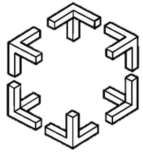


Distributed Computing and Systems
Chalmers University of Technology

CHALMERS

ASAP.SGD: Instance-based Adaptiveness to Staleness in Asynchronous SGD

Karl Bäckström Marina Papatriantafilou Philippas Tsigas

CHALMERS
UNIVERSITY OF TECHNOLOGY

WASP | WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

ASAP.SGD: Instance-based Adaptiveness to Staleness in Asynchronous SGD

Background

Focus: *Asynchronous parallel SGD (AsyncSGD)*

- SGD is a sequential iterative numerical optimization algorithm
- Effective non-convex problems *(read: deep learning)*
- Recently, *asynchronous* parallelization has gained traction
- Asynchrony gives leads to (i) **computational efficiency**
  (ii) **asynchrony-induced noise**

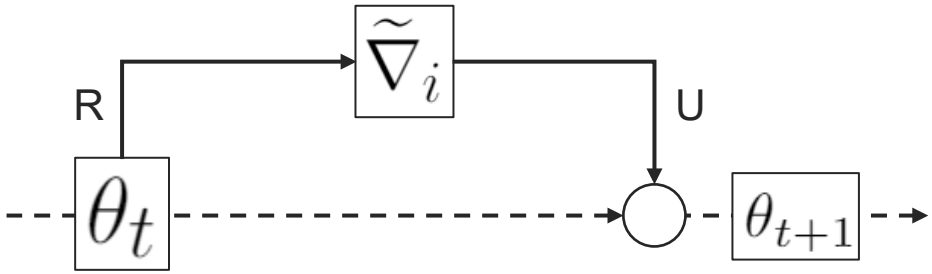
SGD:

Initialize θ_0

Iterate $\theta_{t+1} = \theta_t - \eta \widetilde{\nabla} f(\theta_t)$

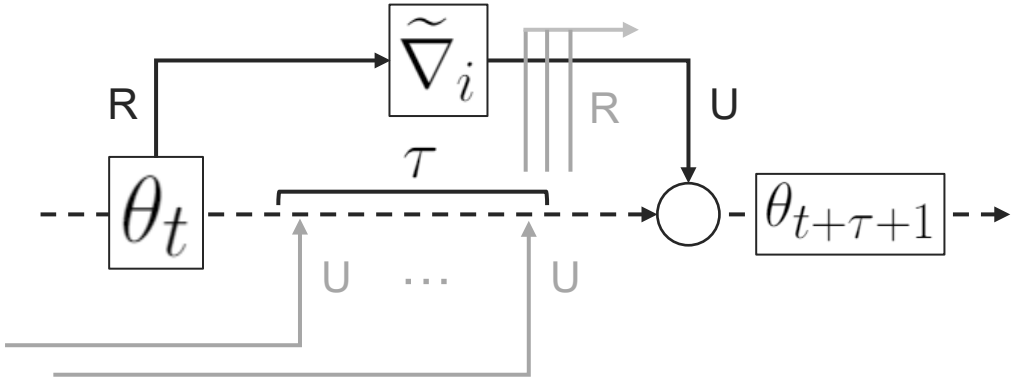
ASAP.SGD: Instance-based Adaptiveness to Staleness in Asynchronous SGD

Background



SGD iteration:

