CAD Grid: Corporate-Wide Resource Sharing for Parameter Studies

Ed Wheelhouse[1], Carsten Trinitis[2], and Martin Schulz[3]

[1] Department of Computer Science, University of Newcastle, Newcastle, UK
edwheelhouse@yahoo.co.uk
[2] Technische Universität München (TUM), Institut für Informatik
Lehrstuhl für Rechnertechnik und Rechnerorganisation,
Boltzmannstr. 3, 85748 Garching bei München, Germany
Carsten.Trinitis@cs.tum.edu
[3] School of Electrical and Computer Engineering, Cornell University,
Ithaca, NY, 14853, USA
schulz@csl.cornell.edu

Abstract. The optimization process for most modern engineering problems involves a repeated modeling of the target system, simulating its properties, and refining the model based on the results. This process is both time and resource consuming and therefore needs to rely on a distributed resource sharing framework in order to optimally exploit the existing resources and minimize the response time for the design engineers.
We have implemented such a framework for the design process of high voltage components and have shown its applicability to a real industrial environment. First results are very encouraging and show a high acceptance rate with the end-users. In addition, experiments with various different models show the profound impact of the optimization on the design of high-voltage components.

1 Motivation

The geometric shape of transformers and other high voltage gear has a profound impact on their electrical properties and on their performance. Suboptimal designs can lead to overheating, flashovers between close parts, higher energy loss rates, etc. . It is therefore necessary to optimize the geometric shape already in the initial design phase using a detailed simulation of the electric properties of the designs. This enables the engineers to detect critical regions and to change the respective CAD geometry in order to obtain an optimal design. This can be automated using an iterative optimization approach. During each iteration, a new CAD model is generated and used as the input for the following simulation of the electric field. The result of the simulation is then used to refine the model. This process continues until an optimal design has been found.

Each iteration requires repeated access to the CAD modeling tool and to the compute server for the simulation. The requests for these resources are interleaved and the usage of each resource potentially requires a significant amount

H. Kosch, L. Böszörményi, H. Hellwagner (Eds.): Euro-Par 2003, LNCS 2790, pp. 433–440, 2003.

of time since both model generation and simulation are non-trivial operations. In addition, the resources for each step are strictly limited by the number of available parallel systems for the simulation and by appropriate licenses for the CAD modeling. Therefore, a single optimization request should not block a compute or a CAD modeling resource for its entire runtime and neither should each engineer require his or her own, dedicated set of resources. Instead, a corporate-wide resource sharing infrastructure is needed, which dynamically assigns jobs to available resources.

We have implemented such a resource sharing environment in cooperation with ABB corporate research. Design engineers can access the necessary resources for the complete optimization process from their desktop. The overall process has been integrated seamlessly into the design workflow. In addition, the infrastructure enables an efficient resource sharing across all corporate–wide resources, which eliminates long response times for optimization requests, allows for a higher system utilization, and significantly reduces the number of required software licenses and hence cost.

2 Optimization Process

In the area of numerical optimization, several software packages are available that enable the user to apply different algorithms to a specific simulation problem, most notably Optimus [7], the DAKOTA iterator toolkit [6], and Nimrod/O [5]. These tools, however, are primarily designed as universal optimization tools requiring significant changes for the use in electric field specific simulation. In addition, these system are unable to cope with limited resources during the model generation phase, as it is given here due to the CAD modeling. We have therefore decided to design our own optimization environment, which we will describe below.

2.1 Optimization Loop

Our framework consists of three primary components: a parametric CAD modeling system based on a commercially available system, a model evaluation component using field simulation, and the numerical optimization algorithm. The latter is designed to restrict the search space to a small subset for an optimal set of parameters.

The workflow between these components is as follows: Starting with an initial set of parameters, the CAD program generates the first instance of the model that is to be calculated. The model is prepared for the field calculation, i.e. the boundary conditions and dielectrics are assigned and the model is discretized. Using simulation, the quality of the generated model is computed and passed to the optimizer. Combined with the result of all previous simulation runs within this optimization invocation, the optimization algorithm defines a new set of design parameters. These design parameters are again read by the parametric CAD modeler, which creates a new instance of the model to be optimized.

In [4] several minimization algorithms have been investigated for a two dimensional field optimization problem. Based on the results of this study, the Hooke-Jeeves Method [8], the Nelder-Mead Method [10], and the Fletcher-Reeves Algorithm [12] have been used for the optimization process in this work.

