideas; assign medium ratings to delta papers and papers on well-known topics but still with some valuable contribution). If the submission is an "Experiments & Analysis" paper, the novelty should be assessed based on new evaluation metrics and methodology, new data sets enabling experiments and exploration that were infeasible in the past, new findings through experiments and analysis that were unknown in the past, etc.
Novel

8. Significance
Improvement over existing work

9. Technical Depth and Quality of Content
Solid work

10. Experiments
OK, but certain claims are not covered by the experiments

11. Presentation
Excellent: careful, logical, elegant, easy to understand

12. Detailed Evaluation (Contribution, Pros/Cons, Errors); please number each point. If the submission is an "Experiments & Analysis" paper, please specifically comment on whether the submission provides new evaluation metrics and methodology, new data sets enabling experiments and exploration that were infeasible in the past, new findings through experiments and analysis that were unknown in the past, etc.
D1. The paper is well written. The organization is clear, making it easy to follow. One problem is that the acronyms in the figures are quite poor, and some are simply unnecessary (e.g., bch for batch!).

D2. The main idea behind the techniques proposed in this paper is interesting. Some conflicts between two transactions actually do not exist and can be resolved by simply changing the commit or operation order. This paper takes advantage of batching to achieve this. The authors extend this idea further to allow validator reordering to be customized for different targets.

D3. The sort-based algorithm sacrifices accuracy for efficiency. However, it's unclear how general it is and how well it will work with other policies and workloads.

D4. The pre-validation reduces the effective batch size for the transaction reordering component, but doesn't really solve the problem of transaction reordering. It's the same as putting the pre-validation in the batch-preparation component and reducing the batch size. This may actually increase the probability of inter-batch conflicts.

D5. The description for the DBMS-X experiment is a bit unclear. If the connections are continuously sending transactions to the system, how can batching make such a big difference in terms of the concurrency level compared to no batching? Shouldn't the concurrency level be decided by the rate at which each connection sends transactions?

D6. How does transaction reordering methodology relate to recently proposed transaction scheduling algorithms such as VATS (SIGMOD'17) and CATS (VLDB'18)? Are the sizes of read and write sets essentially the same thing as the size of a transaction's dependency set in CATS? Please add a discussion.

13. If revision is required, list specific revisions you seek from the authors
1. Add an experiment to show the performance of the proposed techniques in systems where the clients do not send their next transaction until their previous one is finished.

2. Add a more detailed description regarding the DBMS-X experiment.

3. Improve the legends of the figures.

4. Add an experiment to demonstrate to generality of the sort-based algorithm by using more policies or benchmarks.

REVIEWER #2

REVIEW QUESTIONS

1. Overall Rating
Weak Accept

2. Relevant for PVLDB
Yes

3. Are there specific revisions that could raise your overall rating?
Yes

4. Summary of the paper (what is being proposed and in what context) and a brief justification of your overall recommendation. One paragraph
The paper explores the benefits of storage and validator batching and reordering to improve transaction preformance in optimistic concurrency control systems. Reordering of transactions is non-trivial because of dependencies between transactions. The authors present algorithms that can be tailored for various transaction precedence policies, and show how to parallelize the resulting algorithms. The evaluation on both prototype and production systems show increased throughput and reduced latency.

5. Three (or more) strong points about the paper (Please be precise and explicit; clearly explain the value and nature of the contribution).
[S1] The problem is clearly described from an intuitive perspective.
[S2] The paper is well structured, making it easy for the reader to follow.
[S3] The experimentation is appropriate.

6. Three (or more) weak points about the paper (Please indicate clearly whether the paper has any mistakes, missing related work, or results that cannot be considered a contribution; write it so that the authors can understand what is seen as negative)
[W1] There is analysis of the runtime of the algorithms. What are the other theoretical guarantees of the approach?
[W2] Characterisation of the algorithms would be interesting. There are many moving parts in the experiments so it is difficult to picture the tradeoffs holistically. Can the details of the results be characterised using formulas?
[W3] The constraints/assumptions on the work are not formally described.

7. Novelty (Please give a high novelty ranking to papers on new topics, opening new fields, or proposing truly new ideas; assign medium ratings to delta papers and papers on well-known topics but still with some valuable contribution). If the submission is an "Experiments & Analysis" paper, the novelty should be assessed based on new evaluation metrics and methodology, new data sets enabling experiments and

exploration that were infeasible in the past, new findings through experiments and analysis that were unknown in the past, etc.
Novel

8. Significance
Improvement over existing work

9. Technical Depth and Quality of Content
Solid work

10. Experiments
Very nicely support the claims made in the paper

11. Presentation
Reasonable: improvements needed

12. Detailed Evaluation (Contribution, Pros/Cons, Errors); please number each point. If the submission is an "Experiments & Analysis" paper, please specifically comment on whether the submission provides new evaluation metrics and methodology, new data sets enabling experiments and exploration that were infeasible in the past, new findings through experiments and analysis that were unknown in the past, etc.
It was initially difficult to understand what the paper was doing and particularly the constraints on the problem being solved.

The constraints/assumptions on the work are not formally described.
In the introduction the authors say “Our system design is best suited for integration with an in-memory versioned key-value store”. What needs to be changed for other kinds of stores?

You should always write acronyms in full the first time they are used. Please check them all. I suggest that you expand “OCC” in the title or remove it.

