Beyond Bigger, Faster, Cheaper Machines: Intelligent Simulation

Edward O. Pyzer-Knapp, Kirk E. Jordan, Christopher N. Porter, David W. Turek.

In the complex world of HPC-driven innovation and discovery, what really matters to an organization is the time to actionable insight. But in today's AI-augmented world, traditional approaches to HPC design and simulation no longer provide insights at the speed organizations require.

Typically, getting actionable insight takes a significant number of simulations, and the demand for additional fidelity and increased design freedom has forced the number of simulations to grow exponentially with each product generation. To deliver insights faster, the focus has been on compressing the time to run a single simulation – through hardware advances and software optimizations.

New approaches for accelerating simulation workflows focus on extracting more information from fewer simulations. AI-augmented HPC, or intelligent simulation, applies state of the art AI optimization algorithms to minimize the number of simulations and compress design cycles because the fastest simulation is the one you don't have to run.

Bayesian optimization applied to design workflows is one type of intelligent simulation being explored and results are exciting. Applying this new method to a design flow has been proven to reduce design time by an order of magnitude, and this method can be applied to a myriad of situations and use cases. For example, for a single step within silicon chip design process, Bayesian optimization has been shown to reduce simulations 79%, from 135 to 28, and reach as good or better results.

Equally impressive results have been seen in drug discovery where this state-of-the-art method located maximally potent drug candidates 40x faster than traditional search techniques.

This paper discusses how advances with intelligent simulation and Bayesian optimization are creating a paradigm shift in HPC design and simulation to reduce time to discovery and insight while delivering higher levels of quality.

Bigger, Faster, Cheaper Machines are No Longer Enough

Since its inception, High Performance Computing has sought to revolutionize the way the real world and its digital counterpart interact. The development of new computational hardware has been intertwined with its application to new problems, and the evolution has been rapid. The IBM SP (Scalable Parallel) was born from a need for highly parallel systems, evidenced by the success of Deep Blue, and the Blue Gene series tackled problems on a new scale, including the Human Genome Project and protein folding. Recent advances in heterogeneous computing through the use of accelerators has taken the Power series to new heights, tackling a new generation of problems centered around AI and the deep learning revolution. All performed to the constant mantra – bigger, faster, cheaper.

What could you do if you cut your simulation time in half?

This development has always been a struggle against a particularly powerful adversary – the laws of physics. The first law of thermodynamics states that heat is work, and work is heat, and thus the more work you do, the more heat you produce. This essentially drove the multi-core revolution, which couples more low-powered cores to do the equivalent work of a larger, more powerful core. As transistors shrunk, though, we came up against a second law of physics which may prove more fundamental adversary – quantum mechanics. Indeed, with the current technology, there exists a critical size below which the electrons which transfer information around the computer stop behaving in a simplistic Newtonian sense and start to take on a quantum character. This may be the final nail in the coffin of Moore's law.

Despite this, the problems which we need to solve continue to gain complexity, and the timescales in which we need to solve them continue to shrink. Clearly an alternative approach is needed. If we can no longer brute force our way through problems, we must instead be more intelligent in the way in which we attack them. Bigger, faster, cheaper (machines) must metamorphose into a new mantra – bigger (problems), faster (time to insight), smarter (execution).

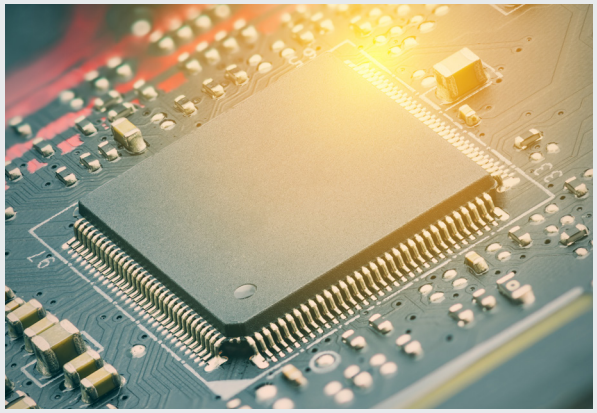
The fastest simulation is the one you don't have to run.

The Risks of Relying on Practitioner Expertise

In today's computational discovery pipeline, experienced practitioners draw on their experience in order to design a set of experiments in to pursue an objective (better understanding, or perhaps to find an optimal solution to a design problem). While the plethora of discoveries powered by computational experimentation shows that this methodology can achieve results, there are two major risks; one business and one scientific.

