易失性实例上的机器学习：模型收敛，运行时间和成本的权衡

Machine Learning on Volatile Instances: Convergence, Runtime, and Cost Tradeoffs

Introduction

随着越来越复杂的模型被应用在越来越多的数据上，往往采用分布式并行的方法来减少单机训练的负担，同时加快模型的训练速度。但是，这样的训练模型往往需要大量的GPU作为硬件基础，即使是采用云计算的方式租赁应用实例，**成本**往往也是难以负担的，因为模型的训练往往会持续几小时或几天。

一种常见的方法是使用**抢占性的易失性实例**来降低成本。这种易失性实例相较于可用性保证实例，往往价格更为低廉，但是会有**任务被抢占的风险**存在。

同时，亚马逊提供了出价的方式，通过调整出价方案，用户能够得到额外的灵活度。但这种易失性服务器的价格不断波动的，而且模型训练的中断会引起模型训练的误差，需要引入更多的迭代次数或者工人数量来弥补这种错误，这同样带来了额外的训练成本。

本文参考亚马逊上的易失性实例，通过权衡训练错误，训练时长以及成本，在亚马逊提供的出价规则下，分析并提出了一种降低训练成本的出价策略。

更为具体的工作为：

- 验证了工人数量在动态环境下的训练误差
- 优化了实例的出价策略
- 优化了工人数量
- 在亚马逊上验证了策略优化后的效果

Related Work

DML通常是worker进行本地梯度训练，并将梯度发送给中央服务器(PS-Worker架构)。最近相关的工作主要集中在通过减少通讯开销来进行模型训练加速，或者分析mini-batch的大小和学习率来对模型误差的影响。本文的思想和和其他工作相似，但引入了训练成本作为权衡指标，同时考虑模型在易失性实例上的表现。

其他对瞬态资源的利用研究工作中并未对DML这种分布式工人之间的依赖关系进行考虑，不能完全适用于易失性实例上的DML。

Error and Runtime Analysis of Distributed SGD with Volatile Workers

distributed SGD

- Synchronous Distributed SGD

同步随机梯度下降算法大体上分为以下三种：

Synchronous Distributed SGD, N-synchronous SGD, N-batch-synchronous SGD

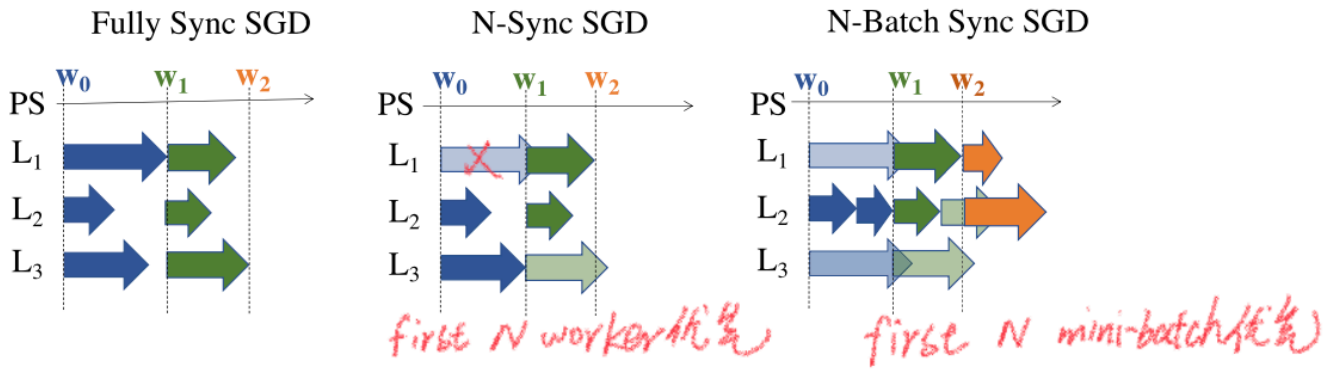


Fig. 1. Gradient computations on three workers for two iterations in fully, $N = 2$-, and $N = 2$-batch synchronous SGD. The x-axis indicates time, and lighter colored arrows indicate workers cancelled by the Parameter Server (PS).

