Scheduling algorithms research with Gruia Calinescu

We propose investigating the algorithmic aspects of assigning jobs to compute nodes together with the scheduling the transfer of files between the cache of the processing and the network attached storage, with the objectives of minimizing makespan and/or communication costs.

The easiest interesting model has variable-sized files and one cache, and the goal is to minimize the total I/O cost of the disk by determining whether the accessed files should be placed in the cache. Two offline algorithms for it have been proposed by [6], a paper among whose co-authors are the PI and one PhD advisee of Gruia. One is an exact dynamic programming algorithm that only works when the number of distinct files is very small; however, in real applications the number of accessed files could be as large as 10,000, which makes the dynamic programming approach feasible. The other is a fast heuristic (no performance guarantee). It turns out that Irani [4] had already introduced this model, and using an involved rounding method of linear programs, solved this problem with a $O(\log k)$-approximation ratio, where k is the ratio of the size of the cache to the size of the smallest file. In work with REU students Andrew Choliy and Max Whitmore presented as a poster at The International Conference for High Performance Computing, Networking, Storage and Analysis (SuperComputing 2017), we obtained a 4-approximation algorithm, using ideas from Albers et al. [1] and Bar-Noy et al. [2]. Chrobak et al. [3] showed that the problem is NP-complete. When the number of distinct files is large, on synthetic data, the 4-approximation gives the best quality solutions among several natural heuristics, including the one from [6].

We propose to investigate the harder model when a number of compute nodes each has local memory as well as access to a cache and a permanent memory, as in [5]; however we consider the offline version where the file requests are known in advanced, and also files are of variable size. We also propose working on more complex models that require assigning jobs to compute nodes as well, taking into account the bounds on computing power of each node.

References

[1] Susanne Albers, Sanjeev Arora, and Sanjeev Khanna. Page replacement for general caching problems. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '99, pages 31–40, Baltimore Maryland, 1999. Society for Industrial and Applied Mathematics.

[2] Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, and Joseph (Seffi) Naor. A unified approach to approximating resource allocation and scheduling. *Association for Computing Machinery (ACM)*, 48(5):1069–1090, 2001.

[3] Marek Chrobak, Gerhard J. Woeginger, Kazuhisa Makino, and Haifeng Xu. Caching is hard - even in the fault model. *Algorithmica*, 63(4):781–794, 2012.

[4] Sandy Irani. Page replacement with multi-size pages and applications to web caching. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, volume 3 of *STOC '97*, pages 701–710, El Paso Texas, 1999. ACM.

[5] Ioan Raicu, Ian T. Foster, Yong Zhao, Philip Little, Christopher Moretti, Amitabh Chaudhary, and Douglas Thain. The quest for scalable support of data-intensive workloads in distributed systems. In Dieter Kranzlmüller, Arndt Bode, Heinz-Gerd Hegering, Henri Casanova, and Michael Gerndt, editors, *Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing, HPDC 2009, Garching, Germany, June 11-13, 2009*, pages 207–216. ACM, 2009.

[6] Dongfang Zhao, Kan Qiao, and Ioan Raicu. Hycache+: Towards scalable high-performance caching middleware for parallel file systems. In *IEEE/ACM CCGrid '14*, 2014.