There are a number of typos and grammatical problems that need to be addressed. I list a subset of them below:
“processes” in the introduction, and in Section 3.2 should be “process”
“reducing same object conflicts” -> “reducing some object conflicts”
in section 4.2, “we trim the remaining the graph again”
at the end of section 5, “through” -> “though”
in section 6 “How does these algorithms”
in section 6, “transations”

13. If revision is required, list specific revisions you seek from the authors
Add a discussion which explains what constraints on the system make the system best suited for integration with an in-memory versioned key-value store and what would need to change to be integrated with other stores.

There are many moving parts in the experiments so it is difficult to picture the tradeoffs holistically. I suggest that either the results are represented in a formula or represented in a table showing the tradeoffs.

REVIEWER #3

REVIEW QUESTIONS

1. Overall Rating
Weak Reject

2. Relevant for PVLDB
Yes

3. Are there specific revisions that could raise your overall rating?
Yes

4. Summary of the paper (what is being proposed and in what context) and a brief justification of your overall recommendation. One paragraph
This paper focuses on how to exploit semantic batching at various stages of OCC protocol/transactions' lifecycle including read, writes, and validation phases.

5. Three (or more) strong points about the paper (Please be precise and explicit; clearly explain the value and nature of the contribution).
S1) The problem of batching transaction at different stages by either re-ordering operations or re-ordering commit (backward validation) is promising

S2) Unlike past approaches, the proposed greedy algorithms focused on reducing intra-batch aborts does not assume knowing the read/write sets

S3) Detailed experimental evaluation that studies the impacts proposed optimizations

S4) The paper is very well written

6. Three (or more) weak points about the paper (Please indicate clearly whether the paper has any mistakes, missing related work, or results that cannot be considered a contribution; write it so that the authors can understand what is seen as negative)
W1) The papers contains a bag of ideas/optimizations, arguably unrelated, based on known techniques, so the overall contributions and novelty is limited

W2) Although the evaluation is comprehensive and detailed, the authors only presented a micro benchmark, a self-comparison without considering other state-of-the-art approaches.

w3) The authors provide a black box comparison of their approach in DB-X, but the basis of the comparison is unclear (also not sure what does "Good Throughput/Transaction" means).

7. Novelty (Please give a high novelty ranking to papers on new topics, opening new fields, or proposing truly new ideas; assign medium ratings to delta papers and papers on well-known topics but still with some valuable contribution). If the submission is an "Experiments & Analysis" paper, the novelty should be assessed based on new evaluation metrics and methodology, new data sets enabling experiments and exploration that were infeasible in the past, new findings through experiments and analysis that were unknown in the past, etc.
With some new ideas

8. Significance
Improvement over existing work

9. Technical Depth and Quality of Content
Solid work

10. Experiments
Obscure, not really sure what is going on and what the experiments show

11. Presentation

Reasonable: improvements needed

12. Detailed Evaluation (Contribution, Pros/Cons, Errors); please number each point. If the submission is an "Experiments & Analysis" paper, please specifically comment on whether the submission provides new evaluation metrics and methodology, new data sets enabling experiments and exploration that were infeasible in the past, new findings through experiments and analysis that were unknown in the past, etc.

D1) The papers contains a bag of ideas/optimizations, arguably unrelated, based on known techniques, so the overall contributions and novelty is limited

D2) Although the evaluation is comprehensive and detailed, but the authors only presented a micro benchmark, a self-comparison without considering other state-of-the-art approaches. Here are few important related CCs (related work discussion can also take into these approaches as well)

Per-Åke Larson, Spyros Blanas, Cristian Diaconu,Craig Freedman, Jignesh M.Patel, and MikeZwilling. High-performance concurrency control mechanisms for main-memory databases. PVLDB'11

Stephen Tu, Wenting Zheng, Eddie Kohler, Barbara Liskov, and Samuel Madden. Speedy transactions in multicore in-memory databases. SOSP '13

Mohammad Sadoghi, Mustafa Canim, Bishwaranjan Bhattacharjee, Fabian Nagel, and Kenneth A. Ross. Reducing database locking contention through multi-version concurrency. PVLDB'14

H. Kimura. FOEDUS: OLTP engine for a thousand cores and NVRAM. SIGMOD'15.

C. Yao, D. Agrawal, G. Chen, Q. Lin, B. C. Ooi, W. F. Wong, and M. Zhang. Exploiting single-threaded model in multi-core in-memory systems. TKDE'16

Kangnyeon Kim, Tianzheng Wang, Ryan Johnson, and Ippokratis Pandis. ERMIA: fast memory-optimized database system for heterogeneous workloads. SIGMOD'16

Tianzheng Wang, Hideaki Kimura: Mostly-Optimistic Concurrency Control for Highly Contended Dynamic Workloads on a Thousand Cores. PVLDB'17

Jose M. Faleiro, Daniel Abadi, Joseph M. Hellerstein: High Performance Transactions via Early Write Visibility. PVLDB'17

D3) The authors provide a black box comparison of their approach in DB-X, but the basis of the comparison is unclear (also not sure what does "Good Throughput/Transaction" mean). Although the effort is appreciated, it would be much better to compare with relevant, disclosed existing algorithm, or at the very least, the concurrency of the model of DB-X is carefully explained. A black box graph adds no value.

D4) It is unclear why the authors choose to have 300 active transactions, which would imply the need for having 300 physical threads. I further suppose, the authors considering in-memory implementation given all OLTP DB can fit entirely in memory today. If in fact, the number of active transactions is larger than threads, then there will be many context switches, and frankly, the whole setting would be questionable, which could also

explain why the overall throughput is never exceeded half-million transactions/second.

D5) In some graphs, x-axis label is cut off.

13. If revision is required, list specific revisions you seek from the authors

Addressing D1-D5