Multiobjective Optimization of the Variability of the High-Performance LINPACK Solver

Tyler Chang[a,b], Jeffrey Larson[b], and Layne Watson[a]

[a]Dept. of Computer Science, Virginia Tech

[b]Mathematics and Computer Science Division, Argonne National Laboratory

December 18, 2020

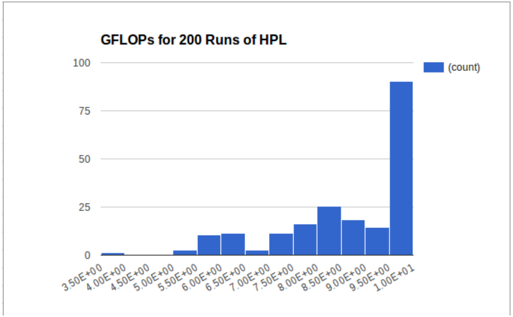
Outline

Outline

Performance Variability

Observation:
We run the same (deterministic) program multiple times on a HPC system with the same settings and inputs, but the performance values fluctuate...

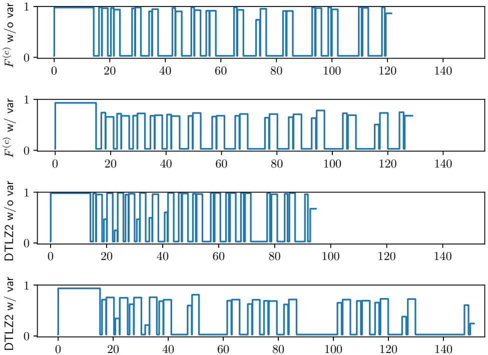
This is what we mean by **performance variability**



Why care about variability in simulations? A motivating example

A toy parallel simulation optimization problem:

- Batches of sims requested by optimization algorithm – evaluated in parallel
- Iteration tasks act as synchronization barriers
- Asynchronously distribute each batch of sims across 36 cores
- Total CPU utilization (right) for 2 toy problems, with and w/o variability



Chang, Larson, Watson, and Lux. “Managing computationally expensive blackbox multiobjective optimization problems with libEnsemble,” in Proc. 2020 Spring Simulation Conference, article no. 31.

Statement of Problem

In this project, we are looking to **control throughput variability**

- Throughput variability must be controlled synergistically with expected throughput
- We treat this as a *multiobjective optimization problem*:
  - **maximize mean throughput**
  - **minimize throughput std deviation**
- As an example task, we use the High-Performance LINPACK Benchmark (HPLB) problem — solve a dense linear system using massively parallel resources
  - A sparse solver might be more representative of a sim workload; HPLB is a standard benchmark problem and often tuned
  - The techniques used on HPLB are also relevant to most sparse solvers

Outline

High-Performance LINPACK

- HPLB throughputs are used to rank HPC systems on the HPC Top 500 list
- The standard solver for the HPLB is called `HPL`
- `HPL` has numerous algorithmic parameters that can be tuned
- Tuning `HPL` to achieve maximum throughput is encouraged before submitting to the Top 500
- **We perform multiple runs of `HPL` at each configuration and consider the mean and standard deviation of the observed throughputs**

Tuning HPL

We chose **6 integer-valued parameters** most relevant for tuning `HPL`

All other parameters are fixed to recommended values

There are over 10^{11} possible combinations of these variables

Parameter	**Lower Bound**	**Upper Bound**
NB	1	256
P	1	36
NBMIN	1	256
NDIVS	2	36
DEPTH	0	4
SN	1	256

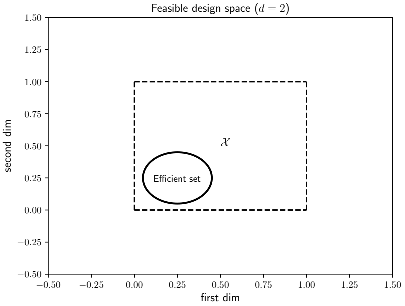
Note: P is the lead dimension of the $P \times Q$ processor grid — Q is inferred based on P and the total number of procs

Multiobjective Optimization Problems

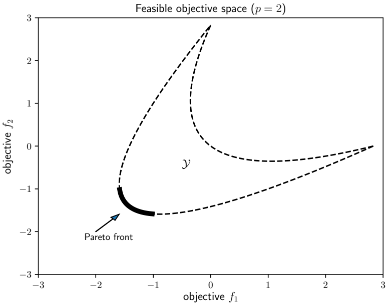
The multiobjective optimization problem (MOP):

$$\min_{x \in \mathcal{X}} F(x) \qquad \mathcal{X} \subset \mathbb{R}, \quad F : \mathbb{R}^d \to \mathbb{R}^p$$

Pareto front balances tradeoff between p conflicting objectives:



$\rightarrow$



VTMOP

- `VTMOP` is a solver for computationally expensive blackbox MOPs developed at Virginia Tech in collaboration with Argonne
- Assumes $\mathcal{X}$ is a simply bounded set and F is a computationally expensive blackbox function
- Requires slight “hacking” to work with integer variables for tuning `HPL`
  - Uses DIRECT and MADS to solve scalarized surrogate problems
  - Must configure DIRECT and MADS to only sample on integer lattice
  - Can be done by adjusting tolerances and “binning”
  - Further details in paper