- Distributed SGD on Volatile Workers

在易失性实例上的Distributed SGD以另一种方式安排：Parameter Server是一个持久化的实例，而每次迭代过程都会有若干worker进行梯度计算。

易失性实例对模型收敛的影响

通过对易失性实例上进行**随机工人数量和模型收敛度关系**的数学推导，文章得到了以下结论。第J次迭代后的损失函数为：

$$E(G(\mathbf{w}_{J+1}) - G^*) \leq (1-\alpha c\mu)^J E[G(\mathbf{w}_1)] + \frac{\alpha^2 LM}{2} \sum_{j=1}^{J} (1-\alpha c\mu)^{J-j} E[\frac{1}{y_j}]$$

- 当每次迭代中工人数量不再随机变化且为预期值时，误差界限减小；此时工人实例不再是易失性实例。
- 频繁的抢占或中断工人会产生更差的误差收敛。

在full-sync SGD中，第J次迭代后的损失误差可以表示为：

$$E[G(\mathbf{w}_{J+1}) - G^*] \leq (1-\alpha c\mu)^J E[G(\mathbf{w}_1)] + \frac{B}{n_1^\chi} \cdot (1-\alpha c\mu)^{J-1} \cdot \frac{1-x^J}{1-x}$$

在full-sync SGD中，虽然公式证明了工人数量和模型收敛的关系，但往往人们会随着时间的推移增加工人数量，这是由于在模型训练**后期**，梯度趋近收敛，此时梯度的准确性尤为重要。反之，在训练**前期**使用较少的工人数量能够进一步减少训练花费。

若工人数目递增，且最大工人数量为n_{max}时，损失误差可以表示为：

$$E[G(\mathbf{w}_{J+1}) - G^*] \leq (1-\alpha c\mu)^J E[G(\mathbf{w}_1)] + B \cdot \max\{\frac{(1-\alpha c\mu)^{J-1} 1-x^J}{n_1^\chi (1-x)}, \frac{1-(1-\alpha c\mu)^J}{n_{max}^\chi \alpha c\mu}\}$$

通过对**递增工人数量和模型收敛度关系**的推导，文章证明了随着工人数量的增加是可以降低模型收敛度的，进而得出推论：使用额外的工人能够减少迭代次数。同时也对递增数量的上限和模型收敛的关系给出了论证。

值得注意的是，**以上推论只适用于full-sync SGD**，这是因为每次迭代中引入更多的工人节点能够很好的降低训练梯度的方差，而N-sync和N-batch-sync只是降低了训练时间，对训练方差并无影响。

- 对于SGD的变体，我们可以通过设置N和N-batch的关系来达到上述的效果。
- 证明了静态递增工人步长能够获得最快的误差下降趋势。

在易失性实例上的运行时间影响

由于后台进程，节点中断，网络延迟等影响，节点的计算时间是在不断波动的。通常，在易失性实例上的运行时长可以表示为：

$$E[\tau] = \sum_{j=1}^{J} E[R(y_j)] + E[idle\ time\ with\ no\ active\ workers].$$

通常，当节点收到中断请求是，会立即将梯度计算的中间值存储起来；当下次迭代开始而节点仍未从中断中恢复时，节点将放弃该中间值。

Optimizing Spot Instance Bids

统一出价

在这里，我们给所有的工人出价为b。出价b只影响迭代的频率而不影响每次迭代中工人的数量。

通过分析运行时间/运行花费和出价的关系，文章指出迭代次数和期望的运行时间与运行花费呈现正相关性。

优化方案：

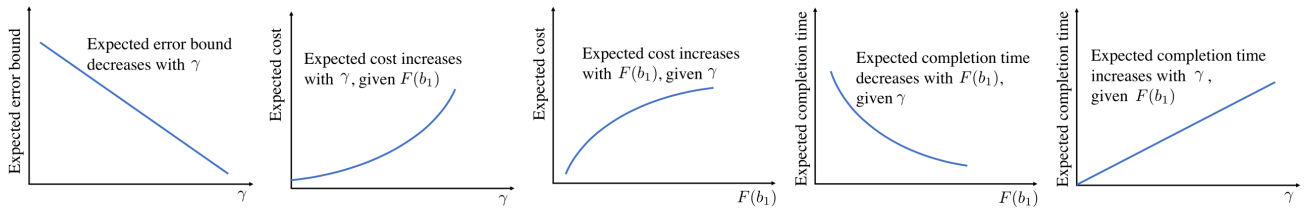
> 根据亚马逊的出价策略，出价b是在不知道未来实例现价的情况下指定的，并且适用于整个工作周期。虽然在工作中能够更改出价规则，但是会带来额外的迁移开销。
>
> 因此，文章提出对每一个实例进行持久化请求，实例的现货价格低于出价时，则执行任务。任务不会在实例之间进行迁移。

不同出价

在统一出价的基础上，文章又提出了易失性实例的不同出价策略。

不同出价策略的提出是由于发现了现货市场上实例价格的相关性，当一些实例的价格较低时，有很多实例的价格也是相对较低的，使用这些实例能够很好的降低训练误差和花费。

文章中为了简化分析，采用两级出价规则：一部分节点给予较低的出价b_l，一部分节点给予较高的出价b_h。在这种出价规则下，文章通过数学分析，对训练时间、训练代价以及误差边界的相关性进行了解读。



