

Bilas Talukdar¹, Prasanna Balaprakash², Xingfu wu²,

¹ New Jersey Institute of Technology

² Argonne National Laboratory

Introduction

Achieving optimal performance has been one of the major concerns with the increasing number of tunable parameters in a HPC system. The **ytopt** is an autotuning tool developed in ECP PROTEAS-TUNE project that optimizes the search over an autotuning search space. But ytopt encounters some deployment and performance portability issues while working with large scale HPC systems. To address these we explore the use of one of the ECP workflow managers, namely libEnsemble. Ytopt with libEnsemble:

- scales the auto-tuning capability.
- enhance parallel evaluation ability of existing work.

We applied the approach to two ECP proxy applications: XSBench and sw4lite with different tuning parameters. We investigate the effectiveness of these tuning parameters at scale with respect to performance portability on ALCF Theta.

XSBench with ytopt + libEnsemble

Tunable Parameters for XSBench

number of threads
block size for dynamic schedule
omp parallel

omp placement
OMP_PROC_BIND

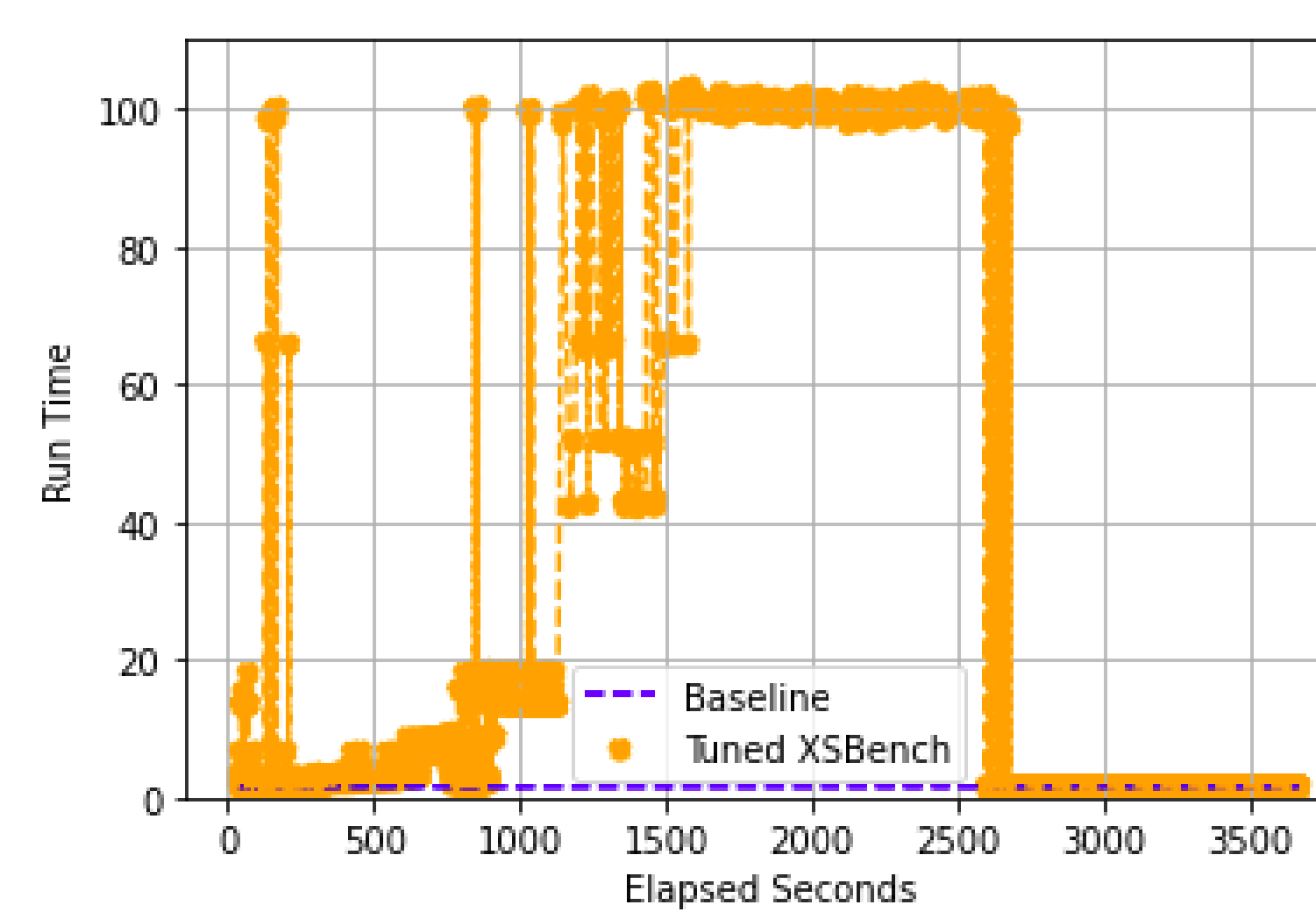


Fig. 2: Tuned XSBench (nodes=128, ranks=8, workers=16)

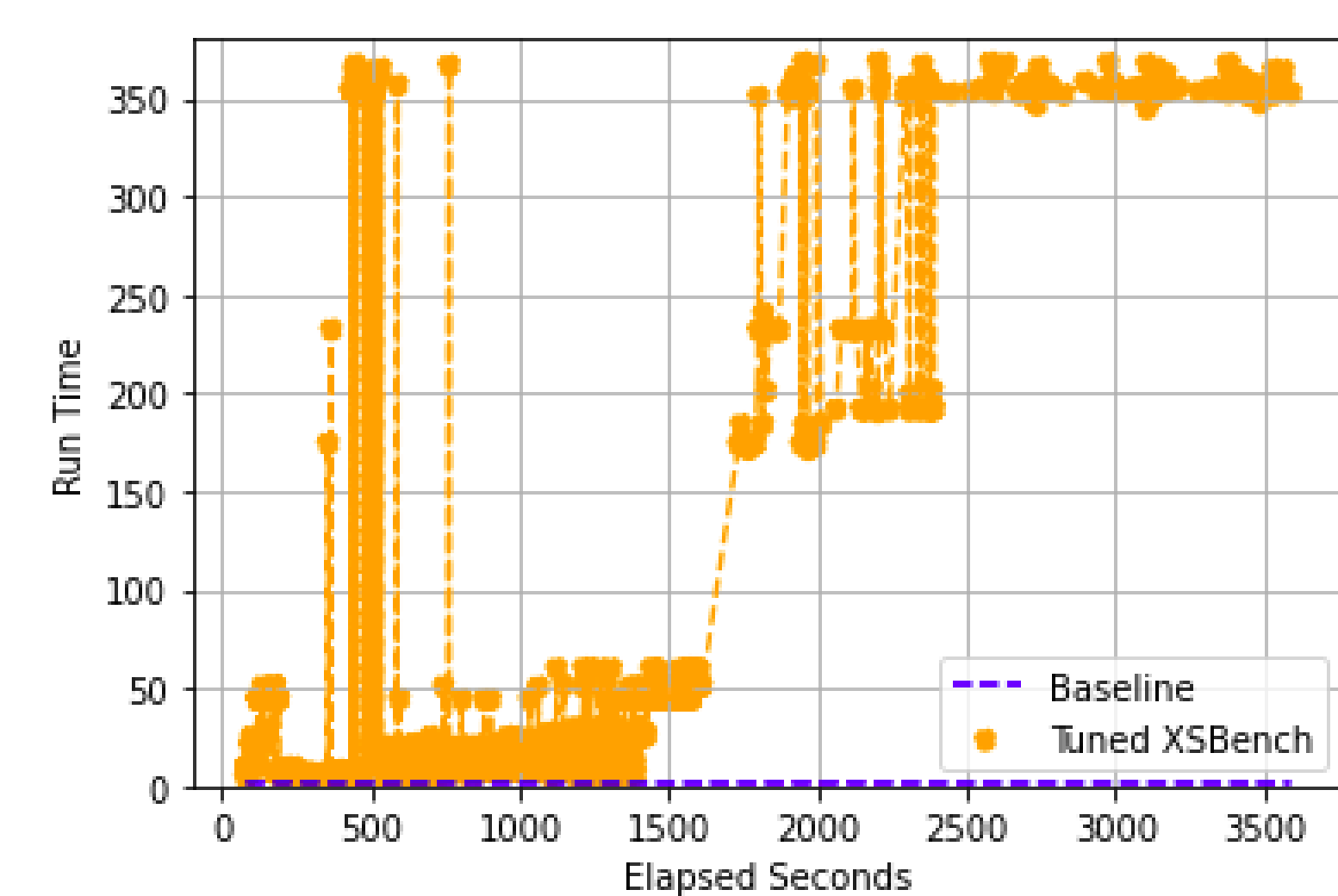
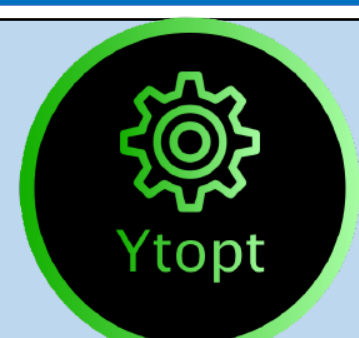


