Applications of Parallel Computing in the Utility Industry

Paul Coddington

Northeast Parallel Architectures Center
Syracuse University, Syracuse NY 13244, U.S.A.

Phone: 315-443-4883
FAX: 315-443-1973
Email: paulc@npac.syr.edu

Extended Abstract

1 Introduction

As part of its InfoMall industrial outreach program, the Northeast Parallel Architectures Center (NPAC) at Syracuse University is involved in a project with Niagara Mohawk Power Corporation, the utility serving central New York state, to investigate the application of parallel computing to utility applications, especially for computer simulations used to manage large electrical power networks.

Power utilities have two main constraints – security and economy. The power network must be operated in such a way that it is stable under any reasonable conditions, including possible equipment failures. This generally requires providing extra generation and transmission capacity. The other major constraint is of course economy. Utilities would like to provide the most cost-effective generation and transmission of the power needed to cover demand and meet security constraints. A utility will have available many generators with different generation and transmission costs. Choosing which generators to run and how the power required to match the forecast demand should be distributed among the generators is a complex optimization problem. Environmental constraints are also becoming more important.

Computer simulation plays a major role in both long-term network planning, and the on-line Energy Management System used by control room operators handling the running of the power grid. Many of the simulations are compute-bound, and the use of high performance computing would allow more accurate and/or more frequent simulations to be performed.

2 Security Analysis

An important aspect of the operation of an electrical power network is the security model, which simulates the response of the network to disturbances such as component failure. These simulations allow power utility control room operators to identify and remedy possible sources of cascading failures (one failure causing another component to be overloaded, resulting in another failure, and so on), which could cause a disruption of power generation or transmission. The necessity for this type of simulation was made clear in 1965 when a fault at a Niagara Falls generating unit caused cascading failures which blacked out the entire north-eastern United States.

In order to maintain fault tolerance in the power network, extra generation and transmission capacity must be available. There is a certain margin of error factored in to this extra capacity due to the limited accuracy of the simulations. Faster and more accurate simulations would allow improved system security and reduced operating margins, which would save energy and money, and reduce environmental impact. Due to the huge scale of the power industry, even a 1% reduction in extra generation requirements translates into large savings. Estimates have indicated that improved simulations could save some US utilities millions of dollars per year.

2.1 Contingency Evaluation

Typically the security model is run every few minutes using data on the current state of the network, which is provided every few seconds by telemetry units positioned throughout the network. This raw data gives an over-determination of the variables required to solve the electrical network equations. This is necessary since not all telemetry units will be accurate, or even functioning. A *state estimator* program is first run to make a statistical analysis of the raw data to determine the most likely true state of the network, and this data is used for all subsequent simulations. State estimation, like many other power system simulations, involves the solution of large sparse matrix equations. The sparse matrix is generally representative of the power grid, which is very sparsely connected.

Utilities will generally have lists of many hundreds of *contingencies*, or possible component faults. Most of these contingencies cause very little disruption to the network, so to reduce computation time a *contingency screening* is performed. This is just a fast, very approximate calculation which can eliminate those contingencies which are very stable. This process generally leaves around 30 contingencies which are evaluated using the full simulation. The results are compared to the system stability limits, and the contingencies are then ranked in order of severity. If a contingency is so severe that it could cause problems, some action is taken by the operator to remedy the situation.

This type of simulation is easily handled in parallel, since all the contingencies are independent, so job-level parallelism can be used for both the contingency screening and the full simulations by farming out the work, that is, sending the next contingency to the next processor which becomes idle.

Contingency evaluation uses component stability limits which are determined very infrequently by planning engineers, using worst-case scenarios. Improved stability limits based on real-time data would greatly improve the accuracy of security model simulations, reduce the amount of excess energy produced in order to satisfy security constraints, and increase the amount of power transmission allowed, perhaps allowing regions where generation costs are high to import more power from areas with cheap excess generation. However calculating stability limits on-line requires a large increase in the speed of stability simulations, which currently take many hours. Parallel computing could be used to speed up both on-line simulations and the off-line stability studies which are currently performed.

2.2 Transient Stability Analysis

Transient stability analysis involves the study of the electrical network and generator dynamics for several seconds following a disturbance. It is one of the main factors used by power network planners to set tolerance limits for network components which allow stable operation of the system. Due to the computational complexity of transient stability analysis, it currently is not used for on-line simulation. Instead, contingency evaluation is done using the steady-state solution after the fault, using approximations which neglect the dynamics of the system.

The dynamics of each generator in the system is described by multiple non-linear differential equations, while the coupling of the generators to the electrical network is described by non-linear algebraic equations. Together these lead to a system of differential-algebraic equations (DAEs) that must be solved simultaneously. Since a large system of DAEs needs to be solved at small time increments to minimize the errors in numerical integration of the differential equations, the transient stability analysis is a highly compute intensive problem. The problem is very similar to that of circuit simulation, which utilizes programs such as SPICE.

