Dynamic Data-Driven Systems Approach for Simulation Based Optimizations*

Tahsin Kurc[1], Xi Zhang[1], Manish Parashar[2], Hector Klie[3], Mary F. Wheeler[3], Umit Catalyurek[1], and Joel Saltz[1]

[1] Dept. of Biomedical Informatics, The Ohio State University, Ohio, USA
{kurc,xizhang,umit,jsaltz}@bmi.osu.edu
[2] TASSL, Dept. of Electrical & Computer Engineering, Rutgers,
The State University of New Jersey, New Jersey, USA
parashar@caip.rutgers.edu
[3] CSM, ICES, The University of Texas at Austin, Texas, USA
{klie,mfw}@ices.utexas.edu

Abstract. This paper reviews recent developments in our project that are focused on dynamic data-driven methods for efficient and reliable simulation based optimization, which may be suitable for a wide range of different application problems. The emphasis in this paper is on the coupling of parallel multiblock predictive models with optimization, the development of autonomic execution engines for distributing the associated computations, and deployment of systems capable of handling large datasets. The integration of these components results in a powerful framework for developing large-scale and complex decision-making systems for dynamic data-driven applications.

1 Introduction

The problem of optimal decision making and reliable parameter estimation on physical domains is generally one of the most challenging tasks in many engineering and scientific applications, such as biomedicine, structural mechanics, energy and environmental engineering. In these applications, the feasibility of the tasks depend heavily on how simulations and optimization are effectively combined [1,2,3,4,5,6,7,8,9,10]. The overall goal is both to generate good estimates of optimal parameter values and to reliably predict end results. The objective function of the optimization process can be viewed as a performance measure that depends on an array of controlled variables (e.g., the number and locations of sensors), which define the decision parameters, and a vector of uncontrolled conditions (e.g., electromagnetic properties of the subject sample in magnetic resonance imaging, subsurface properties in reservoir simulations, boundary

* The research presented in this paper is supported in part by the National Science Foundation Grants ACI-9984357, EIA-0103674, EIA-0120934, ANI-0335244, CNS- 0305495, CNS-0426354, IIS-0430826, ACI-9619020 (UC Subcontract 10152408), ANI-0330612, EIA-0121177, SBR-9873326, EIA-0121523, ACI-0203846, ACI-0130437, CCF-0342615, CNS-0406386, CNS-0426241, ACI-9982087, CNS-0305495, NPACI 10181410, Lawrence Livermore National Laboratory under Grant B517095 (UC Subcontract 10184497), Ohio Board of Regents BRTTC BRTT02-0003, and DOE DE-FG03-99ER2537.

Y. Shi et al. (Eds.): ICCS 2007, Part I, LNCS 4487, pp. 1213–1221, 2007.

conditions), which represent the characteristics of the physical domain. Finding a solution to the objective function requires a systematic search of the parameter space and evaluation of corresponding scenarios within the physical domain. An exhaustive search of the space is often unfeasible, since the space can consist of thousands to millions of data points and requires the evaluation of a very large number of potential scenarios corresponding to these points. Moreover, the values of uncontrolled variables are not known precisely, introducing a high level of uncertainty into the problem. A possible approach is to couple optimization algorithms with simulations and experimental measurements to enable a systematic evaluation of the scenarios. This approach, together with the experienced judgment of specialists, can allow for better assessment of uncertainty and can significantly reduce the risk in decision-making [11,12]. However, such an approach is characterized by dynamic interactions between complex numerical models, optimization methods, and data. Major obstacles to using this approach include the large computational times required by the complex simulations, the challenges of integrating dynamic information into the optimization process, and the requirements of managing and processing large volumes of dynamically updated datasets (obtained either from simulations or sensors).