Fig. 3: Tuned XSBench (nodes=128, ranks=4, workers=32)

Overview of ytopt and libEnsemble



ytopt

- ❖ An ML based autotuning tool in ECP PROTEAS-TUNE project.
- ❖ Leverages Bayesian optimization to explore a user defined parameter space.
- ❖ Uses different supervised ML methods within Bayesian optimization such as random forests, Gaussian process regression.



libEnsemble

- ❖ A Python toolkit for coordinating workflows of asynchronous and dynamic ensembles of calculations.
- ❖ Evaluates multiple parameter configurations simultaneously.
- ❖ Helps users take advantage of massively parallel resources.

sw4lite with ytopt + libEnsemble

Tunable Parameters for sw4lite

number of threads
omp placement
OMP_PROC_BIND
OMP_SCHEDULE

omp parallel for
unroll
omp for nowait
MPI_Barrier

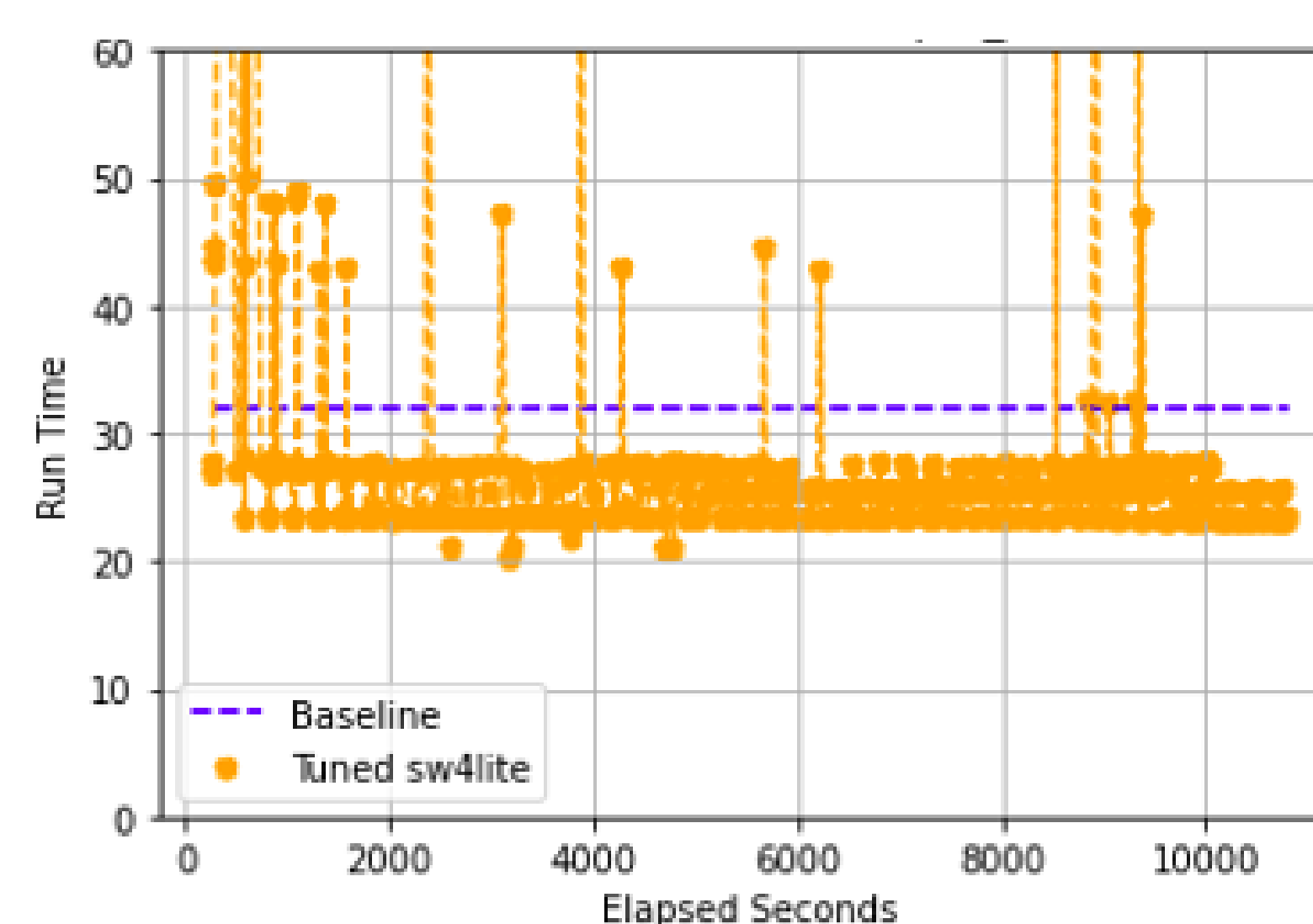


Fig. 4: Tuned sw4lite (nodes=128, ranks=8, workers=16)

