2019 IEEE 12th International Conference on Cloud Computing (CLOUD)

Evaluation of NUMA-Aware Scheduling in Warehouse-Scale Clusters

Richard Wu*, Xiao Zhang†, Xiangling Kong†, Yangyi Chen†, Rohit Jnagal† and Robert Hagmann†
*University of Waterloo, †Google
rzwwu@edu.uwaterloo.ca, {xiaozhang, xiangling, yangyichen, jnagal, rhagmann}@google.com

***Abstract*—Non-uniform memory access (NUMA) has been extensively studied at the machine level but few studies have examined NUMA optimizations at the cluster level. This paper introduces a holistic NUMA-aware scheduling policy that combines both machine-level and cluster-level NUMA-aware optimizations. We evaluate our holistic NUMA-aware scheduling policy on Google's production cluster trace with a cluster scheduling simulator that measures the impact of NUMA-aware scheduling under two scheduling algorithms, *Best Fit* and *Enhanced PVM (E-PVM)*. While our results highlight that a holistic NUMA-aware scheduling policy substantially increases the proportion of NUMA-fit tasks by 22.0% and 25.6% for both the *Best Fit* and *E-PVM* scheduling algorithms, respectively, there is a non-trivial tradeoff between cluster job packing efficiency and NUMA-fitness for the *E-PVM* algorithm under certain circumstances.**

I. Introduction

Machines with multiple NUMA memory nodes are a practical solution to vertical scaling and have become commonplace in warehouse-sized datacenters. However, NUMA machines come with performance implications: processes with remote NUMA memory observe significant run-time performance degradation due to slower memory accesses. Very little work has been conducted on NUMA optimization at the cluster scheduler level. The work of Tang et al. highlights the significant performance improvement of machine level NUMA locality across an aggregate of workloads at Google [1]. It follows that there are great opportunities to implement NUMA-awareness at the cluster scheduler level in addition to at the machine level. However, by introducing NUMA-awareness to cluster scheduling, we place an additional constraint on job schedulability which may negatively impact job packing efficiency. To address this uncertainty, we develop a NUMA-aware cluster simulator for two existing scheduling algorithms, *Best Fit* and *E-PVM*, used by Google's Borg cluster management systems [2]. Our results reveal a surprising finding, that NUMA-aware cluster scheduling with *E-PVM* can be unfavorable under certain circumstances.

II. NUMA-aware Scheduling

A. Scheduling/Scoring Algorithms

In Google's Borg cluster scheduler [2], two scheduling or scoring algorithms with complementary properties, *Best Fit* and Enhanced PVM (*E-PVM*) [3], are employed:

Best Fit Machines are filled as tightly as possible such that machines in use have high utilization and clusters tend to have fewer stranded resources.

E-PVM Tasks are spread out across machines resulting in a more balanced load distribution and more headroom on individual machines for bursty workloads.

B. Machine-level NUMA-aware Scheduling

When a task is scheduled on a machine, a NUMA-oblivious scheduler may arbitrarily allow it to run on any NUMA node. In contrast, a NUMA-aware scheduler can preferentially allocate CPU and memory from a NUMA-fit node for that task. Restricting a task to a particular NUMA domain allows it to benefit from memory locality, although it may suffer from constrained memory bandwidth, compared to a NUMA-oblivious scheduler which may optimize *for* memory bandwidth. To better understand this performance tradeoff, we select the top 100 jobs in terms of CPU usage in a production cluster consisting of tens of thousands of dual NUMA node machines and conduct an A/B test to compare their performance under both NUMA-aware and NUMA-oblivious schedulers. We quantify the job's performance in terms of instructions per CPU second which is measured by a CPU performance counter. As shown in Figure 1, the relative performance improvement of NUMA-aware scheduling indeed varies across jobs: 8% of jobs even exhibit performance degradation of which the worst experience 6% degradation under the NUMA-aware policy. However, the vast majority of jobs indeed observe a positive performance improvement and the average increase in performance across the 100 jobs is 7.4%. This result suggests NUMA-aware scheduling at the machine-level is indeed net positive for the most demanding jobs.

C. Cluster-level NUMA-aware Scheduling

We say a task is NUMA-fit if its requested CPU and memory resources are allocated to the same NUMA node (or physical CPU socket) on a machine. The corresponding NUMA nodes and machines on which NUMA-fit tasks are scheduled are called NUMA-fit nodes and NUMA-fit machines. In the context of a warehouse-scale cluster consisting of tens of thousands of machines, machine-level NUMA-aware scheduling is operating at the "whim" of a NUMA-

* Work performed during internship at Google.

