On the Potential Use of High Throughput Computing in the Development of Material Models

Ahmed M. Hussein

An essential ingredient of modern materials science is computer simulations. Most of the studies trying to answer the open questions in this field rely to a high degree on simulating the behavior of one or more materials in a given environment under a given loading, be it mechanical, thermal, electrical, etc... Usually, the response of the material and the evolution of its characteristics are sought after. However, in order to obtain significant results, a high fidelity mathematical description of the material behavior is required as an input to the simulation process. Due to the high diversity in the behavior of different materials, the process of developing such mathematical descriptions is complicated and the resulting models are naturally prone to errors and at the same time, are subject to considerable optimizations.

1 Material Models

Material models are the mathematical descriptions of the materials' behavior. A suitable material model must be able to capture all the features relevant to the problem being studied. For example, when studying the mechanical behavior of Alumninum, one is interested to find a set of mathematical functions which may relate the load applied on an Aluminum specimen to the resulting deformation in that specimen. Of course, this is a very crude example since the deformation will naturally depend on several other factors such as the geometry of the specimen, its fixation, and the distribution of the load over its surface among other things. Abstractions are required in order to minimize the number of variables in the model. A suitable abstraction in this case would be the stress-strain relationship. This is the typical model used in the solid mechanics and materials science communities. However, when studying the electric properties of Aluminum, the stress-strain relationship mentioned above is of no good use because it only relates mechanical variables. Other constitutive models, with terms representing the electric current and the potential difference between certain points on the sample have to be developed.

Without resorting to specific details, the problem of material model development can be stated as follows.

Given a material along with some quantities of interest, it is required to develop an algorithm which when given some of these quantities as inputs, solves for the rest of them. Let the input and output quantities be $I_1, I_2, ...I_n$ and $O_1, O_2, ...O_m$ respectively, one is looking for equations of the form given below. Notice that for some cases, it is not possible to case the model in the form of mathematical equations. In such cases, an algorithm is required instead.

$$\begin{aligned} O_1 &= f_1(I_1, I_2, ..., I_n) \\ O_2 &= f_2(I_1, I_2, ..., I_n) \\ &\vdots \\ O_m &= f_m(I_1, I_2, ..., I_n) \end{aligned} \tag{1}$$

In most cases, the general behavior of the model is proposed based on experimental observations. However, in order to develop a general model, the model must contain some parameters, called material properties, which depend on the material. The material properties serve the purpose of extending the model to fit variants of the same material instead of developing the model from scratch for each material variant of interest.

The process of material models development can be thought of as the process of proposing a model, solving for the material properties, and validating the computational results from the model against the experimental ones. To do this, several sets of material properties are proposed and a model quality measure is defined. The quality measure may be the inverse of the root mean square relative error of the results when compared to the experimental measurements in a given set of experiments for example. Hence, each set of properties is used as if it is the correct set in the simulation, the quantities of interest to the quality measure are extracted from the simulation and their quality measure is evaluated. If they satisfy certain quality requirements, the set is chosen to be the target material properties, otherwise, more properties sets are tested.

While the approach seems simple, it is computationally expensive since for one properties set, several tests are required along with extensive analysis for the output data. The problem that each of these tests is a complete simulation by itself which has a time complexity in the order of the problems to be solved by the model under development.

2 High Throughput Computing

High Throughput Computing (HTC) is when several computing resources, possibly existing at different sites, are connected together to offer a huge shared computing resource. The main difference between HTC and High Performance Computing (HPC) is that HPC systems contain several nodes, up to thousands, dedicated for only one group of users and they allow parallel tasks to be executed on them. While HPC systems save a lot of time for big jobs in terms of the number of floating point operations per second compared do personal computers or small servers, the nodes in HTC systems may be small servers or desktop computers and usually they do not support parallel algorithms. On the other hand, HPC systems suffer from the queuing problem, where the submitted jobs enter a queue and they have to wait till executed, since many users have to share the same computing resources. This causes the total job completion time to increase since the time is the sum of the queue waiting time and the actual execution time. For HTC systems, the waiting time is almost eliminated since the resources are huge and easily extensible, so the job completion time is almost equal to the job execution time.

