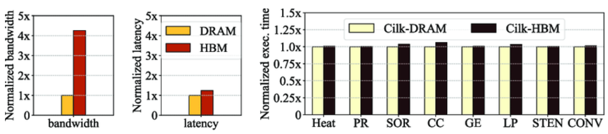
Fig 4 - uploaded by Yuxian Qiu



(a) Bandwidth (b) Latency (c) Performance

Bandwidth and latency of DRAM and HBM, and the impact of latency on application performance.

Source publication

+1

Bandwidth and Locality Aware Task-stealing for Manycore Architectures with Bandwidth-Asymmetric Memory

Article Full-text available

Dec 2018

Han Zhao · Quan Chen · Yuxian Qiu · [...] · Minyi Guo

Parallel computers now start to adopt Bandwidth-Asymmetric Memory architecture that consists of traditional DRAM memory and new High Bandwidth Memory (HBM) for high memory bandwidth. However, existing task schedulers suffer from low bandwidth usage and poor data locality problems in bandwidth-asymmetric memory architectures. To solve the two proble...

Contexts in source publication

Context 1

... 4(a) and 4(b) show the bandwidth and access latency of an HBM node normalized to the counterparts of a DRAM node measured with Intel MLC tool [1]. Observed from Figure 4, the bandwidth of an HBM node is 4.2× the bandwidth of a DRAM node, while the latency of an HBM node is 1.1× of the latency of a DRAM node. To show the impact of the slightly longer access latency of HBM nodes on application performance, we run all the benchmarks in Table 2 with one thread so that the required memory bandwidth is smaller than the bandwidth of a DRAM node. Figure 4(c) shows their performance in Cilk-DRAM and Cilk-HBM that store data in a DRAM node and an HBM node, respectively. ...

View in full-text

Context 2

... show the impact of the slightly longer access latency of HBM nodes on application performance, we run all the benchmarks in Table 2 with one thread so that the required memory bandwidth is smaller than the bandwidth of a DRAM node. Figure 4(c) shows their performance in Cilk-DRAM and Cilk-HBM that store data in a DRAM node and an HBM node, respectively. Observed from Figure 4(c), except CC, the slightly longer latency of HBM nodes does not degrade application performance. ...

View in full-text

Context 3

... show the impact of the slightly longer access latency of HBM nodes on application performance, we run all the benchmarks in Table 2 with one thread so that the required memory bandwidth is smaller than the bandwidth of a DRAM node. Figure 4(c) shows their performance in Cilk-DRAM and Cilk-HBM that store data in a DRAM node and an HBM node, respectively. Observed from Figure 4(c), except CC, the slightly longer latency of HBM nodes does not degrade application performance. This finding is consistent with the observation in prior work [35]. ...

View in full-text

Context 4

... 4(a) and 4(b) show the bandwidth and access latency of an HBM node normalized to the counterparts of a DRAM node measured with Intel MLC tool [1]. Observed from Figure 4, the bandwidth of an HBM node is 4.2× the bandwidth of a DRAM node, while the latency of an HBM node is 1.1× of the latency of a DRAM node. To show the impact of the slightly longer access latency of HBM nodes on application performance, we run all the benchmarks in Table 2 with one thread so that the required memory bandwidth is smaller than the bandwidth of a DRAM node. Figure 4(c) shows their performance in Cilk-DRAM and Cilk-HBM that store data in a DRAM node and an HBM node, respectively. ...

View in full-text

Context 5

... show the impact of the slightly longer access latency of HBM nodes on application performance, we run all the benchmarks in Table 2 with one thread so that the required memory bandwidth is smaller than the bandwidth of a DRAM node. Figure 4(c) shows their performance in Cilk-DRAM and Cilk-HBM that store data in a DRAM node and an HBM node, respectively. Observed from Figure 4(c), except CC, the slightly longer latency of HBM nodes does not degrade application performance. ...

