SATORI: Efficient and Fair Resource Partitioning by Sacrificing Short-Term Benefits for Long-Term Gains

M202173488 谢晨

背景介绍

在芯片多处理器(CMPs)上协同定位工作负载能够提高资源利用率，并降低运行其数据中心的资本和运营成本，工作负载也会增加，但这是一件好事，因为在数据中心中的工作一般是长期运行的，并没有严格的延迟要求。

虽然工作负载的协同定位可能潜在地增加吞吐量，但它可能导致不公平。同时让吞吐量和公平性达到优化目标仍然具有挑战性，因为吞吐量和公平性是根本上相互冲突的目标——优化一个往往会损害另一个。

除此之外，实现吞吐量以及公平性的最优化，有两个挑战，首先是共享资源（例如，高速缓存、内存等）在运行时同时实现优化目标。其次是在运行时同时为相互冲突的目标进行共同优化。

先前的一些实现优化目标的方法及存在的问题

dCAT：改变了在共同定位的工作负载之间动态分配的最后一级缓存(LLC)方式，以实现高吞吐量。

CoPart：单个资源的划分是单独完成的，但决策会被沟通，以达到一个全局最优的资源划分。

Heracles：将每个资源划分，在一维中执行(类似于梯度下降)

CoPart和Heracles方法中，同时联合勘探多个资源是必要的。但增加一种资源时，在对其他资源的分配被适当地调整大小之前，对给定作业的分配可能无法完全实现，这可能会导致搜索空间爆炸。优化一个目标是足够困难的了，但还需要挑战性地增加第二个目标。如果第二个目标与第一个目标相冲突（例如，提高吞吐量和公平性），那么这个两个目标的同时优化将变的更加具有挑战性。先前的工作，当优化公平性或吞吐量时，往往试图针对一个目标，并以最佳努力改进另一个目标。

Satori：第一个同时主动控制系统吞吐量和公平性目标的解决方案。

- 开发了一种基于贝叶斯优化(BO)理论的新方法，以智能地探索多资源划分配置空间。
- 建立了简单和准确的模型找到接近最优的解决方案，不需要离线分析、仪器、离线深度学习/强化训练或构建复杂的性能模型，因此可以避免可能引起的高开销和不可移植的问题
- 动态地、分别地监控系统级吞吐量和工作级公平指标，以构建一个针对多个目标的新的整体组合BO目标函数。同时通过边界保持传统BO的预期的对目标的动态重新优先级化的规模和周期。

Satori：设计与实现

- BO构建了一个低开销的代理模型来预测不同配置样本的性能，并使用有原则的、智能的搜索空间探索，对BO导向样本进行评估。BO的基础代理模型在设计上并不完全准确，而是不断地变得更加准确，适合在线使用。
- Satori仔细地构建了目标函数，它包含了同时实现多个目标(包括公平性和吞吐量)，并且其功能被设计成可扩展的。

$$x^* = \underset{x \in \mathcal{X}}{\operatorname{argmax}} f(x) \tag{1}$$

BO是一种基于理论基础的解决最大化黑盒目标函数问题的方法。
BO不需要知道输入和目标函数之间的关系。

- satori可以通过使用给定的配置运行系统并观察所有共定位作业的性能，来评估给定输入x（资源分区配置）的目标函数f的值。目标函数的值表示价值数字的质量（例如，吞吐量或公平性）。