(a) Error-vs-γ **(b)** Cost-vs-γ **(c)** Cost-vs-$F(b_h)$ **(d)** Compl. time-vs-$F(b_h)$ **(e)** Compl. time-vs-γ

Fig. 3. Illustration of how the expected cost, completion time and error vary w.r.t. $F(b_h)$ and $\gamma = \frac{F(b_l)}{F(b_h)}$. As a larger γ leads to a smaller expected error (Fig. 3a) but a larger expected cost (Fig. 3b) and completion time (Fig. 3e), and the expected error is only controlled by γ, the optimal γ should be the smallest possible γ, *i.e.*, the one that yields error $= \epsilon$. The optimal $F(b_h)$ should be the one that yields the completion time equal to the deadline under the optimal γ (Fig. 3d).

文章对于不同SGD算法也进行验证分析，从理论上分析了价格区间的选定与模型最终花费的关系。

Optimizing Number of preemptible Instances

不同平台的购买规则也是不一样的，像GCP（Google Cloud Platform)这种平台，用户只能指定抢占实例的数量而不能进行出价。

文章根据上文提到的模型收敛和工人数目的联系，提出了工人数目和迭代次数的优化，具体为随着迭代次数的上升，工人数目以指数形式增加。

$$J^* = \min\{\arg\min_{J\in\{J_1,J_2\}} \frac{BJ(1-\beta^J)}{(1-\beta)(\epsilon - A\beta^J)}, \lfloor\theta\delta\rfloor\},$$

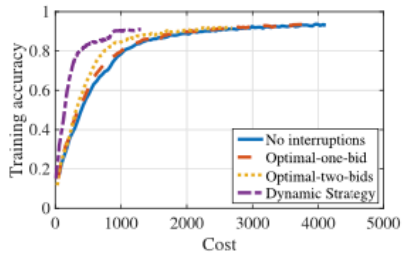
$$J_1 = \lfloor\tilde{J}\rfloor, J_2 = \lceil\tilde{J}\rceil, \frac{A\beta^{\tilde{J}}(\tilde{J}ln\frac{1}{\beta}+1-\beta^{\tilde{J}})}{1+\beta^{\tilde{J}}(\tilde{J}ln\frac{1}{\beta}-1)} = \epsilon,$$

$$n^* = \lceil\frac{B(1-\beta^{\tilde{J}})}{(1-\beta)(\epsilon - A\beta^{\tilde{J}})}\rceil,$$

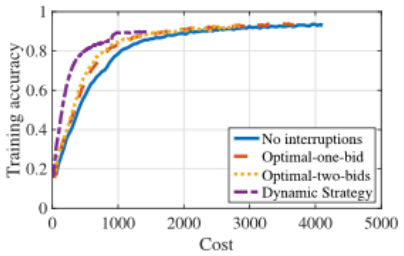
在分别考虑stragglers有无影响的条件下，文章对工人指数增长的参数η和迭代次数J的关系进行了分析。具体是为每个实例假设有p的使用几率，结果表明上述结论在此情况下不受影响。

Experimental Validation

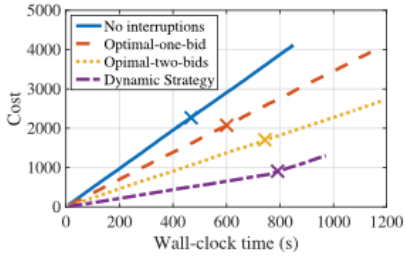
出价策略的优势



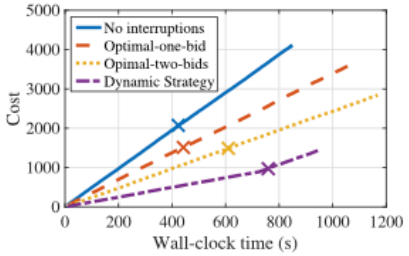
(a) Accuracy-vs-cost, uniform spot price distribution