The primary business risk to this approach is that it places significant reliance on the expertise and relevance of the practitioner. This experience is built up over many years and is tied to a certain 'way of doing things'. If your business loses a seasoned practitioner, be it to retirement or to a rival firm, the choice is stark – go through the expense and difficulty of replacing like for like, or the opportunity cost of training up a replacement. Additionally, skill sets may be tied to a particular architecture or operating system, and so may preclude a business from making a strictly value-for-money decision when choosing to acquire new hardware.

From the scientific side the risk is a little more subtle. When the practitioner is designing their set of experiments, it is highly likely that they will, either consciously or unconsciously, favor design spaces which they know and understand. This can lead to a bias in the underlying construction of the experiments, which could lead to the user completely missing a potentially powerful solution.



CASE STUDY:

Accelerating Chip Design

The IBM chip design team working on the Power10 processor has compared BOA to their current methods. The group focused on communication signal integrity (core to core, core to GPU, etc) uses an in-house IBM simulator (HSSCDR similar to Cadence Sigrity SystemSI, Synopsis hSPICE or Mentor Graphics Hyperlynx). The design space is large, but not prohibitively large that a brute force method could be used to find an optimal design point for each communication link. However economic realities force the team to look for ever increasing efficiencies in their process.

How does BOA address these challenges? BOA was able to reduce the number of simulations required in the example problem from 135 simulations down to 28 (a 79% savings) and compared to an exhaustive search, it reduces the number of required simulations by over 98%.

What value does this bring? The primary value this brings is time to market. The chip design process has several steps within it, many depending on those upstream from it. Any reduction in time at one step gives the next step to increase the quality, the performance, or the time to market which often gives a company an edge in capturing market share.

Further, if the simulator being used was one of the commercial packages mentioned above, then there would be significant cost savings in terms of the number of licenses required to achieve a completed design within a timeframe.

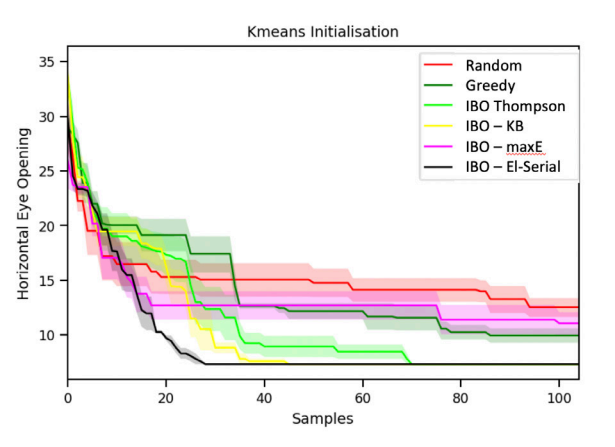
Limitations of Search in HPC

One of the most common uses for HPC is for searching a parameter space based on some kind of response. This process has many names:

- Design of experiments
- Parameter sweep
- Response characterization

In its most rudimentary form, this process includes a person or group of people who run an analysis, check results, calculate a response value (sometimes referred to as an “objective function”), modify their parameters, and run the analysis again.

With the advent of distributed memory systems, it became possible to run many independent analyses in parallel, so other methods became practical to minimize the exploration time and identify a set of near optimal parameters.



Yet there are limitations with these methods. Methods like:

Grid search

Divides the domain into evenly spaced sampling points within and along the edges of the domain. At each one of these points, the simulator is executed, and the objective function is calculated.

Limitations

1. There is no guarantee that the minimum or maximum value will fall on one of the grid points. As such, a true optimum will be missed.
2. The search is incomplete until all the grid points are simulated – potentially wasting a lot of time and resources (compute cycles, software license time, user time).

Random search

Instead of evenly covering a design space (grid search), random search uses a random number generator to select points within the design space to be simulated.

Limitations

1. There is no guarantee that the minimum or maximum value will fall on one of the grid points. As such, a true optimum will be missed.
2. The search is incomplete until all the grid points are simulated – potentially wasting a lot of time and resources (compute cycles, software license time, user time.

Gradient descent search

A slightly more sophisticated method which numerically calculates the gradient of the response in n dimensions, and then calculates the next "best" point in the direction of steepest gradient as the next simulation.

Limitations

1. The method is vulnerable to local minima or maxima where the gradients go do zero in a local region within the parameter space.
2. Further this method can get very expensive (meaning it takes more time and resource) because at each point, the gradient calculation cost is 2N where N is the number of design parameters.