$$x^* = \underset{x \in \mathcal{X}}{\operatorname{argmax}} f(x) \qquad (1)$$

- BO使用一个代理模型M(x)来随机估计空间中不同配置的性能。

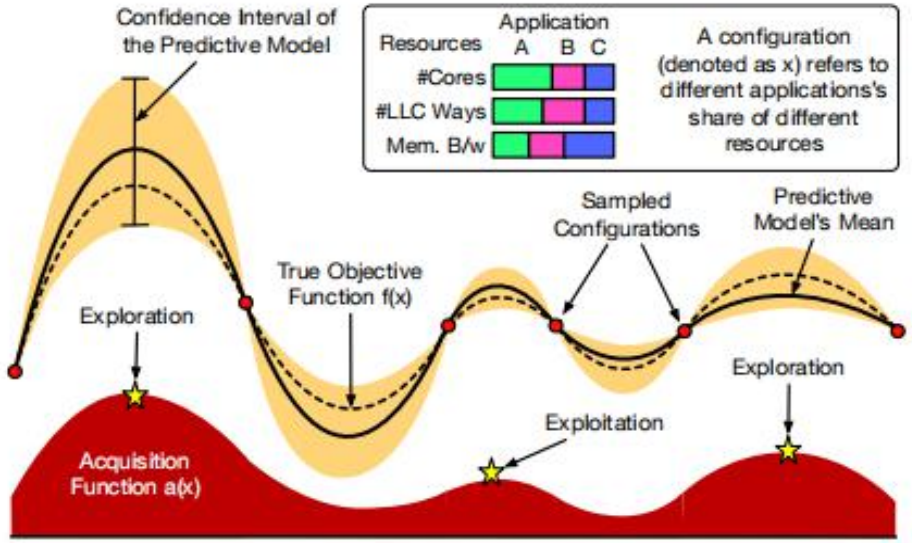


Fig. 4. Visual depiction of Bayesian optimization (BO) components including configuration, true objective function, proxy model, and acquisition function.

Algorithm 1 SATORI BO Engine Algorithm.

Input: Initial resource configurations (S_{init}).
Run the system with S_{init} & record throughput and fairness.
Record baseline (isolation) performances.
while TRUE **do**
    Generate objective function (details in Sec. III-B and III-C).
    Update the proxy model $\mathcal{M}(x)$.
    Compute the acquisition function $a(x)$.
    Optimize $a(x)$ and find the next sample to evaluate.
    Run the system with the selected configuration x.
    Record throughput and fairness for x.
        & add these values to the existing set.
    if End of a job **or** Start of new job **or** Every reset interval
        Reset baseline (rerecord isolation performances).

生成目标函数
更新代理模型M（X）
计算采集函数a(x)
优化一个a(x)，并找到下一个要进行评估的样本。
使用所选的配置x运行系统。
记录x的吞吐量和公平性。
并将这些值添加到现有的集合中。

- 与传统的BO不同，Satori需要同时评估多个性能（公平性和吞吐量），所以需要构建一个全新的目标函数来指导分配给配置的性能分数。

$$f(x) = \sum_{i=1}^{K} W_i \times \mathrm{Goal}_i(x) = W_T \times \mathrm{T}(x) + W_F \times \mathrm{F}(x) \quad (2)$$