$$\theta_{t+1} = \theta_t - \eta \widetilde{\nabla} f(\theta_t)$$

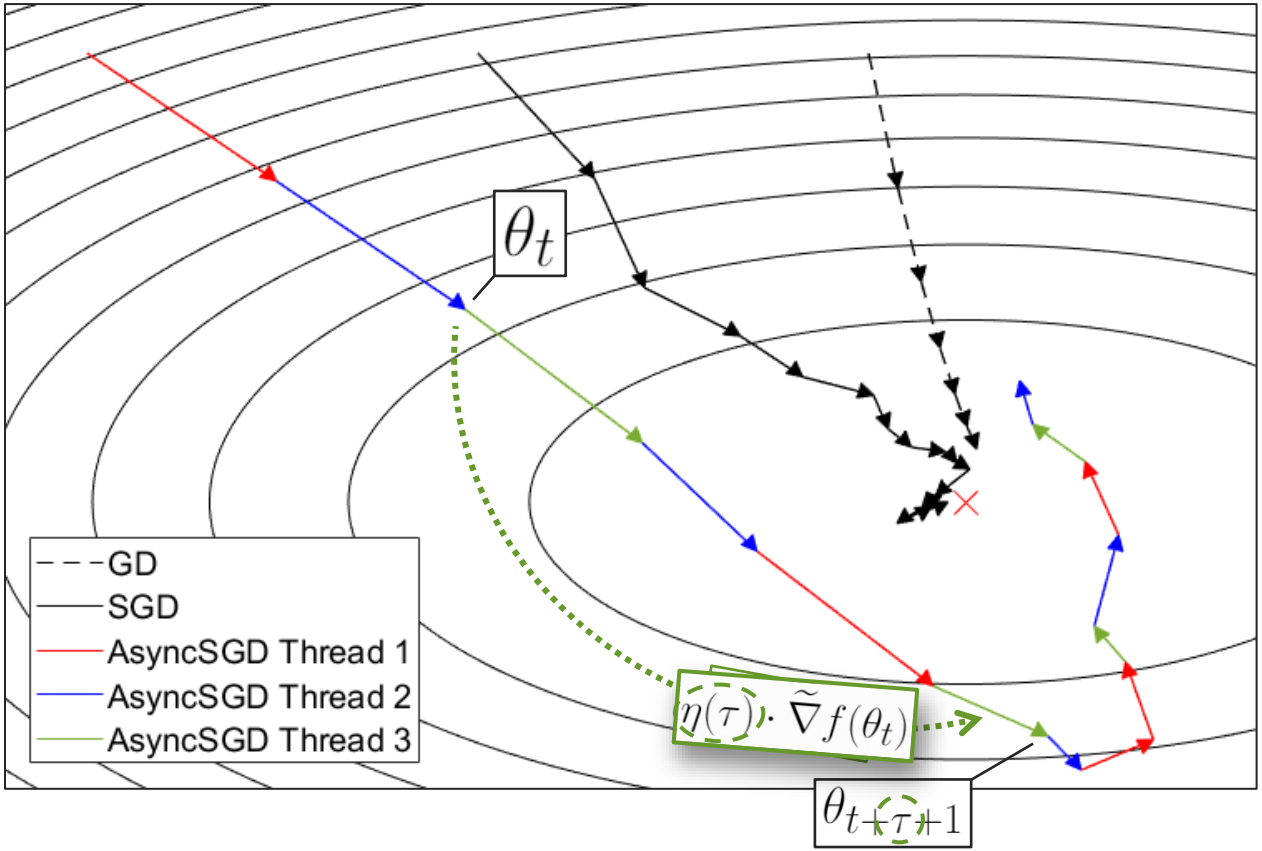


AsyncSGD iteration:

$$\theta_{t+\tau+1} = \theta_{t+\tau} - \eta \widetilde{\nabla} f(\theta_t)$$

ASAP.SGD: Instance-based Adaptiveness to Staleness in Asynchronous SGD

Background



$\exists$ staleness

$\Rightarrow$

overshooting

$\approx$

asynchrony-induced noise

ASAP.SGD: Instance-based Adaptiveness to Staleness in Asynchronous SGD

Background

Static dampening

- Pre-defined heuristic rule for dampening based on staleness

Static dampening

$$\eta(\tau) = \eta_0/\tau$$

$$\eta(\tau) = \eta_0 \cdot e^{-\beta\tau}$$

Constant (standard)

$$\eta(\tau) = \eta_0$$

Underlying factors:

- UMA/NUMA
- Algorithmic implementation
- Synchronization mechanism
- N.o. workers

✗ **Changes overall step size magnitude**

✗ **Step size vanishes for high parallelism**

ASAP.SGD: Instance-based Adaptiveness to Staleness in Asynchronous SGD

Contributions

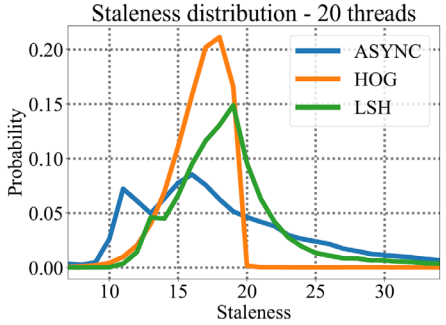
The ASAP.SGD theoretical framework for staleness-adaptiveness, that ensures:

- Overall step size magnitude is preserved
  (i) Step size sensitive applications (such as DL)
  (ii) Comparability between methods

