VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kansliets noteringar Kod	Dnr
2014-35904-115258-21	2014-5480

2014

Project Grant Junior Researchers

Area of science
Natural and Engineering Sciences
Announced grants
Research grants NT April 9, 2014
Total amount for which applied (kSEK)

2015	2016	2017	2018	2019
1321	1279	1317	1377	1444

APPLICANT

Name(Last name, First name)
Black-Schaffer, David

Date of birth
780405-2632

Gender
Male

Email address
david.black-schaffer@it.uu.se

Academic title
PhD

Position
Lektor (as of May 1st)

Phone
0768242017

Doctoral degree awarded (yyyy-mm-dd)
2008-06-15

WORKING ADDRESS

University/corresponding, Department, Section/Unit, Address, etc.
Uppsala universitet
Institutionen för informationsteknologi
Datorteknik
Box 337
75105 Uppsala, Sweden

ADMINISTRATING ORGANISATION

Administrating Organisation
Uppsala universitet

DESCRIPTIVE DATA

Project title, Swedish (max 200 char)
Proktive hantering av minneshierarkier

Project title, English (max 200 char)
Proactive Memory Hierarchy Management

Abstract (max 1500 char)
The memory hierarchy is critical for a computer´s performance and power efficiency. However, today´s memory hierarchies are reactive and brute-force, repeatedly replicating and moving data, resulting in significant energy and performance losses. Proactively managing the memory hierarchy can dramatically reduce the energy (by moving data to the right place the first time) and improve performance (by more efficiently using the cache space).

This project will investigate proactively managing the memory hierarchy by building upon our breakthrough cache designs [MICRO13, ISCA14], which reduce energy (-60%) and latency (-40%) by eliminating cache tags and providing a metadata table to locate data. We will extend the metadata to enable proactive data management policies for installation and eviction. Uniquely, the underlying design provides a unifying framework to address the full hierarchy and incorporate other designs and advances.

We will use this framework to: 1) investigate proactively moving data to eliminate the wasted energy and space of current designs, 2) explore the tradeoff between associativity and number of cache levels to better match data behavior to storage, and 3) to dynamically adjust the granularity of data, thereby improving sparse data utilization. The unique framework for this project enables all three of these approaches to be undertaken together, making it possible to integrate such disparate optimizations for the first time.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-35904-115258-21

Name of Applicant
Black-Schaffer, David

Date of birth
780405-2632

Abstract language
English

Keywords
memory system, power efficiency, cache design, performance,

Review panel
NT-2

Project also includes other research area

Classification codes (SCB) in order of priority
10206, 20206, 10202

Aspects

Continuation grant
Application concerns: New grant
Registration Number:

Application is also submitted to

similar to: **identical to:**

ANIMAL STUDIES

Animal studies
No animal experiments

ENCLOSED APPENDICES

A, B, C, N, S

APPLIED FUNDING: THIS APPLICATION

Funding period (planned start and end date)
2015-01-01 -- 2019-01-01

Staff/ salaries (kSEK)

Main applicant	% of full time in the project	2015	2016	2017	2018	2019
Black-Schaffer	60	663	680	697	714	732
Other staff						
PhD Student	80	484	513	534	577	626
	Total, salaries (kSEK):	1147	1193	1231	1291	1358

Other project related costs (kSEK)	2015	2016	2017	2018	2019
Research Hardware	75				
Premises	35	35	35	35	35
Travel	51	51	51	51	51
PhD laptop	13				
Total, other costs (kSEK):	174	86	86	86	86

Total amount for which applied (kSEK)

2015	2016	2017	2018	2019
1321	1279	1317	1377	1444

ALL FUNDING

Other VR-projects (granted and applied) by the applicant and co-workers, if applic. (kSEK)

