Fluid: Resource-Aware Hyperparameter Tuning Engine

Peifeng Yu†
Jiachen Liu †
Mosharaf Chowdhury
†Equal contribution

SymbioticLab
UNIVERSITY OF MICHIGAN

Hyperparameter Tuning Trial Execution

Method	Parallel Eval	Model-based Generation	Early-stopping	Async Evaluation	Exec Strategy
Grid	✓				
SMBO		✓			
Hyperband	✓		✓		
BOHB	✓	✓	✓	✓	Async
ASHA	✓		✓	✓	Async
PBT	✓	✓			
HyperSched	✓		✓	✓	✓

- Trials execution tightly coupled with algorithm
- Hard to apply to other algorithms
- Hard to improve w/o deep knowledge of the algorithm itself

Case Study:
Lack of Elasticity Reduces Utilization

Existing trials can not easily use new idle workers

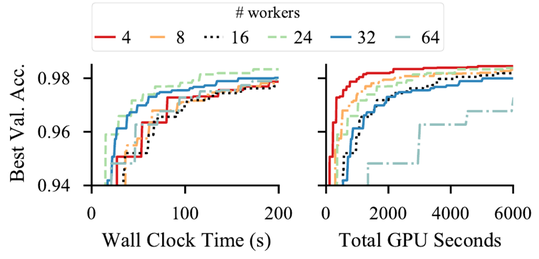
Successive Halving (SHA)
- # of workers is static
- Each trial uses one worker
- # of trials is diminishing

# Workers	Utilization	Runtime(s)
2	81.2%	2356
4	63.0%	1432
8	45.8%	1073
16	47.0%	475
32	25.2%	432

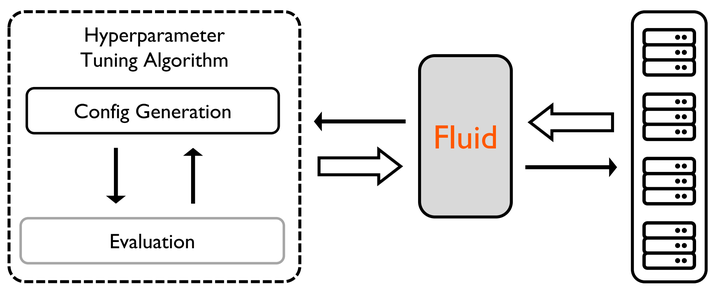
Case Study:
High Utilization != Useful Work

Increase utilization by increasing # of concurrent trials does not always work

Asynchronous SHA (ASHA)
- Trial concurrency == # of workers
- Can not scale up beyond a certain # of workers



Hyperparameter Tuning Execution Engine: Fluid

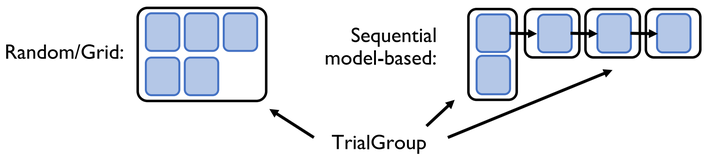


- Wide variety of tuning algorithms
  - Random/Iterative/Sequential
  - ✓ TrialGroup
- Heterogeneity & dynamicity
- ✓ Multiple source of parallelism
  - Inter-GPU: elastic dist. training
  - Intra-GPU: Nvidia MPS
- ✓ `StaticFluid/DynamicFluid`

The Interface: **TrialGroup**

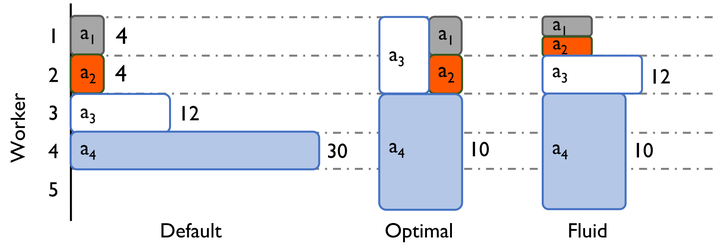
- Definition

  A group of training trials with a training budget associated to each trial.
- Example: given 5 trials to evaluate: x5



Problem Definition: Strip Packing

- Input: TrialGroup $A = \{a_1, a_2, \cdots, a_k\}$, resources $M = \{m_1, m_2, \cdots, m_n\}$
- Output: resource allocation $W = \{w_1, w_2, \cdots, w_n\}$
- Goal: minimize the length L of strips



Fully utilize the resources and mitigate the straggler

The Algorithm: `StaticFluid`

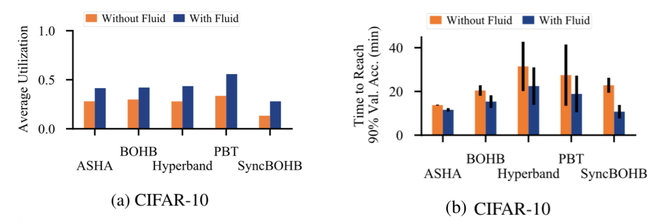
$$w_i = \min(\max(\left\lfloor \frac{h_{i,1}}{\sum_j h_{j,1}} n \right\rfloor, \frac{1}{c}), d)$$

Intra-GPU overhead
Inter-GPU overhead
Budget ratio

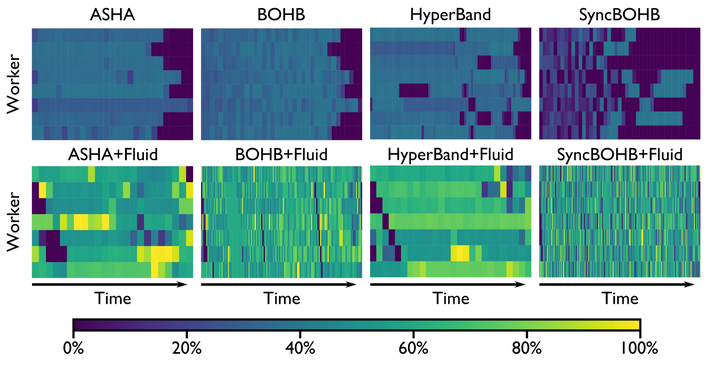
- h: trial training budget
- n: available resources
- c: maximum intra-GPU parallelism (# of packing trials)
- d: maximum inter-GPU parallelism (# of distributed workers)

Evaluation

- Average resource utilization: 10%-100% improvement
- Average job completion time: 10%-70% improvement



- Resource utilization over time



Check out our GitHub repo:
https://github.com/SymbioticLab/fluid