PAMP: Categorized and Specialized Caching for SMPs

– Requested Additional Information

Erik Hagersten
Dept of Information Technology
Uppsala University
S-751 05 Uppsala, Sweden

1 Requested Information

This document contain additional information regarding requested by the external evaluators

1. The relationship and collaboration with Swedish industry is not clear. What is Ericsson's role?
2. Why will the proposed research lead to viable results. Similar approaches have been tried before for other types of applications.

2 The Role of Ericsson

As stated in the letter from Ericsson, the research topic is very interesting to them indeed. The proposal addresses two relevant questions: how to best make use of commercial caches for their kind of workload and how to improve the real-time properties of certain parts of their workload. Per Holmberg, a computer architect at Ericsson UAB, will be the technical contact at Ericsson.

- He will provide the project with appropriate "benchmarks" to make sure the outcome will be relevant for their problem areas.
- He will participate in project steering meetings
- He will stay close to the development of the project
- His position in Ericsson's organization allows him to make use of the results, as a whole or in parts, in future designs.

3 Related Work and the Project Plan

A computation typically consist of accesses of different distinct data types, such as write-once, write-many, producer-consumer, private, migratory, result, read-mostly and synchronization. This has been reported by Bennett, Weber, Archibald etc. As stated in the application, several dynamically adaptive protocols have previously been proposed that detects and optimizes the treatment of certain categories. These protocols adjust some system behavior according to how data is being used. Most such proposals alter the coherence protocol used to handle the cache line, page or data object. Most of them have been very successful.

- **Migratory Sharing** Various proposals by Stenstrom etc. The basic idea is to detect cache lines to which migratory traffic is common and to optimize the coherence protocol applied to the cache line accordingly. Very successful.
- **Update/Invalidate** Various proposals by Archibald, Hagersten, the SPARCcenter2000 team, etc. The basic idea is to detect producer/consumer cache lines and to apply an appropriate protocol variation to them. Questionable success.
- **Multicast/Point-to-Point Protocol** This idea was presented by Hill and Wood at this year's ISCA. A better scalability is achieved while maintaining a low communication latency. The proposal is to dynamically detect the anticipated sharing pattern and to chose the appropriate coherence mechanism per cache line. Very successful.
- **NUMA/COMA** Various proposals by Falsafi, Hagersten, Ekanadham etc. The basic idea is to categorize pages into coherence traffic pages and capacity traffic pages and to apply a suitable "caching strategy", NUMA or COMA, to the pages. Very successful.
- **Re-mapping addresses of conflict misses** This was proposed in a paper at ASPLOS 1996?. Pages with frequent conflict misses are detected and relocated in memory. Fairly successful.

Some proposals have added optimizations support for different categories of data in the cache coherence protocol and rely on user assistance, or compiler technology, to chose the most efficient protocol handling, such as proposals by Hill, Wood, Bennett, etc.

Categorization of data in combination with protocol optimization has also been very common in the area of Software-based distributed shared memory with many proposals from Bennett, Carter, Li etc.

So, there are strong indications that dynamic detection and adaptive algorithms indeed can be beneficial to system performance.

Our proposal is different in that we try to find the most optimal use for a cache memory by deciding which data objects should get cached and, if so, where they should get cached. We do not propose an alteration of the coherence protocol as most other proposals.

Caching a data object may actually result in a slower execution if the data object replaced is more likely to get re-used. Ericsson's applications have a mixture of data with high and low probability of re-use, dominated by data with a low probability of re-use. Previous Ericsson designs have opted for a software-controlled cache. That approach is not viable when a commercial processor and cache is used. We intend to detect data with high likelihood to be re-used and to lower their risk for replacement using different caching strategies. We will include proposals that would work with today's commercial processors as well as propose alteration to the cache design itself.