There are other methods commonly used today as well, but fundamentally these methods can be distilled into combinations of the above methods (e.g. genetic algorithm search is simply a grid search combined with a random search).

Now that AI and machine learning techniques are coming of age, we can apply that technology married to Bayesian statistics to create a method to rival all of these methods and address their limitations.

The clear value of intelligent simulation is being able to extract more information from a smaller amount of simulation.

Bayesian Optimization Brings Intelligence to Simulation

In order to close the efficiency gaps with traditional search methodology, optimization algorithms are bringing intelligence to the design and deployment of computational experiments. Bayes equation, and the statistics behind that equation, provides guidance on the

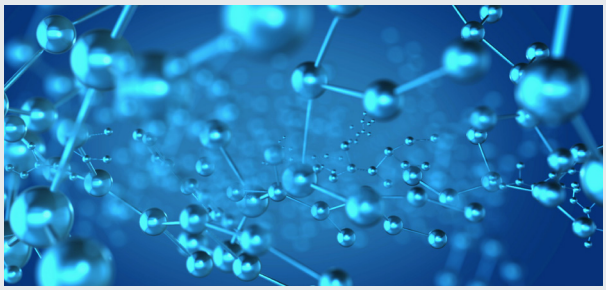
most probable parameter set to advance the exploration of the response function. Bayesian optimization answers the question "based on the limited information I have, what is the best thing for me to do next?"

Key to the Bayesian methodology is an understanding that it is as important to understand the things which you don't know as the things that you do. By balancing the twin pressures of exploration (the acquisition of new knowledge) and exploitation of the knowledge which has already been acquired, Bayesian optimization simultaneously minimizes response uncertainty while using knowledge of the response to select the best parameters to simulate next.

Brute force approaches, such as random and grid search, to simulation ensembles[1] should be seen as the HPC equivalent of the dot matrix printer. It is true that you will eventually get an answer, but the resolution is unlikely to be great and you won't know what it is until everything has completed. Using Bayesian optimization is more like focussing a blurry SLR camera – even early on you can make out the broad features and as you adjust, a clear picture rapidly reveals itself.

The potential uses for Bayesian optimization are widespread – any time you can describe your problem as trying to optimize an outcome over a set of parameters (i.e. a design space), you can use Bayesian optimization to work smarter and achieve more.

Computational experiments are typically optimized for two outcomes:
1) Quantity - How many experiments can you perform in a given time or for a given cost (more being better)
2) Quality – How sophisticated is the experiment to be performed in a given time or for a given cost.



CASE STUDY:

Accelerating and Enriching Simulation Ensembles in Computational Chemistry

In the shipping industry, optimization is a task that is undertaken on many levels, from routing, to fleet improvements, to automation. One such case, an IBM partner was designing an industrial lubricant for the crank case of a diesel truck. This new lubricant had three chemical components, but in order to be effective those components needed to stay in solution, in a single phase. The partner was using a common simulator, NAMD, to simulate the molecular dynamics of the chemicals under temperature and pressure, and planned to search the design space where the percentage of each chemical component was varied.

The goal of this search was to:

1. define the phase boundary where the components fell out of solution
2. find the optimal mix of the three components to minimize wear

How does BOA address these challenges? BOA exploited its understanding of the information content of each simulation to reduce the number of simulations by 60%, and because the information in these simulations was maximally leveraged, BOA was able select and execute the most crucial simulations (with uncertainty quantification) within the simulation space. Concentration of the simulations around the parameter values of maximum value, while discarding simulations which added no insight gave BOA the ability to define the phase boundary at a resolution increase of four orders of magnitude than previous methods.

What value does this bring? The IBM partner was able to choose a better mix of the chemicals in their lubricant product, while at the same time performing 60% fewer simulations compared to previously used brute force methods which translated to reduced cost of design and faster time to physical test and verification of the simulated result.

But Bayesian optimization creates a different paradigm for setting objectives and measuring success.

1) Savings in resource – what can you do with your budget (time, size of system) which you could not do before?

2) Savings in opportunity cost – what can you do with the time saved both in setup and execution of your problem?

3) Increase in innovation and insight – unburdened by subconscious biases, BOA is able to steer research in potentially new and exciting directions.

Introducing IBM Bayesian Optimization Accelerator

IBM Research has built a powerful software stack based upon the principles of Bayesian statistics, named IBM Bayesian Optimization Accelerator (BOA), to help accelerate simulation workflows by applying sophisticated algorithms to real world problems with thousands of design variables.