WT是吞吐量T(x)的权重，WF为公平性F(x)的权重。

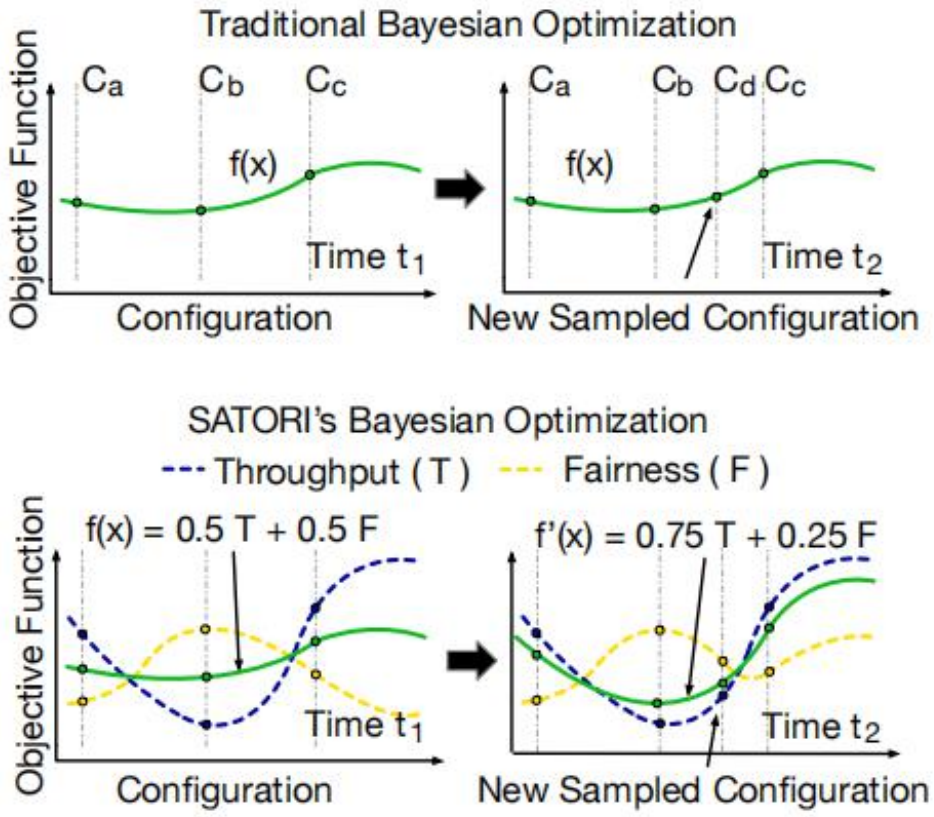


Fig. 5. SATORI maintains separate throughput and fairness performances and constructs a fresh objective function every iteration based on dynamic weights.

与传统BO相比，SATORI通过动态改变权重来维持吞吐量和公平性

- 同时，为了避免每次更新目标函数时都需重建目标函数代理模型的麻烦，SATORI采用了一种新技术，这使得配置依赖的各种单一冲突目标的表现的单独记录得以维持。

- 为了实现动态的改变吞吐量以及公平性的权重，SATORI引入了以下的权重平衡

$$W_{TE} = \frac{1}{2}t_e - \sum_{i=1}^{t_e} W_{T_i} \quad \& \quad W_{FE} = \frac{1}{2}t_e - \sum_{i=1}^{t_e} W_{F_i} \quad (3)$$

如果一个目标已经获得了较高的权重，那么它的权重将被削减

$$W_{TP} = \frac{1}{4} + \frac{1}{2}\frac{\Delta_F}{\Delta_T + \Delta_F} \quad \& \quad W_{FP} = \frac{1}{4} + \frac{1}{2}\frac{\Delta_T}{\Delta_T + \Delta_F} \quad (4)$$

如果一个目标充分利用了它的权重，那么它的将获得高权重

$$W_T = \frac{t_e}{T_E} W_{TE} + \left(1 - \frac{t_e}{T_E}\right) W_{TP} \tag{5}$$

$$W_F = \frac{t_e}{T_E} W_{FE} + \left(1 - \frac{t_e}{T_E}\right) W_{FP} \tag{6}$$

实验的效果

TABLE I. PARSEC benchmarks [6] used in this study.

