LI Yuchen

From: Transactions on Knowledge and Data Engineering <onbehalfof@manuscriptcentral.com>
Sent: Thursday, March 12, 2020 6:13 PM
To: LI Yuchen; zhangjing000@zju.edu.cn; liuyue1013@zju.edu.cn; lvzheng.lz@alibaba-inc.com; hzd@zju.edu.cn; sunjl@zju.edu.cn
Subject: Decision Re: TKDE-2019-06-0571.R1

RE: TKDE-2019-06-0571.R1, "Dynamic Hash Tables on GPUs"
Manuscript Type: Regular

12-Mar-2020

Dear Dr. Li,

We have completed the review process of the above referenced paper that was submitted to the IEEE Transactions on Knowledge and Data Engineering. Enclosed are your reviews. We hope that you will find the editor's and reviewers' comments and suggestions helpful.

I regret to inform you that based on the reviewer feedback, we could not recommend publishing your paper. Final decisions on acceptance are based on the referees' reviews and such factors as restriction of space, topic, and the overall balance of articles.

PLEASE NOTE:

The current editorial TKDE policy states that a rejected paper not be resubmitted to TKDE within the next 12 months. We also anticipate that the revised version, if resubmitted, should contain significantly new technical content compared to the original version.

Thank you for your interest in the IEEE Transactions on Knowledge and Data Engineering.

Sincerely,

Xuemin Lin
Editor-in-Chief
IEEE Transactions on Knowledge and Data Engineering xueminlin.TKDE@gmail.com

****
Editor Comments

Associate Editor
Comments to the Author:
While the reviewers appreciate the authors' efforts to address their comments, two of the reviewers still have many important concerns that remain unaddressed in the revision. I hope the authors find these comments helpful in improving the manuscript and submitting elsewhere.

****
Reviewer: 1

Recommendation: Accept With No Changes

Comments:
The authors have answered my questions. I recommend accepting this paper for publication.

Additional Questions:
1. Which category describes this manuscript?: Practice / Application / Case Study / Experience Report

2. How relevant is this manuscript to the readers of this periodical? Please explain your rating under Public Comments below.: Relevant

1. Please explain how this manuscript advances this field of research and/or contributes something new to the literature.: This paper introduces a dynamic cuckoo hash table on the GPU and proposes a two-layer cuckoo hashing scheme to ensure efficient hash table operations. Compared to relative methods, their methods achieve excellent performance and memory efficiency.

2. Is the manuscript technically sound? Please explain your answer under Public Comments below.: Yes

1. Are the title, abstract, and keywords appropriate? Please explain under Public Comments below.: Yes

2. Does the manuscript contain sufficient and appropriate references? Please explain under Public Comments below.: References are sufficient and appropriate

3. Does the introduction state the objectives of the manuscript in terms that encourage the reader to read on? Please explain your answer under Public Comments below.: Yes

4. How would you rate the organization of the manuscript? Is it focused? Is the length appropriate for the topic? Please explain under Public Comments below.: Satisfactory

5. Please rate the readability of the manuscript. Explain your rating under Public Comments below.: Readable - but requires some effort to understand

6. Should the supplemental material be included? (Click on the Supplementary Files icon to view files): Yes, as part of the digital library for this submission if accepted

7. If yes to 6, should it be accepted: As is

Please rate the manuscript. Please explain your answer.: Excellent

Reviewer: 2

Recommendation: Author Should Prepare A Major Revision For A Second Review

Comments:
C1: While the authors strengthened the motivation, I missed an incorrect statement during my first review: “One must resort to expensive PCIe data transfers between CPUs and GPUs as the hash table takes an unnecessarily large memory space”. It is common practice to have multiple data-structures like hash tables co-exist on CPU and GPU and do the compute where the data resides. This often eliminates data transfers which are less costly using NVLINK connected GPUs. I suggest the authors just drop this argument and focus on efficient memory management as they do in the following paragraph.

C2: The authors should just include the general explanation applicable to CPU and GPU as suggested by the comment.

C3: While the authors already re-ran the experiments, they still chose a GPU from 2 generations ago. Current GPUs are readily available in the cloud, e.g., V100. Moreover, the authors run their experiment on CUDA 9.1 which was released in Dec. 2017, more than 2 years ago. CUDA 10 was released in Sept. 2018. They should upgrade to the latest CUDA 10.2 release in Nov 2019 at this point. While I understand that procuring the newest hardware is not always possible, authors should make an effort to at least leverage the latest advances in software, especially when available for free. Older software and drivers may not demonstrate the best performance.

C4: I was referring to CUDPP performance.

The statement that the K40 and GTX1080 have similar specs is incorrect. The K40 achieves 288GB/s memory bandwidth while a GTX1080 achieves up to 352GB/s, i.e. 22% more. As discussed in C2 random memory bandwidth is the gating factor for hash table performance. Moreover, newer GPU architectures with improved TLB coverage and management achieve significantly higher random memory bandwidth over older ones, c.f., appendix of [9].

Achieving significantly lower performance for the same experiment on newer -or even similar hardware as the authors claim- raises questions about the authors experimentation/setup. In [3] Fig 4.a CUDPP achieves 500M inserts/sec while in Fig. 10 in the original version of this paper the authors reported less than 400M for CUDPP, both for a fill factor of 65%. Similarly [3] Fig 4.b reports 1B searches/sec for CUDPP while in Fig. 10 in the original version of this paper the authors reported less than 600M for CUDPP, again both for a fill factor of 65%. The authors need to investigate these discrepancies by revisiting their experimental setup (also see C3) and if the discrepancies persist, they need to be explained.

