Computational Optimization and Applications, 29, 123–126, 2004

COAP 2003 Best Paper Award

In each year, starting 2004, the Computational Optimization and Applications (COAP) editorial board will select a paper from the preceding year's COAP publications for the "Best Paper Award." The initial recipients of the award are Jeff Linderoth, Lehigh University, and Stephen Wright, University of Wisconsin-Madison, for their paper "Decomposition Algorithms for Stochastic Programming on a Computational Grid," published in Volume 24, pages 207–250.

This paper describes work carried out by the authors at Argonne National Lab in 1999–2001 under the auspices of the metaNEOS Project, which was supported by NSF from 1997–2001 and by Argonne's DOE sponsors. metaNEOS was a collaboration between specialists in algorithmic optimization and in grid computing (still known in 1997 as "metacomputing"); in particular, representatives of the Globus and Condor projects. The purpose of metaNEOS was to use the power of computational grids to solve a variety of very large and challenging optimization problems, including problems in combinatorial optimization, global optimization, and stochastic optimization. The research involved developing middleware (software toolkits and libraries that facilitated implementation of a variety of algorithms on the unreliable, heterogeneous grid platforms at hand), developing new algorithms that could exploit the power of the grid platforms while not being affected too seriously by its less felicitous features, and finally implementing these algorithms and using the resulting codes to solve touchstone problems in optimization and to perform computational investigations that were not previously possible.

Work on this paper began in the fall of 1999, when metaNEOS researchers had decided to develop a runtime support library (soon named MW) as a vehicle for implementing algorithms of master-worker type on computational grids running the Condor system. Condor, developed in the Computer Sciences Department at Wisconsin by a group headed by Miron Livny, manages distributively-owned collections ("pools") of processors spread across a campus or organization. The owners of each machine specify the conditions under which Condor can schedule other jobs on their machine, to ensure minimal intrusion with their own workload. Thus, Condor "scavenges" computational cycles that would otherwise have been wasted and puts them to work for other users. Condor allows a single workstation to submit multiple jobs into the pool and to remain in contact with each of them, thereby providing a natural basis for parallel computations of the master-worker variety, with the submitting workstation playing the role of master. The MW library builds on Condor's resource discovery and communication features (including the Condor implementation of the parallel processing protocol PVM), and adds features to allow transparent rescheduling of tasks in the event that the worker assigned to that task is unable to complete it. (MW has since been abstracted to run on other parallel platforms in addition to Condor and PVM). Users wishing to implement their master-worker algorithm using MW must define ten methods in C++ to specify such operations as initialization of a worker, description of a single task, and the action that the master should take in response to a completed task.

During initial development of MW, Linderoth and Wright sought applications that could be used to test it and drive its evolution. Two-stage stochastic linear programming was a fairly obvious candidate. The master process develops a lower bounding piecewise linear approximation to the first-stage objective, where the approximation is made up of cuts returned by the second-stage problems. Solution of the second-stage problems could be carried out by workers; a number of such problems are bundled into an individual task, and each task can return a specified number of cuts. Their initial code implemented the well known L-shaped algorithm (multicut Bender's decomposition). Asynchronicity was introduced into this algorithm, allowing the master problem to choose a new iterate without waiting for all the second-stage problems to solve to completion. (Such a feature is important on a grid platform, when heterogeneity and unreliability of the workers makes it inefficient to wait for all of them to complete their specified tasks.) Later, in mid-2000, they added a box-shaped trust region to restrict the size of the first-stage steps. The appeal of an ℓ_∞ trust region is that it allows the first-stage iterates to be obtained by solving a linear program, but anecdotally their performance was thought to be inferior to methods that used a quadratic regularization term and which therefore required a specialized quadratic programming code for their solution. (We did not find evidence to support this belief.) The paper contains an analysis of an asynchronous version of this algorithm. It also contains details of the implementation called ATR (for "asynchronous-trust-region"). The implementation reported in the paper used CPLEX to solve the master problem and SOPLEX to solve the second-stage problems on the workers. Results were reported for sampled approximations of a number of standard problems from the stochastic programming literature. One such problem was a flight mobilization model in which we sampled ten million scenarios, yielding a linear program with a total of over 10^{10} unknowns. Using a computational grid of over 1000 processors spread across the U.S. and Europe, the problem was solved in a little over one day of wall clock time.

In a later collaboration with Alexander Shapiro, the ATR code was used to perform extensive computational studies of sampled approximations to various problems, and to obtain estimates and variances of lower and upper bounds to the solutions of these problems [7].

Jeff Linderoth was also part of the team that solved the nug30 quadratic assignment problem, described in the paper [1], winner of the SIAM Activity Group on Optimization Prize in 2002, along with fellow metaNEOS research Jean-Pierre Goux and collaborators Kurt Anstreicher and Nate Brixius of the University of Iowa. That happy collaboration arose almost by accident from a summer internship that Nate spent at Argonne in 1999, ostensibly to work on a different project. metaNEOS researchers also played a part in solving the Seymour integer programming problem [5], and in developing solvers for linear integer programming [3, 4], and nonlinear integer programming [6].

Work continues on various offshoots of the metaNEOS project. The ATR code was analyzed further in a collaboration with performance modeling experts [2]. The current version of the code uses the freely available COIN-OR simplex solver CLP to solve both master and worker problems. Support was recently obtained from NSF for further development of the MW toolkit, and it is now being used in a number of new applications. Jeff and his student Jierui Shen are working on an extension of the two-stage stochastic programming code to multistage problems.

Further details about the metaNEOS project, and the Condor and MW software frameworks can be found at the following web sites:

- `http://www.mcs.anl.gov/metaneos`
- `http://www.cs.wisc.edu/condor`
- `http://www.cs.wisc.edu.condor/mw`

References

1. K. Anstreicher, N. Brixius, J.-P. Goux, and J.T. Linderoth, "Solving large quadratic assignment problems on computational grids," Mathematical Programming, Series B, vol. 91, pp. 563–588, 2002.
2. D. Buaklee, G. Tracy, M. Vernon, and S.J. Wright, "An adaptive model to control a large scale application," in Proceedings of the 16th Annual ACM Intl. Conf. on Supercomputing (ICS 2002), 2002.
3. Q. Chen and M.C. Ferris, "FATCOP: "A fault tolerant Condor-PVM mixed integer program solver," SIAM Journal on Optimization, vol. 11, pp. 1019–1036, 2001.
4. Q. Chen, M.C. Ferris, and J.T. Linderoth, "FATCOP 2.0: Advanced features in an opportunistic mixed integer programming solver," Annals of Operations Research, vol. 103, pp. 17–32, 2001.
5. M.C. Ferris, G. Pataki, and S. Schmieta, Solving the Seymour problem, Optima, vol. 66, pp. 1–7, 2001.
6. J.-P. Goux and S. Leyffer, "Solving large MINLPs on computational grids," Optimization and Engineering, vol. 3, pp. 327–354, 2003.
7. J. Linderoth, A. Shapiro, and S.J. Wright, "The empirical behavior of sampling methods for stochastic programming," Annals of Operations Research (2004), forthcoming.

Jeff Linderoth received a Bachelor's degree in General Engineering from University of Illinois at Urbana-Champaign in 1992, a Master's in Operations Research from Georgia Tech in 1994, and his Ph.D. from Georgia Tech in 1998. On graduating, Jeff went to the Mathematics and Computer Science Division at Argonne National Lab, first as a postdoctoral scholar, and then as the Enrico Fermi Scholar. In 2000, he joined the Optimization consulting firm of Axioma, Inc., and subsequently moved to Lehigh in 2002.

Steve Wright received his Bachelor's of Science (Hons) in Mathematics in 1981, and a Ph.D. in Mathematics in 1984, both from the University of Queensland. Steve moved to the U.S. in November 1984 and has remained there since, holding visiting appointments at Arizona (1984–1986), academic appointments in the Mathematics Department at North Carolina State (1986–1990), and staff positions in the Mathematics and Computer Science Division at Argonne National Lab (1988–1989 and 1990–2001). He moved to UW-Madison in August 2001.