The previous discussion gives an idea about the nature of the problems suitable for HTC systems versus the HPC systems. If one makes a distinction between data centric computations, where the problem to be solved requires the execution of a rather simple algorithm but for a large number, possibly billions, of data sets, and algorithms centric computations, where the input data sets are few in number, possible one, but the processing required is complicated. An example for a data centric problem is the gene mapping where the input data set is a DNA sequence and the algorithm running is simply comparing DNA strings. On the other extreme, process centric problems are ones like molecular dynamics simulations where the position and velocity of each atom or molecule are required in every time step and calculating them requires the solution of large systems of linear equations several times for each time step. For data centric problems, HTC systems are better suited because the run time is short and the number of runs is huge so it is impractical to spend a lot of time waiting in queues. On the other end of the spectrum, HPC systems are suitable for algorithm centric problems where parallelism is necessary, the run time is large and the number of complete runs is relatively small so the queue waiting time is reasonable.

The problem in hand seems to be a mix of both. Large number of runs is required to try different material properties and at the same time, each run takes a significant amount of time because it is a series of simulations. The decision to use HTC or HPC systems depends on the complexity of the algorithm processing the data and the number of expected runs. Other less obvious factors enter the scene such as the average waiting time for the HPC system to be used and the average execution time on the HTC system.

3 Implementation Plan on HTC

A simple implementation plan for the solution on HTC systems is to use the Directed Acyclic Graph Manager (DAGMAN) to control the submission of the simulation jobs to different HPC nodes. To get the most of the HTC architecture, the following procedure is proposed.

1. Compile the code which takes a single set of material properties and output the overall quality measure

2. Compile the code which generates sets of material properties. The generation may be completely random or may be governed by certain guidelines according to the material being modeled

3. Run the properties generator locally and generate N sets of mechanical properties

4. Write a dag file $D1$ which will execute m jobs simultaneously,$m < N$, each job is simple a simulation for one set of mechanical properties. In this dag file, write a post running script which will indicate that the dag file was executed one more time. Thus, if this is the first time, it should keep the value of 1, if it is the second, it should read the current value, 1, and increment it to 2, and so on

5. Write a dag file $D2$ which is similar to $D1$, however, a slight modification would be to run a script before running any jobs to check if any of the currently obtained quality measures is satisfactory. If a satisfactory result is found, the script exits after saving a pointer to the candidate solution, if not, it proceeds to execute the same jobs again but with input sets ranging from jm to $(j+1)m$ where j is the total number of executions of the dag jobs

6. Write a final dag file $D3$ which will run $D1$ only once and $D2$ k times where $k = N/m$. If a good solution candidate is found at any time, the execution will stop before completing the k iterations.

The previous procedure is not the optimum one but it is one that does the job. A more efficient and more sophisticated procedure would be to use Genetic Algorithms were all the genetic operators (selection, cross-over, mutation, etc...) are to be applied by running scripts from the $D2$ file. One drawback of that approach is that the number of generations of the genetic algorithm is limited to the number of $D2$ calls in $D3$ while it is determined by the convergence criteria in practice.

4 Potential Application in Similar Problems

The use of HTC systems may be extended to solve a large class of similar problems using a procedure similar to the one outlined in the previous section. The problem described here can be considered an optimization problem where the optimum set of mechanical properties is required and the optimization is for the quality measure. Optimization is found at the core of all engineering design problems. One can cast any design problem in a form similar to the one described here with the trial sets being sets of design variables and the quality measure being a measure indicating the degree to which the design satisfies the requirement and respects the constraints. While this may not be the best way to solve design problems because it lacks any feedback driven guidance for the solution, it can help solving problems where the quality-input relationships are complicated and sensitive, a case which is found in most real life design cases.