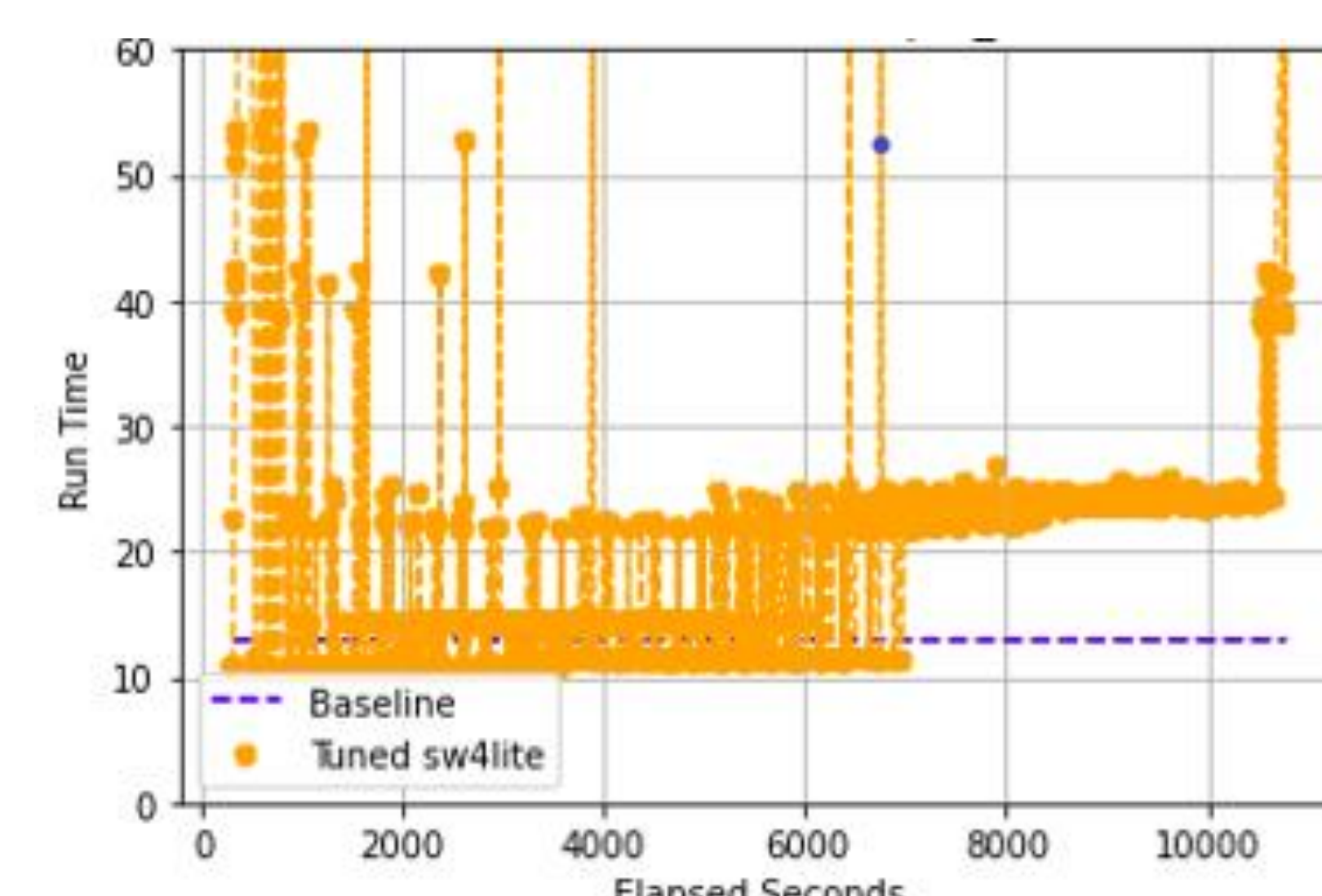


Fig. 5: Tuned sw4lite (nodes=128, ranks=4, workers=32)

ytopt + libEnsemble

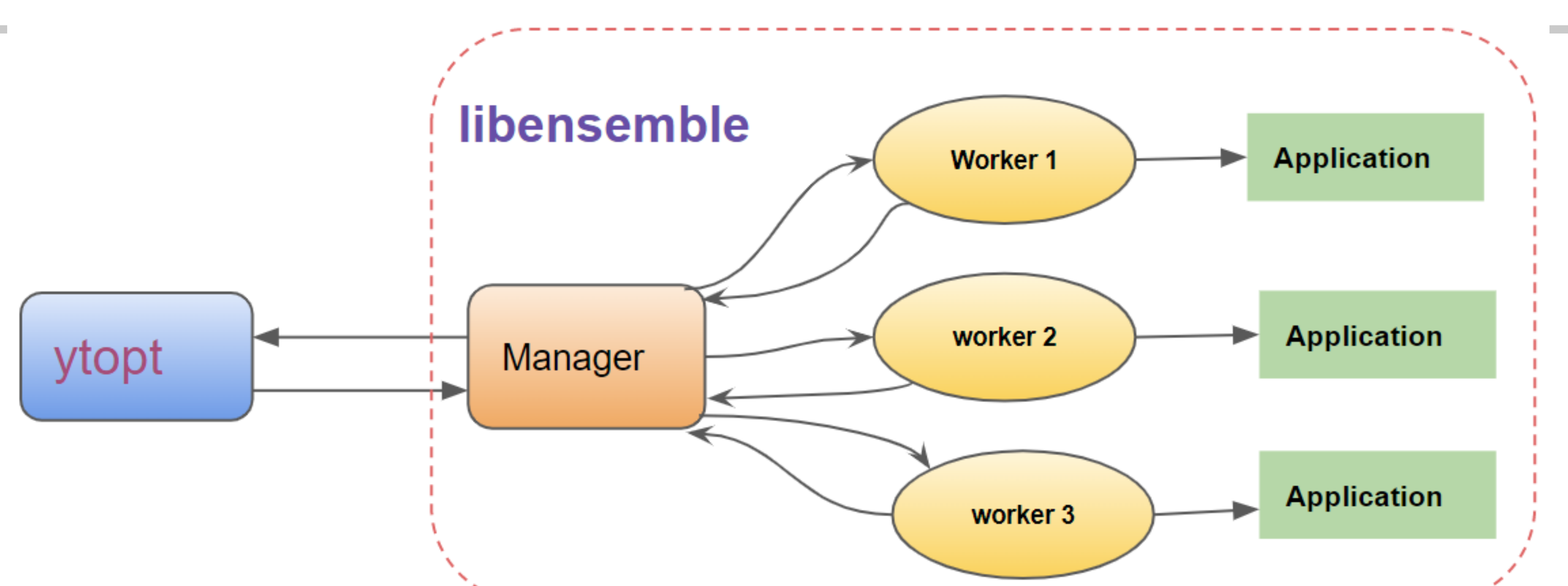


Fig. 1: Integration of libEnsemble with ytopt.

- Enhances the parallel evaluation capabilities of the ytopt.
- Multiple workers can do computations in parallel.
- Scales the whole auto-tuning process.

Related Links

Install ytopt

<https://github.com/ytopt-team/ytopt>

Find Documentation and tutorials on:

<https://ytopt.readthedocs.io/en/latest/index.html>

libEnsemble

<https://libensemble.readthedocs.io/en/main/>

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, in support of the nation's exascale computing imperative.