C5: Comparing the authors results in Fig. 10 vs [8] Fig. 7 it appears they chose to only report Warp Drive results for group size 1. Choosing the optimal group size of 4 yields more than 30% performance improvement for insertion and up to 70% for search. While this makes DyCuckoo performance appear inferior to Warp Drive, one has to keep in mind the superior performance of DyCuckoo when it comes to resizing. I appeal to the authors academic honesty to change their comparison with WarpDrive to use the appropriate settings, i.e., group size 4 and clearly explain their observations and differences.

Additional Questions:
1. Which category describes this manuscript?: Research/Technology

2. How relevant is this manuscript to the readers of this periodical? Please explain your rating under Public Comments below.: Relevant

1. Please explain how this manuscript advances this field of research and/or contributes something new to the literature.: The paper describes a hash table implementation for the GPU, that allows to dynamically re-size hash tables with controlled performance degradation.

2. Is the manuscript technically sound? Please explain your answer under Public Comments below.: Partially

1. Are the title, abstract, and keywords appropriate? Please explain under Public Comments below.: Yes

2. Does the manuscript contain sufficient and appropriate references? Please explain under Public Comments below.: References are sufficient and appropriate

3. Does the introduction state the objectives of the manuscript in terms that encourage the reader to read on? Please explain your answer under Public Comments below.: Yes

4. How would you rate the organization of the manuscript? Is it focused? Is the length appropriate for the topic? Please explain under Public Comments below.: Satisfactory

5. Please rate the readability of the manuscript. Explain your rating under Public Comments below.: Readable - but requires some effort to understand

6. Should the supplemental material be included? (Click on the Supplementary Files icon to view files): Does not apply, no supplementary files included

7. If yes to 6, should it be accepted:

Please rate the manuscript. Please explain your answer.: Fair

Reviewer: 3

Recommendation: Author Should Prepare A Minor Revision

Comments:
-I am on the fence as to whether this paper should be accepted. The evaluation is extensive, the work is helpful to the community to understand how to implement a high-throughput GPU hash table, and I am not familiar with an abundance of work on dynamic GPU hash tables in the research literature. That being said, I think there are some concerns that remain.

-The one thing that needs to be fixed is the following: No plot showing how much memory each of the different designs use. You say it's about the same. If so, make a plot or table showing it's about the same. Are they within 2x of each other? This matters given how comparatively scarce memory is on a GPU and where total cost of ownership when deploying a GPU-hash table is going to be largely driven by how efficiently each design uses memory. A discussion of memory use and performance trade-offs between the designs is necessary for determining whether DyCuckoo is practical. Further, if a design needs to page more of its memory to the remote host's DRAM (supported on newer GPU designs), that would have big impacts on performance. I am willing to accept that you do not consider off-GPU memory accesses as part of the scope of your study, but at least show us that you make conservative use of the on-GPU memory. Otherwise, a more nuanced discussion is necessary.

- The update does not address my prior comment on throughput as table size is varied. A simple plot or two could suffice.

-The extensive benchmarking on the RAND dataset is still an issue for me given real workloads are often highly-skewed in their key distribution. I think that these figures could appear in the paper. In the figures included in your rebuttal but which are not in the paper, DyCuckoo appears to perform less favorably to the other works. Inclusion of these less favorable results along with a brief discussion of why would be helpful to readers.

-A related concern is that DyCuckoo does not appear to be considerably faster than WarpDrive. However, DyCuckoo is dynamic whereas WarpDrive is not. Is this enough? Compared to the other dynamic approaches surveyed it is significantly faster.

- I still also hold my concern regarding there being a lack of results presented for throughput versus table size over the different designs.

- I'm not convinced that the performance comparison with libcuckoo is an entirely fair one given that libcuckoo supports finer-grain concurrency (i.e., a workload consisting of a mix of puts and gets). This paper needs to batch puts and gets, so when benchmarked against libcuckoo, there is a chance that libcuckoo is unfairly penalized. I don't think there is an easy way to fix this other than mentioning this in the discussion. A reimplementation of the proposed design on CPUs seems unnecessary given that it is well-established that hash tables perform well on the GPU. I realize the libcuckoo results are included to placate another reviewer, but in my opinion comparison with libcuckoo is a bit of a straw man.

- I think the presentation of r and the associated results could be made clearer. Perhaps a back reference to Table 3 in the captions of plots in which it is used. Some of the labels on Figures are too small. I noticed this in particular for plots where the x-axis is labeled with r.

- That being said, I believe the concerns of the three reviewers were not in alignment, and I think that the authors have made a valiant go of our addressing our diverging feedback.

Additional Questions:
1. Which category describes this manuscript?: Research/Technology

2. How relevant is this manuscript to the readers of this periodical? Please explain your rating under Public Comments below.: Relevant

1. Please explain how this manuscript advances this field of research and/or contributes something new to the literature.: -Presents a thorough evaluation of GPU hash tables of varying types -Applies or slightly reworks known techniques for resizing and high-throughput puts and gets on CPU hash tables to GPU hash tables

2. Is the manuscript technically sound? Please explain your answer under Public Comments below.: Yes

1. Are the title, abstract, and keywords appropriate? Please explain under Public Comments below.: Yes

2. Does the manuscript contain sufficient and appropriate references? Please explain under Public Comments below.: References are sufficient and appropriate

3. Does the introduction state the objectives of the manuscript in terms that encourage the reader to read on? Please explain your answer under Public Comments below.: Yes

4. How would you rate the organization of the manuscript? Is it focused? Is the length appropriate for the topic? Please explain under Public Comments below.: Satisfactory

5. Please rate the readability of the manuscript. Explain your rating under Public Comments below.: Readable - but requires some effort to understand

6. Should the supplemental material be included? (Click on the Supplementary Files icon to view files): Does not apply, no supplementary files included

7. If yes to 6, should it be accepted:

Please rate the manuscript. Please explain your answer.: Good