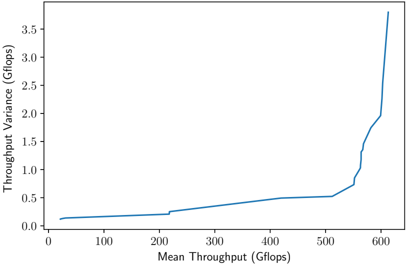
Outline

The Single-Node Experiment

- The single-node optimization of `HPL` takes place on an Intel Broadwell node of the HPC system `Bebop` at Argonne
- Each Broadwell node is a 36-core Intel Xeon E5-2695v4 processor with 128 GB of DDR4 RAM
- The problem size is a $10,000$ variable linear system
- To calculate mean and variance, 40 runs are done at each configuration
- In order to avoid wasted computation, if the estimated mean and variance after 5 runs are worse than some threshold (see paper) the run is aborted and the low-fidelity approximation is returned
- `VTMOP` is given a budget of 2000 evaluations
- The recommended configuration is provided for sanity check and to warm-start optimization

Results

24 approximately Pareto optimal solutions found:



Recommended setting is Pareto optimal, produces the max mean throughput: $\mathbb{E}[T(x)] = 613.04$, $\sqrt{Var(T(X))} = 3.8$

$\mathbb{E}[T(x)]$	$\sqrt{Var(T(x))}$	*NB*	*P*	*NBMIN*	*NDIVS*	*DEPTH*	*SN*
21.31	0.120	2	19	48	27	3	122
21.35	0.121	2	19	48	27	3	123
25.89	0.131	3	25	47	28	2	122
30.48	0.139	3	20	47	28	3	122
217.82	0.208	129	1	129	19	0	129
218.15	0.252	129	1	256	2	0	1
400.37	0.471	128	1	16	10	0	128
419.73	0.494	129	1	1	2	0	1
511.87	0.523	214	4	15	33	0	72
551.07	0.734	204	4	3	33	0	62
552.12	0.852	204	4	25	35	0	62
560.54	0.991	204	4	15	23	0	185
562.78	1.030	204	4	6	22	0	185
562.84	1.053	204	4	6	33	0	66
562.95	1.080	204	4	6	23	0	182
564.01	1.177	204	4	6	22	0	195
564.06	1.314	204	4	6	22	0	191
567.05	1.355	204	4	9	22	0	191
568.34	1.461	133	3	9	3	2	128
581.69	1.746	128	3	9	3	2	123
599.21	1.961	128	3	4	3	1	128
601.69	2.264	128	3	9	9	1	123
602.93	2.539	128	3	9	6	1	124
613.04	3.800	128	6	8	2	1	128

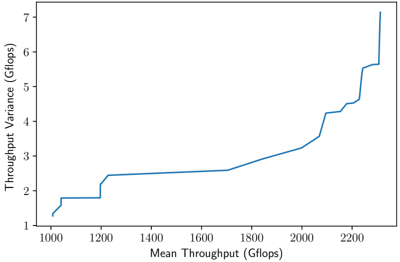
Outline

The Multi-Node Experiment

- Now using 4 nodes of `Bebop`
- Problem size increased to $20,000$ variables
- Only $s = 30$ runs of `HPL` used for computing mean and variance
- Budget decreased to just 1000 evaluations
- Because the budget was decreased, we eliminate the least important variable: fix $NB = 128$
- The recommended configuration is given again

Results

24 approximately Pareto optimal solutions found:



$\mathbb{E}[T(x)]$	$\sqrt{Var(T(x))}$	P	$NBMIN$	$NDIVS$	$DEPTH$	SN
1007.1933	1.2741281	3	123	26	3	123
1007.4800	1.3432847	3	123	26	3	118
1040.2800	1.5831875	3	117	31	3	123
1040.3267	1.7903830	3	117	31	3	118
1196.7100	1.7968075	3	117	21	2	123
1196.7167	2.0589725	3	117	21	2	121
1197.0267	2.1971664	3	117	23	2	123
1199.9100	2.2000549	3	116	21	2	123
1227.7600	2.4454885	3	112	31	2	123
1704.5900	2.5906130	8	129	20	0	138
1840.7733	2.9114439	6	117	26	3	121
1998.3433	3.2330922	7	117	26	3	123
2069.1700	3.5682653	9	124	21	3	134
2095.6033	4.2372310	9	114	21	3	134
2153.0533	4.2825011	6	123	21	1	127
2178.2900	4.5079508	6	117	21	1	121
2205.2133	4.5274438	6	112	15	1	121
2228.8300	4.6360767	9	112	21	2	123
2238.9800	5.3728821	9	107	15	2	123
2243.0733	5.5329068	7	114	21	1	123
2281.7467	5.6351565	9	119	23	1	129
2306.3233	5.6427973	9	114	23	1	123
2309.9733	6.6279416	9	107	21	1	123
2312.3500	7.1394364	9	107	26	1	118

Note the recommended settings produce the observations $\mathbb{E}[T(x)] = 2236.558$ and $\sqrt{Var(T(x))} = 21.362$, **not Pareto optimal**

Outline

Conclusions

- ✗ I should tune `HPL` for variability
- ✗ I should do a full multiobjective optimization of my kernels/solvers to figure out the best parameters for balancing variability and mean throughput
- ✓ I should be mindful of
  (1) whether variability matters for my problem; and
  (2) how the parameters I choose effect variability in addition to mean throughput

Acknowledgements

This work was supported by the U.S. Dept. of Energy (DOE) through the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two DOE organizations, the Office of Science and the National Nuclear Security Administration.

This work was also supported by the Office of Science Graduate Student Research (SCGSR) program. The SCGSR program is administered by the Oak Ridge Institute for Science and Education (ORISE), which is managed by ORAU under contract number DE-SC0014664. All opinions in this paper are the authors' and do not necessarily reflect the policies and views of the DOE, ORAU, or ORISE.

This work was also supported by the National Science Foundation under Grant No. CNS-1838271 (the VarSys project: https://varsys.cs.vt.edu/).

The authors gratefully acknowledge the computing resources provided on Bebop, an HPC system operated by the Laboratory Computing Resource Center at Argonne National Laboratory.