In this paper, we provide an overview of the problems in optimization in domains with large search space and the approaches we have devised to address these issues. Our efforts have resulted in (1) algorithms to support parallel multi-block numerical models coupled with global stochastic optimization algorithms, (2) execution engines that implement adaptive runtime management strategies to enable efficient execution of optimization and simulation processes in distributed and dynamic computational environments, and (3) systems to handle very large and potentially dynamic multi-dimensional, scientific datasets on large scale storage systems. This paper is structured as follows. The next section provides further motivations through the description of different simulation based optimization scenarios. Section 3 describes support for multiblock simulations. Data management and processing solutions are presented in Section 4. Section 5 concludes the paper..

2 Optimization Challenges in Large-Scale Domains

In making decisions based on a modeling of the physical domain under study, there are at least two key goals to consider: (1) the design and deployment of man-made objects to optimize a desired response, and (2) the reproduction of the behavior of the physical phenomenon by matching the numerical model response with the field measurements. The first goal deals with forecasting the behavior of the model under a given set of conditions. The objective here is to find the optimal values of the operational parameters. The second goal implies an estimation and understanding of the parameters (state and control variables) describing the model (e.g., porosity, permeability, pressures, temperatures, geometry), and analysis and validation of the corresponding numerical model. The experimental data obtained from sensors can be compared with predictions obtained through a numerical model with the objective of reducing mismatch between the computed and observed data.

Consider optimization in oil reservoir management [1] in the context of the first goal described above. The number and locations of wells in an oil reservoir have significant impact on the productivity and environmental effects of the reservoir (i.e., optimum economic revenue, minimum bypassed hydrocarbon, minimal environmental hazards/impact). If the oil extraction wells are not placed carefully, large volumes of bypassed oil may remain in the field. The amount of water that needs to be injected (in order to drive the oil toward extraction wells) and disposed of also depends on the number and locations of injection wells as well as the extraction wells. For example, assume the objective function is formulated as maximizing profit. Here, the decision parameters are the number of injection and extraction wells and their locations. The uncontrolled conditions are the geological and fluid properties of the reservoir and the economic parameters (e.g., the cost of pumping water). Even with a fixed number of wells, finding a solution for this objective function requires the evaluation of a large number of possible configurations. For each placement of the wells, the reservoir model has to be evaluated for many time steps in order to calculate required parameters (i.e., the effective volume of extracted oil and the net value). Moreover, the lack of complete information about these properties requires the use of stochastic approaches in the generation of equally probable scenarios (or realization) using Monte Carlo simulation or in the solution of stochastic PDEs [13].

Another example is high-field Magnetic Resonance Imaging (MRI). High-field MRI devices (e.g., 7 Tesla systems) offer high signal-to-noise ratios, better contrast, greater shift dispersion, and thus the ability to obtain better, higher resolution images [14]. However, a problem in high-field MRI is the non-uniform detection of signals across the subject (sample) being imaged. That is, the brightness in the final image of the sample varies across the spatial domain; some regions are brighter (finer details can be seen), whereas other regions are darker, depending on the location and voltage strength of the coils in the device. A challenging design and operational problem, therefore, is to determine the location of the coils and the amount and duration of voltage values per coil to achieve optimal distribution of brightness across the subject volume. This involves simulating signal distribution within the volume for a given set of design/operational parameters (i.e., coil locations, coil voltage values, and the duration of voltage at each coil) [15], and searching the space of design and operational parameters for optimal values. To speed up the execution of simulations, multi-resolution Grid techniques can be used [16].

From a computational viewpoint, these applications perform function evaluations by solving sets of coupled nonlinear PDEs in three dimensions, for multiple times steps and for different sets of parameters. Furthermore, the simulation and optimization components require the synthesis and querying of data to search the parameter space. The datasets are generated using simulations or obtained from sensors. They are large, multi-dimensional, multi-scale, and may be generated and stored at disparate locations. In the following sections, we present techniques and tools to address these problems.