(b) Accuracy-vs-cost, Gaussian spot price distribution

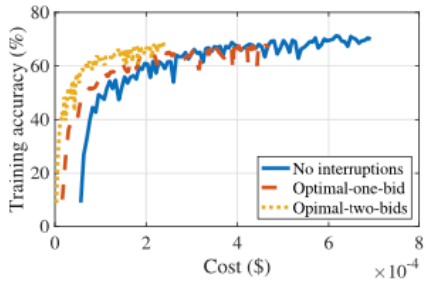


(c) Cost-vs-time, uniform spot price distribution

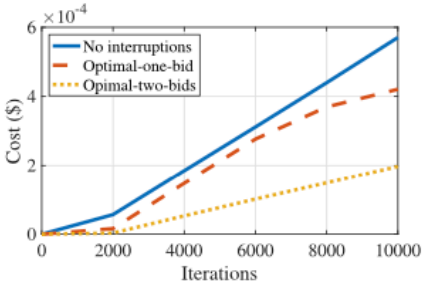


(d) Cost-vs-time, Gaussian spot price distribution

Fig. 4. The dynamic strategy (a,b) achieves the highest training accuracy under any given cost using the ResNet-50 model for CIFAR-10 classification, under synthetic spot prices. The markers on the curves in (a,b) show the cost when achieving a 90% training accuracy; at which point No-interruptions, Optimal-one-bid, and Optimal-two-bids respectively increase the cost by 134%, 82%, 46% under the uniform distribution, and 103%, 101%, 43% under the Gaussian distribution relative to the dynamic strategy.



(a) Accuracy-vs-cost



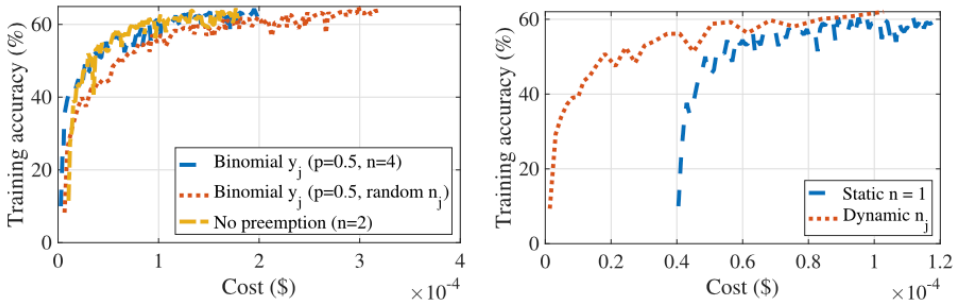
(b) Cost-vs-iterations

Fig. 5. Under historical price traces of the c5x.large spot instances in the region of us-west-2a (Oregon) and using a small CNN for CIFAR-10 classification, Optimal-one-bid and Optimal-two-bids can reduce the cost by 26.27% and 65.46% respectively compared with No-interruptions (Figure 5a) while achieving 96.78% and 96.46% of the training accuracy that No-interruptions achieves (Figure 5b).

实验主要揭示了在易失性实例上的模型训练能够获得极大的效益。

数量策略的优势

- 在亚马逊平台上对工人数量、模型准确率和花费进行分析：



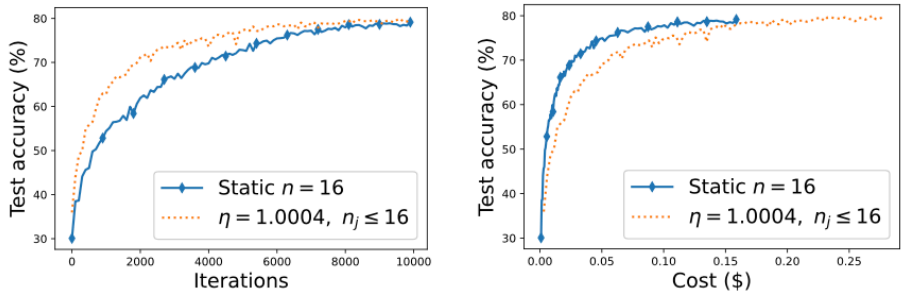
(a) Accuracy-vs-cost varying preemption probability and n

(b) Accuracy-vs-cost: static-vs-dynamic strategies

Fig. 6. Using n estimated based on Theorem 7 achieves higher accuracy per dollar than randomly setting n (Figure 6a). Compared with using 1 worker for $J = 10000$ iterations, dynamically setting $n_j = 1.0004^{j-1}$ and the number of iterations according to Theorem 2 with $\chi = 1$ achieves higher accuracy per dollar on EC2 spot instances.

揭示了易失性实例具有更大的训练效益。

- 在GCP上对动态工人数目的分析：



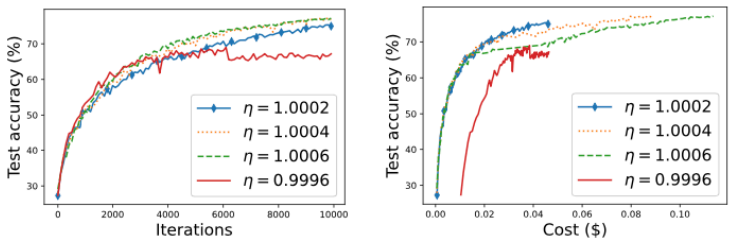
(a) Accuracy-vs-iterations

(b) Accuracy-vs-cost

Fig. 7. Compared with using 16 workers, dynamically setting $n_j = \min\{1.0004^{j-1}, 16\}$ achieves the same accuracy after 10000 iterations (Figure 7a), but can reduce the cost by at least 46% (Figure 7b).