This optimization loop is repeated until a termination criterion is reached. The exact criterion depends on the optimization algorithm in use. In any case, it indicates that a local minimum has been found. Due the physical properties of this particular problem, this will be either equivalent to the global minimum or sufficiently close to it.

2.2 Electrical Simulation

In order to facilitate the evaluation of generated models, a simulation environment for three dimensional electric fields in real-world high voltage apparatus is required. From a mathematical point of view, the calculation of fields can be described as the calculation of the potential $\Phi(x, y, z)$ and the electric field strength $\boldsymbol{E}(x, y, z) = -\nabla\Phi$. This can be achieved by solving Laplace's differential equation $\Delta\Phi = 0$. For electrostatic fields this differential equation can be solved by

$$\Phi(\boldsymbol{r}_p) = \frac{1}{4\pi\epsilon} \int\int\int \frac{\rho(\boldsymbol{r}_q)}{|\boldsymbol{r}_p - \boldsymbol{r}_q|} dV \tag{1}$$

with $\boldsymbol{r}_p$ being the radius vector in the respective point of interest and $\boldsymbol{r}_q$ being the radius vector in the integration point.

To solve these problems, Boundary Element Methods [3] have proven to be very efficient. They reduce the problem to a system of algebraic equations by discretizing the model's surfaces with well selected, small curvilinear patches (boundary elements) on the interfaces between media with different material characteristics. Over a boundary element, the field is expressed as an analytical interpolation function between the field values at the nodes (element vertices).

Based on these principles, a parallel simulation environment named POLOPT [2] has been developed in cooperation with ABB corporate research and is in production for the design of high-voltage equipment. It uses a master/slave approach for its computation and is implemented using MPI [9] as the parallel programming model. Encouraging results showing a high efficiency have been achieved under production conditions using large-scale input sets [11].

2.3 Implications

The overall optimization process requires repeated access to two kind of resources: compute servers to execute the electric field simulation and CAD servers to perform the model generation. The requests for these two are always interleaved and require both substantial execution time. It is therefore inefficient to allocate both resources throughout the whole runtime of the optimization process; instead both resources should be allocated on demand in order to allow a more efficient execution of concurrent optimization requests.

In addition, in a corporate environment these resources should ideally be shared across the whole company. Each of them is associated with a non-trivial amount of money (the compute servers are high–end clusters and CAD servers are associated with expensive CAD licenses). Such resource sharing allows an optimal utilization of each resource and hence the reduction of the number of required resources. This cuts cost and reduces the number of sites hosting the resources, which allows more centralized system management with a lower total cost of ownership.

3 Software Architecture and Implementation

This kind of corporate-wide resource sharing can be achieved in the form of a grid-like infrastructure. This grid splits into two main components: a) the grid for the compute servers and b) a grid for the CAD servers. As a result, the final system, which we call CAD grid, has to deal with two orthogonal sets of resource restrictions: the availability of a set of specific machines required for the simulation and the availability of CAD licenses (the CAD programs itself run on standard workstations).

3.1 System Structure

To achieve such flexible resource management we have designed a modular system. Its structure is depicted in Figure 1 and consists of the following five major components:

1. Clients, from which jobs get submitted.
2. A set of compute clusters to execute the field simulation, forming the simulation grid.
3. A set of CAD workstations to generate the models depending on varying input parameters, forming the CAD grid.
4. The optimization module executing the optimization algorithm.
5. A central coordinator, which retrieves jobs from the clients and is responsible for the execution.

The latter two components will usually be combined within one system, as the optimization algorithm does not require major compute times. Hence it is not necessary to offload this work to remote server farms analog to the simulation or the CAD grid. Note, that this central component, which forms the cornerstone of our framework, does not impose a major bottleneck, since the execution times of the model generation and the electric field simulations are several orders of magnitude higher than the one consumed by the coordinator.

This coordinator is implemented as a server, which acts as the interface for communication involving the clients. An important consequence of this design is that the client needs no knowledge of the overall architecture of the system. In particular, it need not know the location of the CAD hosts or the simulation clusters. Therefore, changes may be made to the rest of the system without altering the client's view of it. This ensures the easy-of-use of the overall system, as end-users generally should not be influenced by such configuration changes, and at the same time significantly increases the manageability and the ability to tolerate system faults.

3.2 Workflow

When a job is submitted by a client to the coordinator, all of the CAD files, which specify the model to be analyzed, are sent, along with any design constraints and the initial parameter values as specified by the user. The optimization program is then started and loops until it reaches an acceptable value. The loop consists of the following stages:

1. An available CAD server is selected depending on the availability of a (potentially floating) software license and the capabilities of the target machine. The CAD files and parameter values needed to generate the model are then sent to the chosen CAD server.
2. The CAD server generates the specified model. For this task it uses special API software allowing to connect to the CAD program. In our work, we used the CAD package Pro/Engineer together with the Pro/Toolkit API [1].
3. A file encapsulating the required details of the CAD model is returned to the coordinator.
4. The coordinator selects an available compute cluster within the simulation grid depending on the compute requirements for the chosen task and then sends the file containing the CAD to this system and initiates POLOPT.
5. POLOPT computes the electric field of the given model and computes the objective value used for the optimization (in our case the maximal field value).
6. The objective value is passed back to the coordinator.
7. The design parameters of interest and the objective value are handed to the optimization algorithm. It analyzes the value and, if appropriate, generates a new set of parameters and sends its result to the coordinator.
8. Depending on the result received from the optimization algorithm, the coordinator starts again with step 1) or aborts the loop and returns the objective value together with the final set of parameters and the last CAD model to the client.

After the completion of the iterative optimization process, the client receives the final result in the form of an optimized CAD model. This can then be displayed together with the final field distribution for a final investigation by the design engineer.

3.3 User Interface

In order to ease the use of the system, we have implemented a graphical user interface which provides a straightforward access to the optimization framework. Users can upload their initial files and specify the design parameters, which

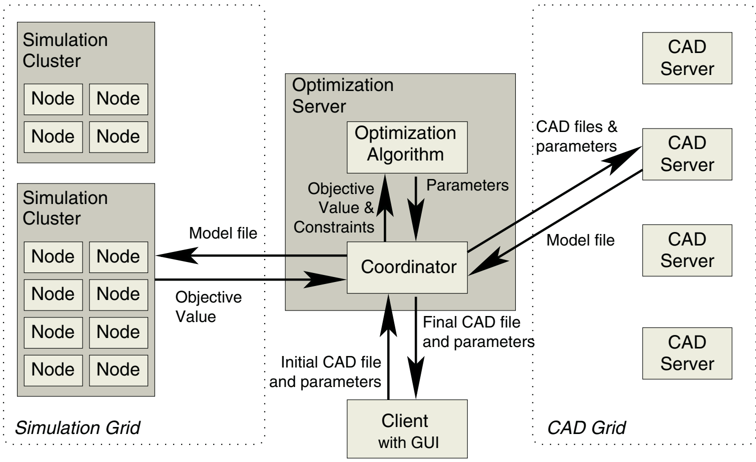

Fig. 1. System Structure

should be investigated by the optimization algorithm. After the job has been submitted, the user can monitor the progress of the optimization, and, after its completion, the final result is displayed. This integrates the optimization framework into the design workflow of the target users, the design engineers, and hence reduces the associated learning curve.

This GUI has been implemented in Java as a standalone client. This decision was made to enable the client to access local data and use this data for uploads, which would have not been possible in applets. This approach, however, maintains the platform independence of the client enabling an easy deployment across a large number of platforms without porting efforts.

4 Technical Evaluation

We have implemented this resource sharing infrastructure in cooperation with ABB corporate research and have deployed it at their site for production use. It currently operates with two PC clusters, each eight nodes, and up to three CAD workstations. Within the compute clusters, the POLOPT environment [11] is used to execute the simulations, while a set of CAD workstations with Pro/Engineer [1] is deployed to generate the model files. First experience shows a positive feedback and high acceptance rate among the design engineers. The unified GUI effectively hides the complexity of the optimization from the user and provides a clean interface, significantly lowering the learning curve. In addition, by its implementation in Java, it can be used from arbitrary workstations, further simplifying its application and deployment.

The availability of this unifying infrastructure provides the design engineers with a straightforward access to corporate-wide resources and enables the computation of complex optimization problems. This is illustrated in the following based on a sample model. Figure 2 shows eight design parameters of a transformer output lead shielding electrodes for which an optimal set with regard to the maximum field strength had to be found. The goal is the minimization of the overall space requirements by this component, while staying within a safe range in terms of possible flashovers.

This example has been optimized using the optimization algorithms discussed above. The numerical changes of the eight design parameter during the optimization process can be seen in Table 1.

