Concurrent hash tables on multicore machines: Comparison, evaluation and implications

Zhiwen Chen\textsuperscript{a}, Xin He\textsuperscript{a}, Jianhua Sun\textsuperscript{a}, Hao Chen\textsuperscript{a,\textsuperscript{*}}, Ligang He\textsuperscript{b}

\textsuperscript{a} College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
\textsuperscript{b} Department of Computer Science, University of Warwick, Coventry CV4 7AL, United Kingdom

\textbf{HIGHLIGHTS}

- A unified testing framework for concurrent hash tables is presented.
- Extensive evaluations are performed from a wide range of perspectives.
- The experiments are conducted on four major multi-core hardware platforms.
- Implications made in this paper could be used as guidelines for future research.

\textbf{ARTICLE INFO}

Article history:
Received 3 August 2017
Received in revised form 13 November 2017
Accepted 27 December 2017
Available online 1 January 2018

Keywords:
Concurrent hash table
Multiprocessors
Multi-core
Synchronization
NUMA
Hardware transactional memory

\textbf{ABSTRACT}

Concurrent hash table has been an area of active research in recent years, and a wide variety of fast and efficient concurrent hash tables (CHTs) have been proposed to exploit the advantages of modern parallel computer architectures such as today's mainstream multi-core systems. As one of the fundamental data structures widely used in software systems, existing works on CHTs focus on either algorithmic improvements, or hardware-oriented optimizations, or application-specific designs. However, there is a lack of a comprehensive and comparative study on different implementations. In this paper, we conduct an experimental study on the state-of-the-art, and our goal is to critically review existing CHTs from wider aspects and with more detailed analysis. Concretely, we have conducted extensive evaluations of five CHTs using a unified testing framework on four multi-core hardware platforms, and implemented our HTM-based concurrent hash table. A variety of metrics such as throughput scalability, latency, impact of memory hierarchy, thread pinning strategies, synchronization mechanisms, and memory consumption, are measured in order to obtain the deep insights about performance impediments and good design choices. With this study, we hope to identify potential issues and pinpoint promising directions for future research of CHTs.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

As the number of cores has been increasing on modern computer architectures in recent years, challenges in inventing new or enhancing existing concurrency data structures to fully leverage the hardware advancements are emerging. Hash table is a well-known data structure, which provides simple interfaces to access the elements. \texttt{lookup, insert} and \texttt{delete} are the three main operations provided by hash tables. As it can offer constant time lookup and update operation, it is widely used in most software systems [1,2]. In spite of the fact that the study on sequential hash tables is relatively mature, the research on concurrent hash tables (CHTs) has attracted a lot of efforts in recent years, due to the promising performance, hardware advancement, and diversified requirements in different usage scenarios.

Ideally, CHTs should achieve high performance and scalability under different workloads and hardware settings. However, designing and implementing such CHTs is very challenging [3,4]. Hardware-conscious CHTs leveraging platform-specific features are often ineffective in obtaining portable performance [5]. On the other hand, hardware-oblivious CHTs often fail to achieve highest possible performance. Similarly, a CHT optimized for a specific type of workloads may exhibit poor performance under a slightly different workload. For example, the Read-Copy-Update (RCU) based hash table is one of the workload sensitive CHTs. It obtains high throughput and shows good scalability when dealing with read-only workloads. However, for workloads with a small fraction of update operation, it exhibits significant performance degradation. In [5], the authors present a complete picture of
how synchronization schemes behave in concurrent algorithms. In most cases, when a scalability issue is encountered, it is not straightforward to identify the root cause that may be the underlying hardware, synchronization algorithm, usage of specific atomic primitives, application context, or workloads.

Although both the industry and academia have proposed a range of CHTs with different target such as Threading Building Blocks (TBB) [6] and ConcurrentHashMap in Java [7], the performance of CHTs not only is related to the application requirements but also relies on the exploitation of lower-level hardware characteristics. When profiling CHTs, we need to perform the analysis by integrating many relevant elements and considering at different levels rather than evaluating based only on intuitive metrics such as throughput and latency. Furthermore, with a unified testing framework and a set of common performance metrics, it seems that no single CHT can outperform others in all aspects when handling a diversified set of workloads. On the other hand, from the perspective of users, the effective way to adopt a CHT is to clearly recognize all the potential performance obstacles. Unfortunately, these practical concerns are rarely mentioned in previous studies. Lacking a unified benchmark to evaluate CHTs, it is hard for the users to make a decision about which CHTs to employ in their software systems in order to attain desired performance. In summary, a comprehensive and in-depth analysis is of significance in using, designing, and optimizing CHTs. Inspired by the practices, we choose five state-of-the-art CHTs to conduct a comprehensive evaluation and analysis across a wide set of metrics. The five CHTs taken from the literature are listed in Table 1 with brief description.

CHTs in this study are written in C/C++, and they are evaluated on 4 multi-core platforms: AMD Opteron, Intel Xeon Phi 7120P (a many-core platform based on Intel MIC architecture), Intel Xeon E5-2630, and Intel Xeon E7-4850. To the best of our knowledge, it is the most comprehensive evaluation of concurrent hash tables to date. We make the following contributions in this paper.

- First, we present a framework, named CHT-bench, which provides a fair testing environment and unified interface for the experiments by hiding the discrepancies of hardware platforms, synthesized workloads, concurrency models, and compiler configurations. The source code of this work can be found at https://github.com/Gwinel/CHT-bench. In this way, we can guarantee the experimental results generated from our framework are fairly comparable between different CHTs.

- Second, the evaluations are explored from a wide range of perspectives including thread scalability, throughput, latency, memory hierarchy impact, low-level synchronization primitives, and memory usage. The inter-correlations between relevant metrics are also discussed when necessary. The experiments are conducted on four major hardware platforms including Intel MIC and three representative NUMA systems. We ported CHTs to the MIC platform, and to our knowledge, this is the first extensive study of concurrent hash tables on Intel MIC architecture.

- Third, implications about pitfalls, design trade-offs, and desirable optimizations are summarized for each evaluated metric, which can serve as guidelines for future research and practical development of CHTs.

The rest of the paper is organized as follows. Section 2 provides brief background on CHTs and modern computer hardware features. We present the experimental platforms and parameter configurations in Section 3. A comprehensive analysis of CHTs are made in Section 4. The related work is presented in Section 5. Section 6 concludes this paper.
دریافت فوری متن کامل مقاله

امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات