Speculative Execution for Information Agents

Greg Barish, Craig A. Knoblock, Steven Minton

Information Sciences Institute and Department of Computer Science
University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292
{barish, knoblock, minton}@isi.edu

Practical deployments of information agents can suffer from sub-optimal performance and scalability for a number of reasons. In the case of web-based information integration, for example, data sources are remote and their latency can have a substantial effect on overall execution performance. Scalability can also be poor, since concurrent queries can cause multiple, simultaneous remote data retrievals (often of the same information), quickly consuming available bandwidth. The frequency of remote retrievals also makes such agents inherently I/O-bound, wasting CPU cycles.

One way of optimizing execution in such scenarios is to engage in speculative execution. Tasks likely to be executed in the future can be performed in advance, such as when an agent is I/O-bound. Correctly guessing can be profitable – the overall end-to-end application could perform faster, bandwidth could be conserved, and the CPU could be scheduled more optimally. Still, designing a technique for speculative information agent execution is not simple. For one, there are competing justifications. For example, an agent could speculate based on the profile of a current user or on the activity of past clients, or on resource availability. Secondly, speculation itself incurs additional overhead: it should not interfere with normal execution and there must be some method of coordinating speculative execution between multiple agents. Third, speculative execution itself needs to be scalable - hinting done on a per-user basis could lead to a prohibitive number of hints in popular agents.

Consider an "information portal" application, where users can view news headlines and stock quotes and charts. Suppose that 90% of the users currently logged into the system have portfolios that contain Cisco Systems stock. In this case, we could speculate that at least one will request detailed information about this stock and thus we could pre-fetch this data, or just the costly part (for example, the chart graphic).

Note that the idea of speculative execution is not limited to pre-fetching data. Other costly operators (i.e., relational joins) could be pre-executed based on branch and data value predictions. The only requirement of speculatively executing an operator is that there exist a way to "undo" the speculated action, should a guess be wrong.

Our research deals with specifying an approach for the scalable speculative execution of information agents. In particular, we are exploring how different *speculative contexts* can be integrated to reduce the risk of speculation. Contexts are different paradigms under which speculative hints can be generated. Example contexts include: the availability of system resources, past user agent requests, and the list of most recently requested agents. Each context can establish its own speculative hints. Hints can then be merged to determine optimal speculation. We are also interested in recovery when making bad guesses as well as making speculative execution itself scalable. In terms of the latter, we are exploring *decentralized speculation* and *speculative bundling*. Decentralized speculation is a technique allowing individual operators (or agents) autonomy in speculation - avoiding the need to have global entity to deal with all of the speculative choices in the system. Speculative bundling suggests that, under heavy load, many speculative hints will be generated, and that it can be more efficient to periodically merge groups of hints, rather than to react to each one.

Speculative execution has historically been associated with computer architecture and compilers. Recently, it has been successfully applied to workflow (Hull et. al. 2000) and operating systems (Chang et. al. 1999). Based on these results and the opportunities for execution in I/O-bound agents, we believe speculative execution is an exciting direction for future research.

Chang. F.; Gibson, G.A. 1999. Automatic I/O hint generation through speculative execution. *Proc 3rd Symposium on OS Design & Implementation*

Hull, R; Lirbat, F; Kumar, B; Zhou, G; Dong, G; Su, J. 2000. Optimization Techniques for Data-intensive Decision Flows. *ICDE-00*