View in full-text

Context 6

... show the impact of the slightly longer access latency of HBM nodes on application performance, we run all the benchmarks in Table 2 with one thread so that the required memory bandwidth is smaller than the bandwidth of a DRAM node. Figure 4(c) shows their performance in Cilk-DRAM and Cilk-HBM that store data in a DRAM node and an HBM node, respectively. Observed from Figure 4(c), except CC, the slightly longer latency of HBM nodes does not degrade application performance. This finding is consis- tent with the observation in prior work [35]. ...

View in full-text

Similar publications

Straggler-Aware Distributed Learning: Communication–Computation Latency Trade-Off

Article Full-text available

May 2020

Emre Ozfatura · Sennur Ulukus · Deniz Gündüz

When gradient descent (GD) is scaled to many parallel workers for large-scale machine learning applications, its per-iteration computation time is limited by straggling workers. Straggling workers can be tolerated by assigning redundant computations and/or coding across data and computations, but in most existing schemes, each non-straggling worker...

View

Citations

... Barghi et al. [35] designed a locality-aware work stealing based on the actor model and NUMA architectures. Many other methods [36]- [44] also have tackled NUMA-aware work stealing by increasing local data access to mitigate NUMA effects on remote task stealing. Instead of creating tasks beforehand, cooperative stealing [45], [46] utilizes the message-passingbased approach where victims create tasks only when the worker sends a stealing request, in order to avoid overhead caused by concurrent deques. ...

CAB-MPI: Exploring Interprocess Work-Stealing towards Balanced M…

Conference Paper

Full-text available

Nov 2020

Kaiming Ouyang · Min Si · Atsushi Hori · Zizhong Chen · Pavan Balaji

View

... HotSLAW [22] proposed to mitigate the rst issue by a heuristic that rst attempts to steal from a victim within a close proximity of the stealing worker. Many others try to address the second issue by repeating the same task mapping across multiple iterations [10,11,19,21, 35] . Obviously, they are applicable only to iterative applications. ...

... Some approaches use the structure of iterative applications to improve data locality [10,11,19,21, 35] . ADWS is di erent from these approaches because the application of ADWS is not limited to iterative applications. ...

Almost deterministic work stealing

Conference Paper

Nov 2019

Shumpei Shiina · Kenjiro Taura

With task parallel models, programmers can easily parallelize divide-and-conquer algorithms by using nested…

View

... In this paper, we design SysMon-H, an OS module (enhanced from [13,19, 28]) to collect the number of TLB misses for huge pages instead of merely relying on access_bit. The core idea of SysMon-H is from the observations that the cold pages (rarely accessed pages) have only a small number of TLB misses; In contrast, hot pages usually incur a large number of TLB misses, as they are frequently required to be loaded into TLB. ...

Thinking about A New Mechanism for Huge Page Management

Conference Paper

Full-text available

Jun 2019

Lei Liu

The Huge page mechanism is proposed to reduce the TLB misses and benefit the overall system performance. On t…

View

Task-Parallel Programming with Constrained Parallelism

Conference Paper

Sep 2022

Tsung-Wei Huang · Leslie Hwang

View

Spring Buddy: A Self-Adaptive Elastic Memory Management Sche…

Conference Paper

Dec 2021

Yihui Lu · Weidong Liu · Chentao Wu · Jia Wang · Minyi Guo

View

Enable simultaneous DNN services based on deterministic operator…

Conference Paper

Nov 2021

Weihao Cui · Han Zhao · Quan Chen · Ningxin Zheng · Minyi Guo

View

Extreme-scale ab initio quantum raman spectra simulations on the…

Conference Paper

Nov 2021

Honghui Shang · Fang Li · Yunquan Zhang · Libo Zhang · Dexun Chen

View

Taskflow: A Lightweight Parallel and Heterogeneous Task Graph…

Article

Aug 2021 · IEEE T PARALL DISTR

Tsung-Wei Huang · Dian-Lun Lin · Chun-Xun Lin · Yibo Lin

Taskflow aims to streamline the building of parallel and heterogeneous applications using a lightweight task…

View

place.