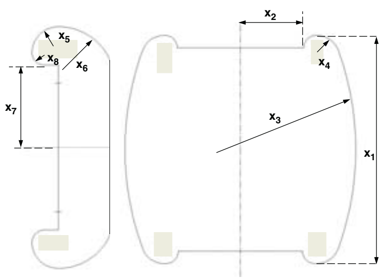

Fig. 2. Design parameters for the transformer

Table 1. Change of design parameters during optimization

Parameter	Initial Value	Optimal Value
$x_1 [mm]$	420.0	438.3
$x_2 [mm]$	113.0	17.3
$x_3 [mm]$	531.0	237.0
$x_4 [mm]$	42.0	135.1
$x_5 [mm]$	32.0	147.0
$x_6 [mm]$	113.0	236.5
$x_7 [mm]$	165.0	56.2
$x_8 [mm]$	25.0	60.7

5 Business Perspective

The ability to compute these optimization processes is invaluable to companies like ABB. They allow a predictable assessment of properties of their products and enable significant design improvements without having to build expensive and time-consuming prototypes. This leads to more competitive products with respect to both price, due to reduced engineering cost and development time, and quality, due to highly optimized systems. The exact impact can hardly be quantified, but can be assumed to be substantial considering that most components in this product area (e.g., large scale transformer and switching units) are custom designs without the ability to mass produce and can be in price ranges well beyond millions of dollars.

6 Conclusions

The optimization of the geometry is an integral part of the design process of high voltage components. This is done iteratively by repeated simulations of the electric field and an adjustment of the geometric model based on the simulation results. Combined with optimization algorithms, which are capable of minimizing a chosen objective value for a given set of parameters, this can be used to automatically compute an optimal set of design parameters. We have presented a framework, which implements such an optimization process. It is designed to efficiently leverage corporate wide resource and to allow an interleaving of several concurrent optimization requests. It thereby distinguishes between two different sets of resources — compute servers to run the electric field simulations and CAD servers to perform the model generation based on a given set of parameters — and ensures the efficient utilization of both.

This framework has been implemented in a real industrial environment and is already in production use. It is highly accepted by the end users, the design engineers, due to its easy-to-learn interface and clean integration into their overall workflow. It has already successfully been deployed to optimize the design of high voltage components. In the example shown in this work, it has lead to significant optimizations in the geometric design and to a reduction of the electric field by over 25%.

References

1. Web Page – URL: http://www.ptc.com.
2. Z. Andjelic. *POLOPT 5.3 User's Guide*. Internal document, Asea Brown Boveri Corporate Research, Heidelberg, 1999.
3. R. Bausinger and G. Kuhn. *Die Boundary-Element Methode (In German)*. Expert Verlag, Ehingen, 1987.
4. C. Trinitis, H. Steinbigler, M. Spasojevic, P. Levin, and Z. Andjelic. *Accelerated 3-D optimization of High Voltage Apparatus*. Conference Proceedings of the 9th Int. Symposium on High Voltage Engineering, paper 8867, Graz, 1995.
5. L. Kotler D. Abramson, J. Giddy. High performace parametric modelling with nimrod/g:ller application for the global grid? *International Parallel and Distributed Processing Symposium (IPDPS)*, May 2000. pp 520–528, Cancun, Mexico.
6. M.S. Eldred and W.E. Hart. Design and implementation of multilevel parallel optimization on the intel teraflops. *Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1998. pages 44-54, St. Louis, MO, AIAA-98-4707.
7. P. Guisset and N. Tzannetakis. Numerical methods for modeling and optimization of noise emission applications. *ASME Symposium in Acoustics and Noise Control Software*, November 1997. ASME International Mechanical Engineering Congress and Exposition, Dallas, TX.
8. R. Hooke and T.A. Jeeves. Direct Search Solution of numerical and statistical Problems. In *Journal of Ass. of Comp., vol. 8*, pages 212–229, 1961.
9. Message Passing Interface Forum (MPIF). MPI: A Message-Passing Interface Standard. Technical Report, University of Tennessee, Knoxville, June 1995. http://www.mpi-forum.org/.
10. J.A. Nelder and T. Mead. A simplex method for function minimization. In *Computer Journal 7*, pages 308–313, 1965.
11. C. Trinitis, M. Schulz, and W. Karl. *A Comprehensive Electric Field Simulation Environment on Top of SCI*. Lecture notes in Computer Science 2474, EuroPVM'2002, Springer Verlag, pp. 114–121, 2002.
12. G.N. Vanderplaaats. *Numerical Optimization Techniques for Engineering Design*. Mc-Gaw Hill Book Company, New York, 1994.