2159-6190/19/$31.00 ©2019 IEEE
DOI 10.1109/CLOUD.2019.00082

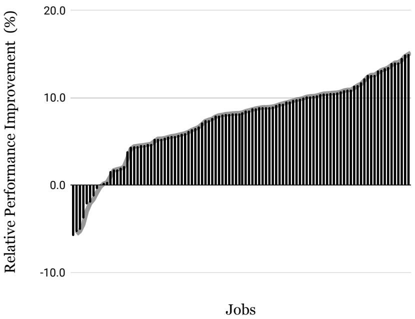


Figure 1: Machine-level performance improvement from NUMA-aware scheduler vs NUMA-oblivious scheduler.

oblivious cluster scheduler. As such we introduce cluster-level NUMA-aware scheduling, a scheduling heuristic at the cluster scheduler level that works in tandem with machine-level NUMA-aware scheduling to optimize for NUMA-fit tasks. Concretely, cluster-level NUMA-aware scheduling will first filter for NUMA-fit machines for an incoming job request. If no such NUMA-fit machines can be found, a machine is selected as per the usual policy.

III. Evaluation Methodology

A. Google Cluster Trace

Isolating the impact of a new scheduling policy in a production environment becomes unwieldy due to the complexities of a warehouse-scale cluster management system and the stochastic behavior of production workloads. To achieve deterministic scheduling results under different scheduling policies, it requires repeatedly draining machines which is quite disruptive to a production cluster. For that reason, we develop a cluster scheduling simulator and utilize a large-scale production cluster trace released by Google [4].

B. Cluster Simulator

We employ the use of a lightweight cluster simulator that ingests a trace of production data on machines and jobs (§III-A) and performs task scheduling that closely resembles the pipeline used in Google's Borg cluster management system. We defer details to the Borg paper [2].

IV. Evaluation Metrics

A. Job Packing Efficiency

While there are many cluster evaluation criteria, we choose job packing efficiency as noted in [5] as our primary focus since it heavily influences datacenter capacity planning and total cost of ownership (TCO). Adapting from the cluster compaction algorithm presented in [5], we apply a Monte-Carlo cluster *expansion* algorithm to approximate a lower bound on the number of machines required to schedule a set of jobs for a given cluster scheduling configuration.

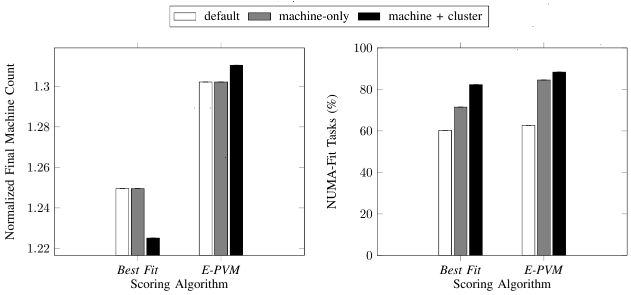


Figure 2: Cluster expansion results from five trials with the *Best Fit* and *E-PVM* algorithms (§II-A) as additional layers of NUMA-awareness are introduced. (Left) Job packing efficiency measured as normalized final machine count. (Right) Proportion of scheduled tasks that are NUMA-fit.

The cluster expansion algorithm first requires a workload that can over-saturate the target cluster of machines. Once a certain percentage of jobs τ enter the scheduling pending queue (*i.e.* there are no feasible machines available to accommodate new jobs), the algorithm expands current cluster with τ percentage of new machines and all pending jobs are rescheduled on the expanded cluster. We performed sensitivity study on τ to ensure it is sufficiently small.

B. NUMA-Fit Tasks Population

In our simulator, we perform bookkeeping to keep track of CPU and memory allocations on a per-task basis and the total number of NUMA-fit tasks. The population of NUMA-fit tasks is thus the percentage of NUMA-fit tasks at the end of a simulation run. We hypothesize that with additional layers of NUMA-awareness built into the schedulers, the population of NUMA-fit tasks should increase correspondingly.

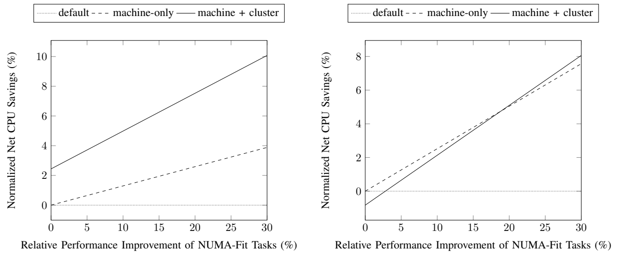
V. Evaluation Results

A. Impact on Job Packing Efficiency

Introducing *cluster-level* NUMA-aware scheduling places a more restrictive constraint on the scheduling process. Based on our simulations across five trials in Figure 2, *E-PVM* incurs a 0.8% increase (from 1.302 to 1.310) in final machine count when *cluster-level* NUMA-aware scheduling is introduced in the *machine + cluster* policy whereas *Best Fit* substantially benefits from a 2.4% decrease (from 1.249 to 1.225) in final machine count. *Cluster-level* NUMA-aware scheduling shares similar scheduling behavior with *Best Fit* since they both preserve empty NUMA nodes and hence enhances job packing overall.

B. Impact on NUMA-fit Tasks Population

Compared to the default policy, *machine-only* NUMA-aware scheduling observes a 11.2% (from 60.3% to 71.5%) and 21.8% (from 62.7% to 84.5%) net increase in NUMA-fit tasks for the *Best Fit* and *E-PVM* scoring algorithms, respectively. The results also confirm that *cluster-level*



(a) *Best Fit*

(b) *E-PVM*

Figure 3: Net CPU savings as a function of the performance improvement from NUMA-fit tasks across the three layers of NUMA-aware scheduling policies.

