Critique of A Bridging Model for Parallel Computation

Xin Wang

1 Summary

The paper "A Bridging Model of Parallel Computation" [3] introduces a bridging model between hardware and software programming, called bulk-synchronous parallel(BSP) model. The BSP model is a promising bridging model for general-purpose parallel computation and it aims to enable parallel programs to be executed efficiently on a variety of architectures. The BSP computers(BSPC) include three components: some unit processors for computation or memory functions, a router to communicate and deliver messages among the processors, and barrier synchronization among a set of processors at L time units. The computational tasks and communication can be performed by components separately. In each superstep, different components work seperately, then send messages and wait for synchronization before moving to the next superstep. Then the automatic memory management on BSPC, the concurrent memory accesses on BSPC, BSP algorithms without hashing, the implementation on packet switching networks and implementation on optical crossbars are also discussed. Advantages of BSP model lie in three aspects. The first advantage is the good programmability given balanced computational and communication bandwidth. Second, BSP model provides a good platform for important algorithms and controls memory management overheads. Third, BSP model can be related to different technology and topology on networks.

2 Positive reactions

The BSP model provides an innovative idea for establishing bridging model between hardware and software programming in parallel computation in 1990. The superstep concept enables computation and communication actions wait for each other without strict order. The superstep ends with barrier synchronization which ensures all one-sided communications. In theory, the BSP model enables asymptotically optimal execution on a variety of computer architectures. As mentioned above, BSP model is advantagable in its programmability, portability and efficiency. The success of Apache Hama, Apache Giraph and Pregel has demonstrated the merits of BSP model. The authors have proposed some possible extensions of BSP model such as parallel-prefix computation BSP, broadcasting BSP and concurrent reads and writes BSP [1]. In fact, a variety of more optimization algorithms and architectures can be applied to the platform that BSP provides so that different level of parallism are implemented. For example, sorting the processors that are currently working in local memory according to an optimal time sequential algorithm so that the remaining tasks are balanced and efficiently partitioned. A dynamic programming algorithm can also be applied so that overload components can share tasks with idle ones. Ref [2] is an example of this kind of extented BSP model with deterministic optimization algorithms.

3 Negative reactions

Even though the BSP model is carefully designed and simulations for BSP have provided good results, certain limitations can still be found in the initial version of this model. First, BSP model is established according to a batch-routing interconnection network without imbalanced communication problems. However, in real world, different kinds of underlying network topology should not be ignored. The model should take more complex topology into consideration rather than network consisting only of local processors and non-local ones.

Efficient rounting algorithms that are designed specifically to different level of network topology may improve efficiency. For example, when large groups of processing components from different places are clustered to form a more complex topology, a hierarchical topology can be considered. The hierarchical structure may need different level of synchronization in subsets and therefore lead to different level of parallelism. Second, complete communication among all processors may sometimes be less efficient especially when the message amount is small and routing topology is complex and time consuming. Instead of sending one message after another one as it may be the case in BSP model, routing large amoung of messages together may further improve efficiency.

4 Extension proposal

Though some useful BSP libraries have been implemented for different purposes and structures already, design and implement one library with good interface for users are very helpful. Such an interface should free users from worrying about the network topology, the workload on different processors, and the rounting phase contention. At the same time, dynamic optimal algorithms are called automatically to fit specific structures and tasks. The interface should also provides a virtual shared memory image for users.

5 Conclusions

To sum up, the paper proposes a useful bridging model for parallel computation with good adaptivity to different platforms, programs and structures as well as with efficiency. It also provides useful ideas to extend the model for different parallism scheme.

References

[1] M Cemil Azizoglu and Omer Egecioglu. Lower bounds on communication loads and optimal placements in torus networks. *Computers, IEEE Transactions on*, 49(3):259–266, 2000.

[2] Alexandros V. Gerbessiotis and Constantinos J. Siniolakis. Deterministic sorting and randomized median finding on the bsp model, 1996.

[3] Leslie Valiant. A Bridging Model for Parallel Computation. *Communications of the ACM*, 33(8):103–111, August 1990.