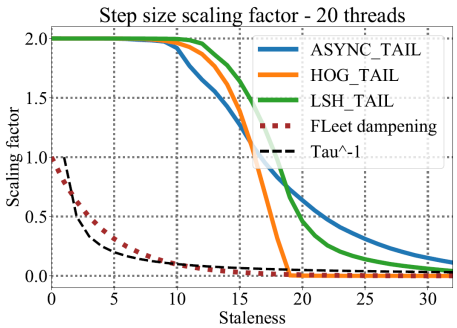
Within ASAP.SGD, we introduce the TAIL-τ instance-based staleness-adaptive step size

- Utilizes the overall staleness distribution $\mathrm{PDF}(\tau)$ to dynamically compute a tailored staleness adaptive step size function

Underlying factors ⇒



TAIL-τ ⇒



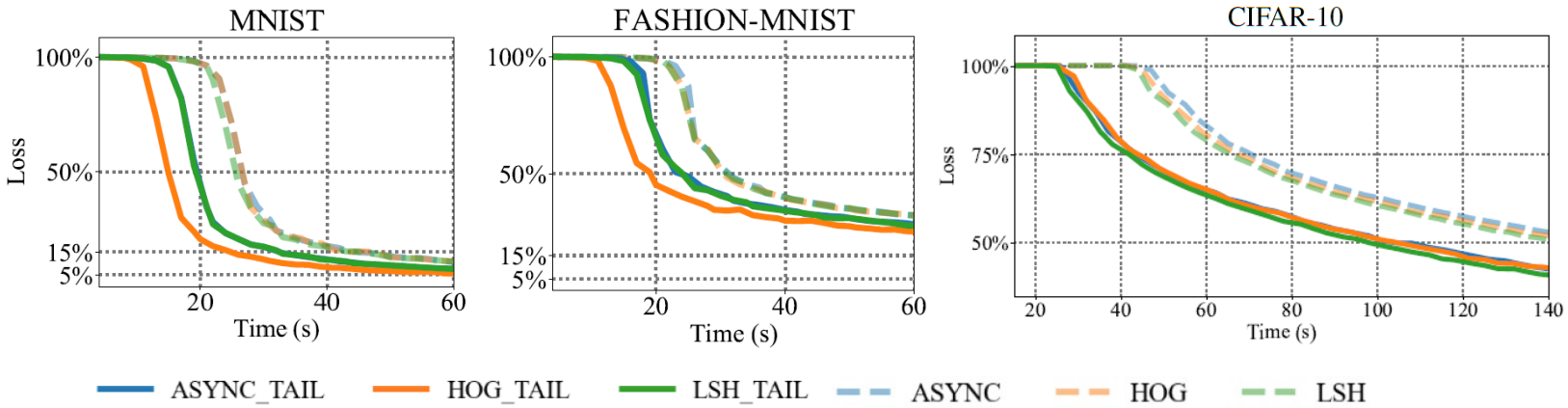
ASAP.SGD: Instance-based Adaptiveness to Staleness in Asynchronous SGD

Evaluation

Benchmark on DL problems:

- Datasets: *CIFAR, MNIST, Fashion-MNIST*
- Architectures: *LeNet, 3-layer MLP*
- AsyncSGD algorithms: *Lock-based AsyncSGD, Hogwild, Leashed-SGD*

29% *speedup on average, for LeNet training on CIFAR-10*



ASAP.SGD: Instance-based Adaptiveness to Staleness in Asynchronous SGD

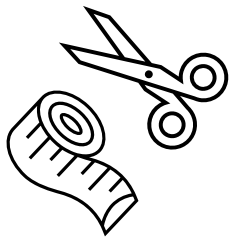
Conclusion

ASAP.SGD

- Framework that guides design of new staleness-adaptiveness step size functions
- Formulates desirable analytical properties on $\eta(\tau)$
- Establish convergence results on convex, non-convex, and Polyak-Lojasiewicz target functions

TAIL-τ *Open source*

- Instance-based (execution-tailored), staleness-adaptive step size function
- Implicitly considers underlying system properties through $\mathrm{PDF}(\tau)$
- Outperforms non-adaptive, and previously proposed dampening schemes



Distributed Computing and Systems
Chalmers University of Technology

WASP | WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

CHALMERS

Karl Bäckström

bakarl@chalmers.se