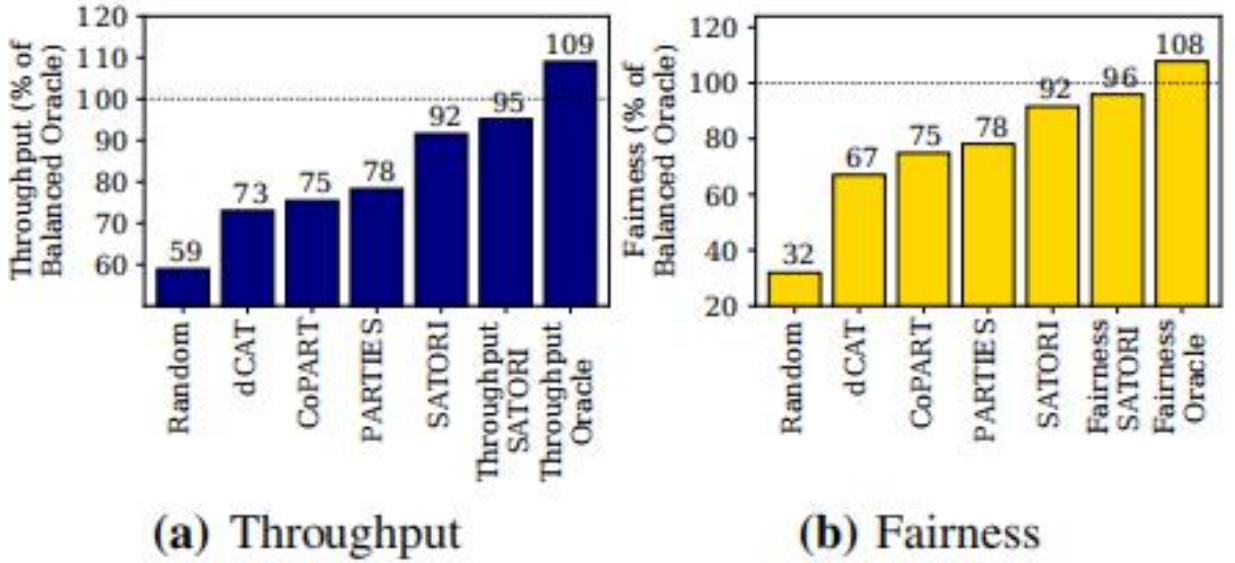
Blackscholes	Option pricing with Black-Scholes Partial Differential Eq.
Canneal	Simulated cache-aware annealing to optimize chip design
Fluidanimate	Fluid dynamics for animation with Smoothed
Freqmine	Frequent itemset mining
Streamcluster	Online clustering of an input stream
Swaptions	Pricing of a portfolio of swaptions

TABLE II. CloudSuite benchmarks [22] used in this study.

Data Analytics	Naive Bayes classifier on Wikipedia entries
Graph Analytics	Page ranking on Twitter data
In-memory Analytics	In-memory filtering of movie ratings
Media Streaming	Nginx server to stream videos
Web Search	Web search algorithm implementation

将其性能分别与随机搜索，dCAT，Copart,PARTIES,甲骨文，通过吞吐量和公平SATORI搜索对比

- SATORI在竞争技术方面表现出色，比排名第二的策略高出14%的吞吐量和公平性。



(a) Throughput **(b)** Fairness

Fig. 7. SATORI achieves better throughput and fairness than other techniques (averaged across different job mixes, PARSEC benchmarks).

SATORI的优势

- SATORI对优先级周期和具有宽范围的均衡周期不敏感。

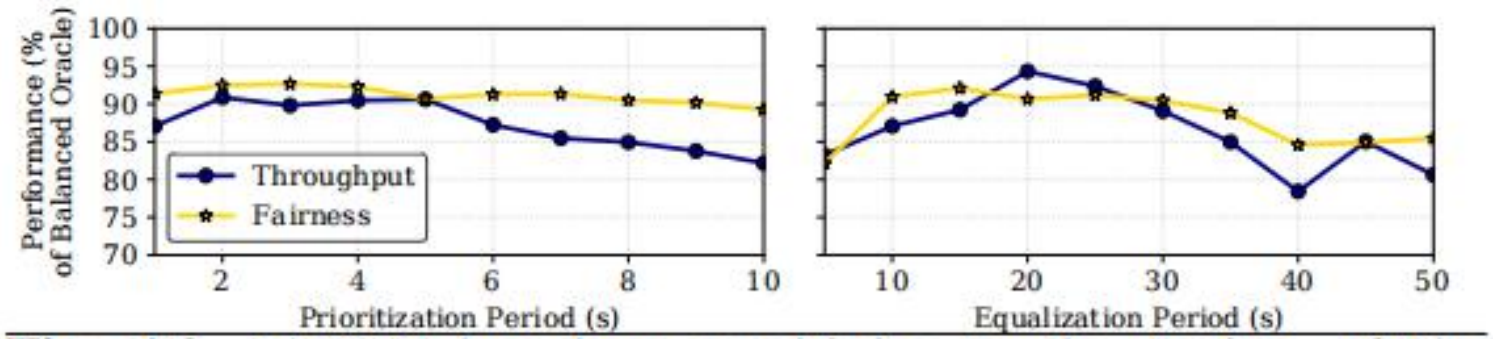


Fig. 16. SATORI has low sensitivity to the tuning of the prioritization period and the equalization period. It does not require large tuning efforts to achieve near-optimal results.

