DLOCK: Unlocking systems with the NEUR approach

Blind submission

Abstract

Several systems maintain a shared state between several writers and readers where writers' execution flow is independent of the shared state, and the shared state update concern simple arithmetric operations. For example, incrementing the number of read block requests in the Linux kernel.

In this paper, we present NEUTRAL RECOMBINE (NEUR) a novel mechanism to ensure consistency for the shared state without requiring locks or transactions. With NEUR, instead of waiting for a lock holder, writers perform their arithmetic operation on the neutral element of the arithmetic operation and buffer the result. Then, the different writers results are recombined to get the shared state which readers can retrieve. Compared to locking mechanims, NEUR does not require waiting for a lock holder completion and is priority inversion free. To ease NEUR adoption and show its advantages, we built DLOCK. DLOCK is a tool that parses the source code of a project and by feeding it with locks primitives definition, detects code sections using locks in a context that matches NEUR scope usage, and transforms the code to apply the NEUR approach. DLOCK also introduces a garbage collector to free writers local copy.

We apply DLOCK on crafted applications and real systems such as the Linux kernel, Zookeeper, HiBench, and Memcached. Our evaluations show that the DLOCK generated versions achieve up to $xxx\times$ better throughput and execute up to $xxx\times$ faster for different configurations due to the NEUR approach.

ACM Reference Format:
Blind submission . 2023. DLOCK: Unlocking systems with the NEUR approach. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Cloud computing and virtualization. Users' computing needs become more diversified leading to a variety of workloads[] forcing data centers holders to propose new classes of services to their clients[]. To efficiently use available hardware resources and power their different services, Cloud providers use virtualization[]. Virtualization provides a secure way to multiplex hardware resources between different users[]. With virtualization, users' workloads run in a blackbox entity refered as virtual machine (VM), which can be rapidly initialized and migrated from one server to another to ensure scalability and fault-tolerance.

Energy usage and frequency states. Unfortunately, with Cloud computing becoming more attractive, the energy used by data centers around the world has skyrocketed in recent years[] and reached up to $xx\%$ of the total use of energy on earth in $xxxx$. This led to data centers being pointed out due to the potential effects on global warming[]. To provide finer control on the energy used by a server, hardware vendors such as Intel and AMD introduced frequency states[]. Todays' processing core can have different frequency states characterized by the performance of the core in each state. For example, some Intel cores can enter {P,E,C}-states where P-states stand for performance — favoring performance at the expense of

2 Background and related work

2.1 Background

2.2 Related work

Parler des transactions, lock-free, consensus numbers, java syncrhonization thread model, Rust forwarding lock

3 Problem: Incorrect accounting scenarios

4 Our solution: uiGOV

5 Evaluations

6 Discussion and future work

7 Related work

NWAP[2] observes polling/interrupts to rapidly change P-core to meet SLO. Also has refs on OS-driven DVFS.

[3] study of better DVFS models for applications.

Peafowl[1] energy aware scheduling for key value stores.

Yawn[4] idle state governor that uses online machine learning to improve idle state predictions on cores.

8 Conclusion

References

[1] Esmail Asyabi, Azer Bestavros, Erfan Sharafzadeh, and Timothy Zhu. Peafowl: In-application cpu scheduling to reduce power consumption of in-memory key-value stores. In *Proceedings of the 11th ACM Symposium on Cloud Computing*, SoCC '20, page 150–164, New York, NY, USA, 2020. Association for Computing Machinery.

[2] Ki-Dong Kang, Gyeongseo Park, Hyosang Kim, Mohammad Alian, Nam Sung Kim, and Daehoon Kim. Nmap: Power management based on network packet processing mode transition for latency-critical workloads. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '21, page 143–154, New York, NY, USA, 2021. Association for Computing Machinery.

[3] Michał P. Karpowicz, Piotr Arabas, and Ewa Niewiadomska-Szynkiewicz. Design and implementation of energy-aware application-specific cpu frequency governors for the heterogeneous distributed computing systems. *Future Generation Computer Systems*, 78:302–315, 2018.

[4] Erfan Sharafzadeh, Seyed Alireza Sanaee Kohroudi, Esmail Asyabi, and Mohsen Sharifi. Yawn: A cpu idle-state governor for datacenter applications. In *Proceedings of the 10th ACM SIGOPS Asia-Pacific Workshop on*

Systems, APSys '19, page 91–98, New York, NY, USA, 2019. Association for Computing Machinery.