Transient stability analysis generally involves simulating the dynamics of the system for around 10-20 seconds after the fault. The equations are solved using time domain techniques, with integration time-steps of around 0.1 seconds. The DAEs are solved iteratively by approximating them by decoupled linearized differential and algebraic equations. The differential equations for the generators are integrated using a simple implicit trapezoidal rule method. The algebraic equations for the network are solved using a direct sparse matrix solver.

The solution of the generator equations can be done independently for each generator, which should give 100% efficiency. However solving the network equations in parallel is much more difficult, since these involve a parallel sparse matrix solver. Previous work in this area has given very poor results, with speed-ups typically in the range 2-5. This has been the major bottleneck in using parallel computers to speed up power systems applications. To our knowledge, parallel transient stability programs have never achieved speed-ups of greater than 10, no matter how many processors were used.

In a typical production transient stability program, roughly 75% of the time is spent in solving the generator equations, and 25% in solving the network equations. Thus even if 100% efficiency could be achieved for solving the generator equations, the total speed-up would only be 4 if the network equations cannot be parallelized effectively. Work on parallel sparse matrix algorithms will be discussed in section 2.5.

2.3 On-line Dynamic Security Assessment

The major research co-ordinator for US utilities is the Electric Power Research Institute (EPRI). EPRI has an ambitious plan to develop an on-line dynamic security assessment (DSA) program over the next few years. The goals of the EPRI DSA project are to provide a full contingency evaluation incorporating system dynamics (i.e. transient stability analysis), rather than the static on-line contingency evaluation that is currently done by utility Operations centers. The program would also provide on-line dynamic stability limits for control room operators (rather than the fixed limits currently provided them), so the operators would know not just whether the system was currently fault tolerant, but how much they could stress the system before running into problems.

A simple way of exploiting parallelism in an on-line DSA program is to parallelize over contingencies, just as for static contingency evaluation. The prototype EPRI program will allow this independent parallelism on a workstation network. The program enables the user to specify computers in the network which can be used, and different contingencies are sent to these machines when they are idle.

This method cannot be used for other sections of the DSA program, however. For example, the stability limit for a component is calculated by taking the worst contingency for that component, and using an iterative procedure to increase the limit until the system becomes unstable. Speeding up this part of the DSA program would require parallelizing the transient stability analysis program. There are other areas such as state estimation which may require parallelization to either avoid being a bottleneck in the overall program, or to allow more accurate methods to be used in the same amount of time.

2.4 DAE Solvers

Traditional approaches to solving transient stability DAEs, such as partitioning and linearization, have used simple trapezoidal integration techniques that require very small step sizes during the integration steps. Meanwhile, there has been significant research in the numerical analysis community on fast, accurate, and efficient DAE solvers. However, very little work has been done to date on using these more recent public domain DAE solvers for the transient stability problem.

We are investigating the applicability of these DAE solvers to the power systems problem. In particular, we have been involved in the development of a parallel DAE solver, CDASSL, which has been successfully used in the simulation of a petrochemical plant. We are studying whether CDASSL could also be used for transient stability analysis.

More efficient and accurate DAE solvers would be especially effective for long-term stability studies, where the time interval studied is on the order of minutes rather than seconds. Currently the time taken to run these simulations is extremely large. Good DAE solvers would allow longer integration timesteps, and give more accurate results in a shorter simulation time.

2.5 Parallel Sparse Matrix Algorithms

Sparse matrix calculations are some of the most difficult problems for parallel machines, but they are key to many power system applications, including load flow and transient stability analysis, for which they represent a significant portion of the computation. Various universities and government laboratories have efforts in progress to develop general concurrent sparse matrix solvers. However, specialized parallel sparse matrix solvers which can capitalize on any regular structure in the sparse matrices for a particular problem will usually give better performance than general solvers. Previous work on parallel sparse matrix algorithms for power systems problems have given very poor results, with maximum speed-ups of typically 5 or less. Research at NPAC is currently focusing on the development of specialized direct sparse matrix solvers that exploit the hierarchical structure in power systems problems.

There are distinct portions of the matrix that represent the generator equations and the power distribution network. The whole matrix is of a block-diagonal-bordered form, where the blocks on the diagonal represent the equations for the different generators and the last block represents the network equations. The borders represent the coupling equations between the generators and the network. It is possible to reorder the network matrix so that it is also in block-diagonal-bordered form. The result of this reordering is that many of the operations that would be sequential in a general solver are now highly parallel. In particular, the blocks can be factorized (using LU or Choleski factorization) independently, and the data can be distributed in such a way that no communication is required in this step. The only communication that is required is in dealing with the borders and the last block, and we have used a technique that minimizes this overhead. Reordering the matrix adds some fill-in which generally makes the last block fairly dense, so we have used standard dense matrix algorithms utilizing efficient BLAS (Basic Linear Algebra Subroutines) to deal with the last block. Reordering can also be used to improve load balancing, by ensuring that the blocks have approximately the same number of non-zero elements.