3 Distributed Multi-block Simulations

The ability to define different regions with differing degree of granularity provides a valuable means for focusing computational resources to critical areas, specially since

simulating high fidelity descriptions of the entire system may be computationally too expensive. Furthermore, incorporating automatic model reduction into solution procedures provides an additional means for increasing computational efficiency by lumping parameters, and simplifying basic principles. From the conceptual and computational standpoint, different models and interactions may take place in the same domain at different spatial and temporal scales. In order to deal with the accurate and efficient solution of these problems, the spatial physical domain may be decomposed (i.e., decoupled) into different blocks or subdomains under the assumption that different algorithms, physical models or scales are solved. In order to preserved the integrity of the overall solution, continuity of fluxes and pressures are imposed across each subdomain.

One way to efficiently establish this spatial coupling/decoupling among subdomains is through mortar spaces [17,18,19,20,21]. These spaces allows imposing physically driven matching conditions on block interfaces in a numerically stable and accurate way. Some of the computational advantages of the multi-block approach include: (1) multi-physics, different physical processes/mathematical models in different parts of the domain may be coupled in a single simulation; (2) multi-numerics, different numerical techniques may be employed on different subdomains; (3) multi-scale resolution and adaptivity, highly refined regions may be coupled with more coarsely discretized regions and dynamic grid adaptations may be performed locally on each block; and (4) multi-domains, highly irregular domains may be described as unions of more regular and locally discretized subdomains with the possibility of having interfaces with non-matching grids.

A key challenge in parallel/distributed multi-block formulations are the dynamic and complex communication and coordination patterns resulting from the multi-physics, multi-numerics, multi-scale and multi-domain couplings. These communication and coordination patterns depend on state of the phenomenon being modeled. Moreover, they are determined by the specific numerical formulation, domain decomposition and sub-domain refinement algorithms used, which, in most practical cases, is known only at runtime. Implementing these communication and coordination patterns using commonly used parallel programming frameworks is non-trivial.

Seine/MACE [22,23], developed as part of this effort, provides a virtual shared space abstraction to support interactions in parallel multi-block simulations. Seine builds on two key observations: (a) formulations of the targeted simulations are based on geometric multi-dimensional domains (e.g., a grid or a mesh) and (b) interactions in these applications are typically between entities that are geometrically close in this domain (e.g., neighboring cells, nodes or elements). Rather than implementing a general and global associative space, Seine defines geometry-based transient interaction spaces, which are dynamically created at runtime, and each of which is localized to a sub-region of the application geometric domain. The interaction space can then be used to share objects between nodes whose computational sub-domains geometrically intersect with that region. To share an object using the interaction space, nodes do not have to know of, or synchronize with each other at the application layer. Sharing objects in the Seine model is similar to that in a tuple space model. Furthermore, multiple shared spaces can exist simultaneously in the application domain. An experimental evaluation demonstrates its

scalability and low operational overheads, as well as its ability to effectively support distributed multiblock simulations [23].

4 Data Management and Processing

Management, querying, and processing of data generated and referenced during optimization present several challenges. In this section, we describe these challenges and the techniques and tools we have developed to address them.

A dataset generated by an optimization run consists of the values of the input and output parameters (i.e., the values of controlled variables and uncontrolled conditions, and the output of the objective function) along with the output from simulations of a numerical model of the physical domain. User-defined metadata also is needed to describe a given optimization run (e.g., the names of the optimization methods used, the id of the optimization run, the name of the simulation model used). By maintaining these data types and datasets, a large-scale knowledge base can be created and used to speed up the execution of optimization runs and to carry out post-optimization analyses. In the simplest case, the knowledge base can be queried during optimization to see if a given step, or a subset of numerical simulations at that step, have already been evaluated, potentially in a previously executed or concurrently executing optimization run. In this case, the query can be formulated to search the knowledge base based on the metadata to check if a given optimization step has already been evaluated and its output stored in the system. Similarly, during post-optimization analyses, a user may want to compare and correlate results obtained from one optimization run with results from another set of optimization runs. However, querying of metadata and information discovery in large, decentralized, and dynamic environments is a challenging problem. To address this challenge, we have developed Squid [24], which is a decentralized distributed information discovery system that supports complex content-based queries including ranges and wildcards. It guarantees that all existing data elements that match a query will be found with bounded costs in terms of the number of messages and the nodes involved. A key innovation is a locality preserving indexing scheme that effectively maps the multidimensional information space to physical nodes. This indexing scheme makes use of Space Filling Curves (SFC) [25]. Squid defines two basic operations: "publish" and "query". In the publishing process, the keywords describing the content of the data element and the SFC-mapping are used to construct the index for the data element, and this index is used to store the element at the appropriate node in the overlay. In the simple querying process, the query is translated into the corresponding region in the multi-dimensional information space and the relevant clusters of the SFC-based index space, and the appropriate nodes in the overlay are queried.