KEY CAPABILITIES

High Dimensional Capability

A common complaint of optimization algorithms is that they do not perform well when there are a large number of dimensions to search. BOA deals with this problem in two ways. For problems which it detects to fulfill certain mathematical criteria, BOA will use a sophisticated compression algorithm to allow it to optimize in its own lower dimensional space. When this is inappropriate, we have developed specialized algorithms which show strong performance in high-dimensional problems compared to current state of the art.

Parallel Optimization

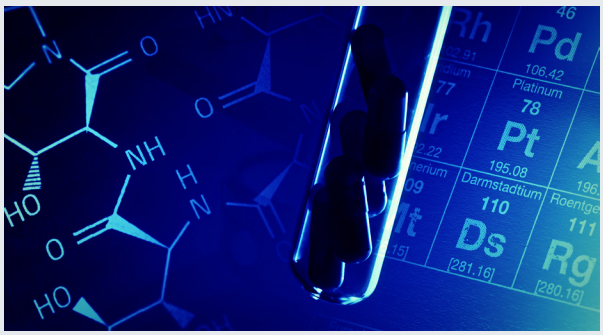
Many real-world tasks acquire data in parallel, including simulation ensembles and automated chemical screening. However, many optimization methods cannot work well when acquiring data in parallel. BOA includes new parallel algorithms developed for real world problems, which show strong performance over a range of tasks and display a significantly enhanced robustness over many repetitions over current state of the art.

Optimization for the non-expert user

Throughout BOA, we have designed the system to be as simple as possible to use, regardless of the user's level of experience. Setup can be performed programmatically through a simple JSON-like configuration, or alternatively through a graphical web-based UI. All that is required is to be able to link the parameters suggested by BOA (i.e. what should the experiment be) to a result (what happened). All the communication between your experiment and BOA is encapsulated using a simple RESTful API or alternatively through an intuitive Python SDK. While BOA contains many sophisticated algorithms, efforts have been made to implement them in such a way as to minimize the amount of user configuration required – letting the user focus on deploying their domain expertise.

Explainable Optimization

In the real world, it is important to understand not just what to do but also the why you should do it. BOA's explainable optimization module analyzes feature importance using state-of-the-art techniques, in real time, and explains the contributions to each decision in customizable, easy-to-understand graphics. Thus, by choosing BOA for your optimization engine, you gain not only improved performance but also increased insight into your challenges.



CASE STUDY:

Accelerating Pharmaceutical Drug Discovery

One key process in the drug discovery pipeline is new lead discovery (promising candidate compounds are referred to as “leads”), where leads are uncovered from a pool of potential molecules. This is akin to finding a needle in a haystack, as thousands of molecules are screened to identify tens of candidates to take to the next stage. This problem is additionally challenging as it is inherently multi-objective and subject to incredibly complex constraints. You may be able to write down the perfect molecule on paper, or test it in silicon, but it simply may not be possible to construct it in the real world. Experiments are also conducted in parallel, posing a challenge to traditional optimization techniques.

How does BOA address these challenges? IBM Research has demonstrated the ability of BOA to provide a 30-40x speedup in a challenging lead discovery problem for anti-malarial drugs. To identify the highest potency compounds in a library consisting of over 20,000 molecules, they were able to test only the most probable molecules in hundreds of experiments, instead of running tests on all 20,000.

What value does this bring? By accelerating the discovery process, BOA can enable pharmaceutical companies to bring products to market faster, with more years of patent protection. The pharmaceutical market is a highly competitive arena, where first movers in a market typically pick up a majority of the sales. Additionally, the majority of the profit made on a particular drug is achieved while it is under patent protection.

Heterogeneous, platform agnostic infrastructure

BOA provides a platform-agnostic, language-agnostic API interface. This allows a user to experience the benefits of BOA regardless of the systems architecture that the simulations (or other data acquisition activities) are built on.

Take Home Messages

As we run headlong into a data-fueled age of discovery, we should all stop and take a minute to think about how we are generating that data. Brute force (aka ‘Big Data’) approaches can only get us so far with fixed resources – instead we should think about how we can be smarter with those resources.

BOA is built to bring such an intelligence into your workflows with as little disruption as possible. While BOA is accelerated by IBM Power, it is capable of accelerating workflows on any architecture. Its powerful API, combined with an intuitive interface, means that there is a very shallow learning curve - leaving you free to focus on harnessing your creativity.

[1]Simulation ensembles are a group of related simulations designed to explore and optimize a set of parameters within a constrained design space based on an objective function (objective functions are sometimes called a “response”).

Iteration 20201117