NUMA-aware scheduling does indeed additively improve the population of NUMA-fit tasks with an increase in the proportion of NUMA-fit tasks of 10.8% (from 71.5% to 82.3%) and 3.8% (from 84.5% to 88.3%) for *Best Fit* and *E-PVM*, respectively. The improvements from cluster-level NUMA-aware scheduling benefit the *Best Fit* algorithm almost three times as much as it benefits *E-PVM*. Cluster-level NUMA-fit filtering forces *Best Fit* to load-balance tasks across machines by occupying all the NUMA-fit nodes first before packing the tasks as tightly as possible under the *Best Fit* heuristic, overriding the tendency of *Best Fit* to suboptimally schedule tasks on NUMA-unfit machines early on. Compared to the default NUMA-oblivious scheduling policy, a *machine + cluster* NUMA-aware policy increases the proportion of NUMA-fit tasks by 22.0% and 25.6% for *Best Fit* and *E-PVM*, respectively.

C. Cost-Benefit Analysis

We observe the *Best Fit* algorithm experiences improvements in both job packing efficiency and NUMA-fit tasks population (§IV). *E-PVM* on the other hand presents a performance tradeoff where there is a cost associated with job packing for the benefit of higher yield on NUMA-fit tasks with the *machine + cluster* NUMA-aware policy. We correspond each performance metric to a unified CPU savings metric to better analyze the overall savings. (1) For job packing efficiency, we calculate how many additional (or fewer) CPU units cluster expansion requires for a given NUMA-aware policy with respect to the default policy. (2) NUMA-fit tasks experience $x\%$ CPU performance improvement on average. We calculate how many additional tasks are scheduled as NUMA-fit for a given NUMA-aware policy, then multiply the aggregate CPU units of the optimized NUMA-fit tasks by $x\%$ to approximate the CPU savings from the additional NUMA-fit tasks. We then combine the savings (or cost) from the two metrics. In Figure 3, we plot the net CPU savings for *Best Fit* and *E-PVM* as a function of the $x\%$ performance improvement experienced by NUMA-fit tasks compared to NUMA-unfit tasks, normalized to the default policy. We note that *Best Fit* unanimously benefits from NUMA-awareness scheduling at both the machine and cluster level. The use of the *E-PVM* scoring algorithm introduces performance tradeoffs between the scheduling policies, which manifest themselves as two intersection points in Figure 3b. The first intersection point at *x = 2.8%* between the *default* and *machine + cluster* policies highlights the point at which the *machine + cluster* NUMA-aware policy becomes strictly more beneficial than the default policy. The *machine-only* NUMA-aware policy outperforms the *machine + cluster* policy until the intersection point at *x = 18.8%*. Only when average run-time performance improvement from NUMA-fitness exceeds 18.8% does it outweigh the decrease in job packing efficiency under the *machine + cluster* NUMA-aware policy, relative to the *machine-only* NUMA-aware policy. From our experiments examining the real performance improvement from the most demanding jobs in a production cluster, we observe an average improvement of 7.4% (Section II-B) which is far less than the the 18.8% inflection point. It appears that a holistic cluster NUMA-aware scheduler under *E-PVM* may in fact be **less favorable in term of overall net savings**.

VI. Conclusion

This paper investigates (1) how cluster-level NUMA-aware scheduling can be implemented in tandem with machine-level NUMA-aware scheduling to improve the proportion of NUMA-fit tasks in production workloads and (2) the interaction between job packing efficiency and NUMA-fitness. Compared to a NUMA-oblivious policy, a *machine + cluster* NUMA-aware policy observes *22.0%* and *25.6%* net increase in the number of NUMA-fit tasks for the *Best Fit* and *E-PVM* scoring algorithms, respectively. A key tradeoff between job packing efficiency and NUMA-fitness under the *E-PVM* scoring algorithm is presented. From our observations of the most demanding tasks in a production cluster, enabling only the machine-level NUMA-aware policy under the *E-PVM* policy is most optimal whereas a holistic cluster NUMA-aware policy works best under the *Best Fit* policy.

References

[1] L. Tang, J. Mars, X. Zhang, R. Hagmann, R. Hundt, and E. Tune, "Optimizing google's warehouse scale computers: The numa experience," in *HPCA*, Feb 2013, pp. 188–197.

[2] A. Verma, L. Pedrosa, M. R. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-scale cluster management at Google with Borg," in *Proceedings of the European Conference on Computer Systems (EuroSys)*, Bordeaux, France, 2015.

[3] Y. Amir, B. Awerbuch, A. Barak, R. S. Borgstrom, and A. Keren, "An opportunity cost approach for job assignment in a scalable computing cluster," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 7, pp. 760–768, Jul 2000.

[4] J. Wilkes, "More Google cluster data," Google research blog, Nov. 2011, posted at http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html.

[5] A. Verma, M. Korupolu, and J. Wilkes, "Evaluating job packing in warehouse-scale computing," in *IEEE Cluster*, Madrid, Spain, 2014.