Another challenge is the large volumes of data and dynamically updated multidimensional datasets that are generated by simulations or field sensors. At each step of an optimization run, one or more simulations (using different values for the uncontrolled conditions) may need to be executed over many simulation time steps on a large mesh modeling the physical domain. Even a single optimization run may generate multiple terabytes of data. During an optimization run, simulations can be executed on distributed collections of compute systems. The environment can be heterogeneous, consisting

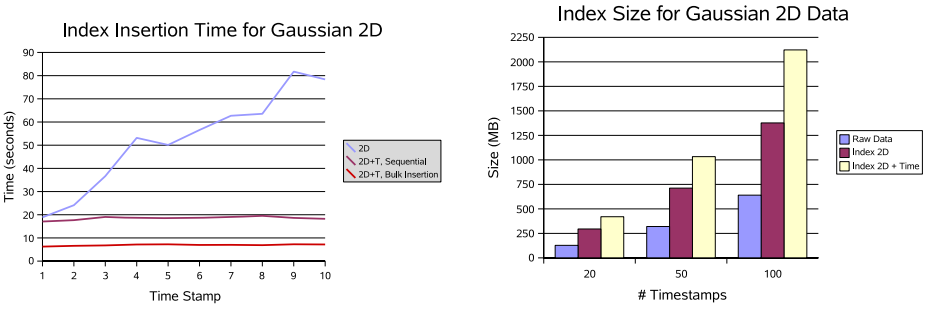

Fig. 1. (a) Index insertion time. (b) Index size.

of systems with different storage and computation capacities. Moreover, datasets can be dynamically updated, from new simulations or from sensors and monitors in the field; data from sensors can be used to validate simulation results as well as to refine numerical models. When datasets are updated and new data elements are added, indexes as well as the organization of the dataset on disk and in memory need to be updated. For example, assume the dataset is initially partitioned into a set of chunks so that every chunk contains an approximately equal number of elements and also contains elements that are close to each other in the multi-dimensional space of the domain. When new data elements are added, one approach is to create new chunks from these data elements. One problem with this approach is that it may result in a very large number of chunks, increasing the cost of indexing. Moreover, if the coordinates of new data elements are spread across the multi-dimensional space, the bounding boxes of the resulting chunks may be large. This may introduce I/O overheads; a query may intersect these chunks even though the chunks may not have any data elements required for the query. Alternately, existing data chunks can be reorganized to store the new data elements. The advantage of this approach is that it reduces the number of chunks and is also more likely to result in chunks with tighter bounding boxes. However, it will require reorganization of the dataset on disk, resulting in I/O overheads during updates. The indexes also need to be be updated when new data is added. A simple approach would be to update indexes as new data arrives. This would require frequent updates to the index. Another possible approach would be to aggregate data updates, organize them, and do a bulk update to the index. We have developed the STORM framework to address the storage and querying of such very large datasets. STORM [26,27] is a service-based middleware that is designed to provide basic database support for very large datasets where data is stored in flat files. Such datasets commonly arise in simulation studies and in biomedical imagery. STORM supports execution of SQL-style `SELECT` queries on datasets stored on distributed storage systems. These services provide support for indexing, data extraction, execution of user-defined filters, and transferring of data from distributed storage nodes to client nodes in parallel.