Preliminary results for our sparse matrix solver have shown good speed-ups (in excess of 10 on 16 processors) for factorization, and reasonable performance for forward reduction and single backward substitution as well. These are much better than previous results in this area. Parallel sparse matrix algorithms are known to scale very poorly, however this is not a major concern since utilities are much more likely to buy a small parallel machine or workstation cluster than a massively parallel supercomputer.

We are beginning to port our sparse matrix code (currently written using CMMD) to a portable message passing language, with the aim to eventually implement it using the Message Passing Interface standard.

3 Optimization Problems

A utility has available many generators with different generation and transmission costs. Generators have to be shut down for scheduled maintenance or unscheduled problems, or when there is not enough demand for them to run. The demand for power (the load) varies substantially through daily and yearly cycles. However generators generally take hours to shut down or bring on-line, so it is necessary to start up generators well in advance of when they are needed. The start-up and shut-down costs must also be considered. The problem of scheduling which generators to commit to power production is called *unit commitment*, and is a difficult combinatorial optimization problem. The number of possible configurations is two to the power of the number of generators times the length of time studied. For a typical utility, the number of generators may be in the hundreds, and the time studied may be hourly intervals for a day or week. Clearly this is not a problem which is amenable to an exhaustive search.

A standard method of approaching the unit commitment problem is the use of dynamic programming optimization techniques, incorporating a prioritized list of generators with the cheapest generators given higher priority. This technique involves a search at each time interval, and the search space can be easily distributed over processors so the search can be done in parallel. Preliminary results have shown good speed-ups for small problems, and the efficiency of this method should improve with larger problem sizes.

A related problem is *economic dispatch*. Given that certain generators have been committed based on the load forecast, the problem is to utilize these generators in the most cost effective way given the current load conditions.

Models to optimize costs can be very complex and time-consuming, especially for long-range planning studies. Integrating pricing and security models would demand even greater computational power, and parallel computing is an attractive option for this type of important and highly compute-intensive application.

4 Conclusions

EPRI (the research arm of US the utility industry) are interested in utilizing parallel computing in the future, and have sponsored a number of research projects in this area, however they are currently adopting a "wait-and-see" approach. There are two main reasons for this.

Firstly, it is perceived that parallel languages, software and hardware are not stable or standardized enough to be used for production programs. The development of standardized parallel languages such as High Performance Fortran and MPI, and the availability of working compilers for these languages on a variety of parallel machines, is crucial for the acceptance of parallel computing in the utility industry (and probably for industry in general). Many utilities are quite conservative about computing facilities – the industry is still making the transition from mainframes to workstations, and so are generally not ready to commit to parallel computers, particularly not the more "exotic" machines. Workstation clusters or machines such as the IBM SP1 are the only likely purchasing option for most utilities.

Secondly, the type of applications for which parallel processing has been studied typically involve sparse matrix solvers, which are one of the most difficult applications to parallelize efficiently. Previous research projects on methods for parallelizing the transient stability program have usually attempted to use parallelizing compilers and/or generic parallel sparse matrix algorithms, and have not surprisingly produced poor results. More efficient parallel sparse matrix solvers will have to be developed before the use of parallel computers is warranted for problems such as transient stability analysis. Again, the utility industry is only just beginning to take advantage of the increased performance of modern high-speed workstations, so parallel computers will need to show a large price/performance advantage over single processor workstations before they will be adopted in the industry.

Another problem is that most power systems simulation software consists of Fortran programs which have been developed by engineers over many years. They are generally not written in a very modular fashion, and in most cases the loops which could provide parallelism (for example, in the solution of the generator equations, which are independent) are not coded in a "clean" way with no data dependencies between the loop iterations. The programs are also quite large, typically of the order of 100,000 lines. Thus, even when parallelism is inherent in the problem, it may require substantial effort to restructure the code in order to exploit this.

Probably the best initial applications of parallel computing in the utility industry would be the use of job-level parallelism for evaluating independent contingencies in security models, or the parallelization over the search space of optimization techniques such as dynamic programming for unit commitment. However we believe that many power systems applications, including transient stability analysis, could be parallelized efficiently using High Performance Fortran and specialized parallel sparse matrix libraries.

Utilities are also concerned with other types of stability problems apart from transient stability, in particular *voltage stability* and *small signal stability*. This type of analysis requires the solution of large sparse eigenvalue equations, which should also be amenable to the use of parallel algorithms.

We are also surveying other possible applications of parallel computing to the utility industry, which could span the range from data processing to the simulation of nuclear power plants or the flow of gas through pipelines.