- SATORI对新目标函数的设计有助于其提取高性能（通过权重和公平性），但并不会使BO表现得出人意料。

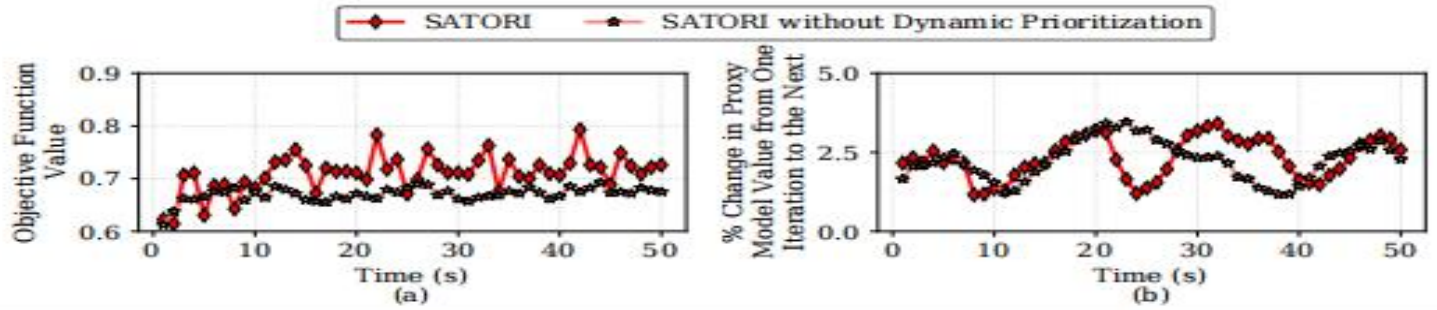


Fig. 17. New objective function construction in SATORI improves the quality of the solution by obtaining (a) higher objective function values, but without (b) unexpected changes in the underlying BO operations.

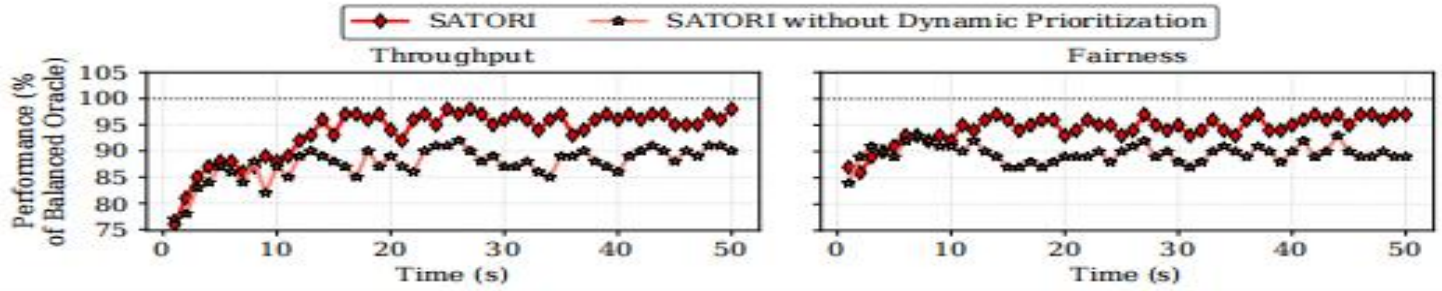


Fig. 18. Variation in the observed performance for both goals is similar for SATORI and SATORI without prioritization.

- SATORI适用于真实系统并且其开销很低。
- 在SATORI中，最耗时的是BO引擎，100ms中测量范围内，BO也只占用1.2ms。
- 实验显示SATORI仅执行了大约1%的工作组合在故障时的指令。

实验得出的结论

- satori能够主动并同时控制多个CMP体系结构资源以实现多个目标的技术。可以有效地处理计算核心、LLC方式、内存带宽和功率限制资源，以实现吞吐量和公平性之间的最佳平衡。

谢谢