Within the STORM framework, we are also investigating runtime support to efficiently execute range and sampling queries against dynamic multi-dimensional datasets. Experimental results on data organization and index updates are presented in Figure 1. Figure 1(a) compares the performance of different strategies for inserting incoming data

tuples into R-tree indexes [28] (to speed up range queries) in terms of insertion time. In these experiments, we used synthetically generated datasets. Data objects are generated within a 2D normalized $[1 \times 1]$ space using a Gaussian distribution for their coordinates at each time stamp. The size of each data tuple is fixed at 64 bytes and we generated data for 500 time stamps. We fixed the number of data objects to be $100,000$. For the index, we have three options. First, we only use the data attribute as the index attribute, second we use the time stamp as an index attribute with sequential insertion, and third we use the time stamp as an index attribute with bulk insertion. The figure shows that the insertion time for the index with only the data attribute as index attribute increase almost linearly as the size of the index increases. On the other hand, the insertion time for the index with time as an index attribute is almost constant, is independent of the size of the original index, and only depends on the size of the inserted data. Furthermore, bulk insertion achieves the best insertion performance, as expected. We should note that by using time stamp as an indexing attribute, we effectively increase the attribute space and the overall storage space for the index file will likely increase. Figure 1(b) shows that the overall size for the index with time as an index attribute is larger than the index with only data attribute as index attributes. This is a space-efficiency tradeoff, and as disk storage capacity continues to grow at very fast rate and price per GB storage continues to fall, we believe that the insertion efficiency would be the deciding factor in most real-world scenarios.

5 Conclusions

Coupling optimization algorithms with simulations and experimental measurements represents an effective approach for optimal decision making and reliable parameter estimation in engineering and scientific applications, and can allow for better assessment of uncertainty and significantly reduce the risk in decision-making. However, the scale, dynamism and heterogeneity of computation and data involved in the approach present significant challenges and require effective computational and data management support. Specifically, achieving high levels of performance and accuracy on multiphysics and multiscale simulation based optimizations require the development of sophisticated tools for distributed computation and large scale data management. Further, dynamic data-driven simulations require the timely orchestration of several components that may deal with different models, to respond adequately to disparate data streams and levels of information. Uncertainty is unavoidable in data and models, and may govern the entire dynamic data-driven process if it is not assessed and managed in an adaptive manner. This paper reported on models, algorithms and middleware level solutions developed by the authors to address these challenges. These solutions have been shown to effectively enable reliable optimizations and decision-making processes for oil reservoir management, contaminant management and other related applications.

References

1. Bangerth, W., Klie, H., Parashar, M., Mantosian, V., Wheeler, M.F.: An autonomic reservoir framework for the stochastic optimization of well placement. Cluster Computing **8**(4) (2005) 255–269