动态工人数目能够使得模型更加快速收敛，但是对应的单位花费随之变高。

- 针对工人数目的分析：



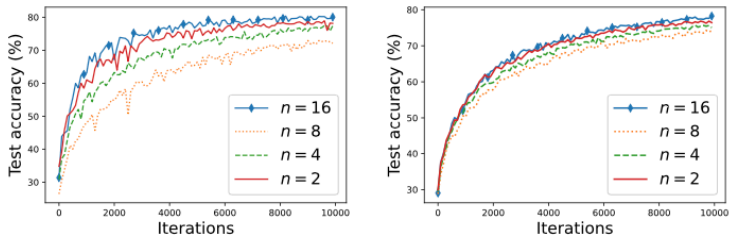
(a) Accuracy-vs-iterations

(b) Accuracy-vs-cost

Fig. 8. Using a ratio $\eta > 1$ to set the number of workers according to $n_j = \min\{\eta^{j-1}, 16\}$ for our dynamic strategy achieves a higher accuracy (Figure 8a) and higher accuracy-per-dollar (Figure 8b) than using $\eta = 0.0096$, which decreases the number of workers.

当工人数目随着迭代次数而增加时，模型准确率也在快速增加；反之，模型准确率难以上升甚至会下降，而且单位收益也很差。

- 文章对于步长选取对模型的影响也做了分析（线性/静态）：

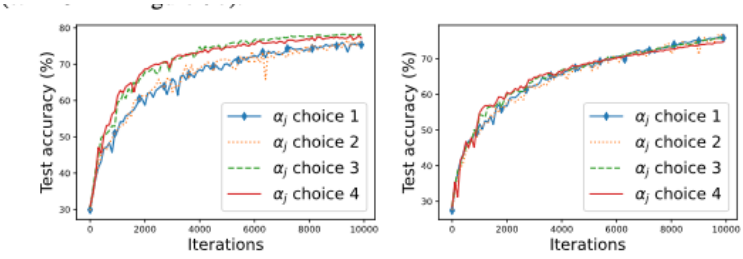


(a) Linearly scaled step sizes **(b)** Identical step size

Fig. 9. Linearly scaling the step size with the number of provisioned workers ($\alpha = 0.03 \times n$ in Figure 9a) improves the accuracy convergence rate compared to using the same step size when varying the number of provisioned workers ($\alpha = 0.1$ in Figure 9b).

实验证明，和工人数目相关的步长能够很好的加快模型的收敛速度，这和独立于工人数目的步长是不同的。

同时，当预置工人数目一定时，逐渐降低的步长能够提高模型的收敛速度；而对于动态工人数目这种现象则不会出现。



(a) Static number of provisioned workers **(b)** Dynamic number of provisioned workers

Fig. 10. Using a diminishing step size (α_j choice 3 or 4) improves the accuracy when using a static $n = 4$ provisioned workers (Figure 10a), but does not significantly affect the dynamic strategy which increases the number of workers with a constant rate ($\eta = 1.0002$ in Figure 10b).

其中，4种步长策略分别是

1). $\alpha_j = 0.1$;

2). $\alpha_j = \min(0.03 \times 1.2^{\lfloor \frac{j}{1000} \rfloor}, 0.18)$;

3). $\alpha_j = 0.03 + \frac{0.15}{\lfloor \frac{j}{1000} \rfloor + 1}$;

4). $\alpha_j = \frac{0.18}{\lfloor \frac{j}{1000} \rfloor + 1}$

实验证明，固定预设工人的情况下，步长方案3，4能够一定程度上优化模型收敛；而对于动态工人来说，模型是和步长相独立的。

Thinking

优势

- 通过数学分析的方式揭示了工人数目与模型训练的关系，对于DMLsys的优化提供了一种新的思路。
- 文章在动态工人的基础上，给出了几种不同的模型优化方向。

劣势

- 文章很大程度关注了模型训练的花费，但这种价格优惠方案很大程度上依赖于平台的策略。
- 采用理论分析-实验论证的方式，但对异常数据没有作出细致的解释。

对于DML应用在边缘计算和雾计算中，描述了一种可能性。