Front.Comput.Sci.
DOI

LETTER

Online Resource 2 of "A Lock-free Approach to Parallelizing Personalized PageRank Computations on GPU"

Zhigang Wang[1], Ning Wang (✉)[1], Jie Nie[1], Zhiqiang Wei[1], Yu Gu[2], Ge Yu[2]

1 Faculty of Information Science and Engineering, Ocean University of China, Qingdao 266100, China
2 School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China

This supplementary file gives the detailed description about experimental settings in Section 5.

Compared solutions: Our baseline competitor is the up-to-date GPU-based Forward Push implementation proposed in Ref. [4], denoted by *push*. It removes atomic detections by atomically reading the value of a destination vertex v, $r(v, s)$, right before updating $r(v, s)$. It then locally compares un-updated and updated $\frac{r(v,s)}{d_v^+}$ with the threshold θ, respectively. Clearly, $\frac{r(v,s)}{d_v^+}$ firstly reaches θ if and only if the former comparison result is negative but the latter is positive. We call this as a valid local comparison. Differently, this paper optimizes atomic detections by separating message propagation and frontier generation, termed as *pushopt*. On the other hand, Pull with binary combination given in Sec. 3 is called as *pull*. Finally, *hybrid* is used to stand for the hybrid framework. Note that for improving resource utilization, all GPU-based implementations directly integrate the flexible workload mapping mechanism and the topology-based load-balancing strategy, the principal advantages of the well-known graph processing library Gunrock [3].

Hardware environments: Experiments are mainly performed on two GPU devices: GeForce GT330 with low configurations (96-Cores, 1.34GHz, and 1GB memory), denoted by *GPUL*; and Tesla V100 with high configurations (5120-Cores, 1.38GHz, and 16GB memory), denoted by *GPUH*. Then we can analyze the performance features of all compared solutions when varying hardware environments. To show the parallel advantages of GPU, we also test *push* and *pushopt* on a multi-cores CPU device equipped with Intel(R) Xeon(R) Silver 4108 (32-cores, 3000MHz) and 32GB memory. By contrast, *pull* and *hybrid* are not tested on CPU since they focus on eliminating atomic writes—the key performance bottleneck of GPU. Solutions using CPU and GPU are distinguished with prefix-markers *cpu* and *gpu*, respectively.

Received month dd, yyyy; accepted month dd, yyyy

E-mail: wangning8687@ouc.edu.cn

Graph datasets: Table 1 summarizes features of graphs used in experiments. Among them, *Pokec*, *Wiki*, *Stack*, and *LiveJ* are four graphs crawled from real applications ranging from social network to Internet web service.

Table 1: Graph datasets used in experiments

Graph	Number of vertices ($\|V\|$)	Number of edges ($\|E\|$)
Pokec[1)]	1,632,804	30,822,675
Wiki[2)]	1,791,489	28,508,141
Stack[3)]	2,601,977	47,903,266
LiveJ[4)]	4,847,571	68,475,302

[1)] http://snap.stanford.edu/data/soc-Pokec.html
[2)] http://snap.stanford.edu/data/wiki-topcats.html
[3)] http://snap.stanford.edu/data/sx-stackoverflow.html
[4)] http://snap.stanford.edu/data/soc-LiveJournal1.html

Important parameters: When implementing PPR, we conventionally set the decay factor α as 0.8. For GPU devices, since the *warp* size w is inherently fixed as 32 by the manufacturer, we tune the parameter b to control the number of threads within a *block*. To find an optimal value, we repeatedly run PPR and manually set b every time. We find that the best performance is achieved when $b = 32$. In addition, we also test the costs of random write-lock, random read, and sequential write, by respectively running the following benchmark operations: 1) atomically saving a pre-fixed residue value into destination vertices along outgoing edges; 2) reading residues of source vertices along incoming edges; and 3) updating PPR scores of every vertex. By recording the runtime and the number of specific operations, we know the unit cost on average. Specifically, we have λ_1:λ_2:λ_3=6.3:3.2:1 for *GPUL*, and 2.3:1.1:1 for *GPUH*. The latter clearly has prominent reading and writing-locking performance.