2. Parashar, M., Klie, H., Catalyurek, U., Kurc, T., Bangerth, W., Matossian, V., Saltz, J., Wheeler, M.F.: Application of Grid-enabled technologies for solving optimization problems in data-driven reservoir studies. Future Generation of Computer Systems **21** (2005) 19–26
3. Eldred, M., Giunta, A., van Bloemen Waanders, B.: Multilevel parallel optimization using massively parallel structural dynamics. Structural and Multidisciplinary Optimization **27** (1-2) (2004) 97–109
4. Eldred, M., Giunta, A., van Bloemen Waanders, B., Wojtkiewicz Jr., S., Hart, W., Alleva, M.: DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Version 3.0 Users Manual. Sandia Technical Report SAND2001-3796. (2002)
5. Stori, J., King, P.W.C.: Integration of process simulation in machining parameter optimization. Journal of Manufacturing Science and Engineering (USA) **121**(1) (1999) 134–143
6. Fu, M.C.: Optimization via simulation: A review. Annals of Operations Research (Historical Archive) **53**(1) (1994) 199–247
7. Gehlsen, B., Page, B.: A framework for distributed simulation optimization. In: Proceedings of the 2001 Winter Simulation Conference. (2001) 508–514
8. Schneider, P., Huck, E., Reitz, S., Parodat, S., Schneider, A., Schwarz, P.: A modular approach for simulation-based optimization of mems. In: SPIE Proceedings Series Volume 4228: Design, Modeling, and Simulation in Microelectronics. (2000) 71–82
9. Mendes, P., Kell, D.B.: Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics **14**(10) (1998) 869–883
10. Klie, H., Bangerth, W., Gai, X., Wheeler, M., Stoffa, P., Sen, M., Parashar, M., Catalyurek, U., Saltz, J., Kurc, T.: Models, methods and middleware for Grid-enabled multiphysics oil reservoir management. Engineering with Computers **22** (2006) 349–370
11. Spall, J.C.: Introduction to stochastic search and optimization: Estimation, simulation and control. John Wiley & Sons, Inc., Publication, New Jersey (2003)
12. Yeten, B., Durlofsky, L.J., Aziz, K.: Optimization of nonconventional well type, location, and trajectory. SPE Journal **8**(3) (2003) 200–210 SPE 86880.
13. Zhang, D.: Stochastic Methods for Flow in Porous Media. Academic Press, Inc (2002)
14. Hoult, D., Richard, R.: The signal to noise ratio of the nuclear magnetic resonance experiment. Journal of Magnetic Resonance **24** (1976) 71–85
15. Han, Y., Wright, S.: Analysis of rf penetration effects in mri using finite-difference time-domain method. Proc SMRM 12th Annu. Meeting, New York (1993)
16. Zivanovic, S.S., Yee, K.S., Mei, K.K.: A subgridding method for the time-domain finite-difference method to solve maxwell's equations. IEEE Trans. Microwave Theory Tech. **39** (1991) 471–479
17. Arbogast, T., Cowsar, L.C., Wheeler, M.F., Yotov, I.: Mixed finite element methods on non-matching multiblock grids. SIAM J. Numer. Anal. **37** (2000) 1295–1315
18. Li, J., Wheeler, M.F.: Uniform convergence and superconvergence of mixed finite element methods on anisotropically refined grids. SIAM Journal on Numerical Analysis **38**(3) (2000) 770–798
19. Peszyńska, M., Wheeler, M.F., Yotov, I.: Mortar upscaling for multiphase flow in porous media. Comput. Geosci. **6**(1) (2002) 73–100
20. Wheeler, M.F., Peszynska, M.: Computational engineering and science methodologies for modeling and simulation of subsurface applications. (Advances in Water Resources) in print.
21. Wheeler, M.F., Yotov, I.: Physical and computational domain decompositions for modeling subsurface flows. In Mandel, J., et al., eds.: Tenth International Conference on Domain Decomposition Methods, Contemporary Mathematics, vol 218, American Mathematical Society (1998) 217–228

22. Zhang, L., Parashar, M.: A dynamic geometry-based shared space interaction framework for parallel scientific applications. In: Proceedings of the 11th Annual International Conference on High Performance Computing (HiPC 2004). Volume 3296., Bangalore, India, LNCS, Springer-Verlag (2004) 189–199
23. Zhang, L., Parashar, M.: Seine: A dynamic geometry-based shared space interaction framework for parallel scientific applications. Concurrency and Computations: Practice and Experience **18**(15) (2006) 1951–1973
24. Schmidt, C., Parashar, M.: Enabling flexible queries with guarantees in p2p systems. IEEE Network Computing, Special Issue on Information Dissemination on the Web **8**(3) (2004) 19–26
25. Sagan, H.: Space-Filling Curves. Springer-Verlag (1994)
26. Narayanan, S., Kurc, T., Catalyurek, U., Saltz, J.: Database support for data-driven scientific applications in the grid. Parallel Processing Letters **13**(2) (2003) 245–271
27. Narayanan, S., Kurc, T.M., Catalyurek, U.V., Saltz, J.H.: Servicing seismic and oil reservoir simulation data through grid data services. In: Proceedings of VLDB Workshop Data Management in Grid 2005 (VLDB DMG'05). (2005) 98–109
28. Guttman, A.: R-Trees: A dynamic index structure for spatial searching. In: Proceedings of the 1984 ACM-SIGMOD Conference. (1984) 47–57