Proj.no.(M) or reg.nr.	Funded 2014	Funded 2015	Applied 2015

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-35904-115258-21

Name of Applicant
Black-Schaffer, David

Date of birth
780405-2632

2012-5332	2000	2000	2000

Project title
Efficient Modeling of Heterogeneity in the Era of Dark Silicon

Applicant
Stefanos Kaxiras

Funds received by the applicant from other funding sources, incl ALF-grant (kSEK)

Funding source	Total	Proj.period	Applied	2015
SSF	10000	2014-2019	2000	

Project title
System-level Intermediate Representation for Program Optimization

Applicant
David Black-Schaffer

Funding source	Total	Proj.period	Applied	2015
FP7	5000	2014-2017	1600	

Project title
Addressing energy in parallel technologies

Applicant
Erik Hagersten

POPULAR SCIENCE DESCRIPTION

Popularscience heading and description (max 4500 char)

For decades we have been constantly increasing the processing power of computers to the point where we can execute billions of calculations per second. However, to do billions of calculations per second, we also need to be able to move billions of pieces of data to and from the processor every second. This data movement is handled by the memory hierarchy, and, unfortunately, it has not kept pace with the processor´s ability to compute. Today´s memory hierarchies operate as reactive brute-force systems that pass data through the entire hierarchy with little regard for where it is needed. As a result, getting data to the processor (via the memory hierarchy) has become the limiting factor in making the most of modern and future computer systems.

This project will develop new techniques to proactively manage the memory hierarchy, and thereby deliver data more intelligently to the processor. By having a more intelligent memory hierarchy, the processor will be able to execute more quickly (since the data will be available in the right place at the right time) and also more efficiently (since less energy will be wasted moving data unnecessarily). In particular, this work will focus on understanding and addressing three issues: 1) how data is being used, and how to move it to the right place at the right time, 2) how we can change how the hierarchy is built to better match the way data is used, and 3) how much of each block of data is used, and how to adjust the data block size to efficiently store it.

To tackle these problems together, this project will leverage recent

breakthrough research results that allow us to design energy-efficient, high-performance cache systems. We will use these designs as a foundation to produce a flexible framework for proactively managing the memory hierarchy. The flexibility in this framework comes from our use of a metadata table to store information about the data in the memory hierarchy. Currently, this metadata allows us to efficiently and quickly locate data within the memory hierarchy, but by extending it, we can implement intelligent policies and efficient methods for moving and accessing data.

To make it all work, a proactive memory hierarchy requires two pieces: a policy to determine what to do and a mechanism to implement it. The policy comes from understanding the behavior of the data in the processor, and, as this behavior changes over time, the policy must be updated as the program runs. The mechanism consists of how the memory hierarchy moves and stores data, and it must be based on fast and efficient designs. One of the strengths of this project is that it allows the policy and the mechanism to be separated via the metadata, while maintaining the ability to have policies that affect behavior across the whole memory hierarchy.

The result of this research will be both a more intelligent memory hierarchy, which will improve computer performance and reduce energy consumption, and a framework for future work in exploring memory hierarchy design. In particular, by separating the policy from the mechanism, this approach will allow us to integrate designs that tackle issues across the memory hierarchy, including existing ideas that have to date only been evaluated in isolation. We believe this flexibility to integrate solutions to multiple problems will be critical to pushing adoption of more intelligence in the memory hierarchy.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-35904-115258-21

Name of applicant
Black-Schaffer, David

Date of birth
780405-2632

Title of research programme
Proactive Memory Hierarchy Management

Appendix A

Research programme

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Proactive Memory Hierarchy Management

Purpose and Aims

Importance: The memory hierarchy is paramount for performance and energy efficiency in today's computer systems. Designers dedicate half of the chip area and core energy for complex, multi-level caches to try and deliver data quickly and efficiently. (See Figure 1.) Software optimization for memory behavior is the first-order concern for both performance and energy, and newer architectures with radical memory system designs are making impressive headway in many domains (e.g., GPUs). On top of this, designs are becoming more complex with deeper hierarchies (e.g., Intel's recent 128MB L4 in-package cache and forthcoming 8GB stacked DRAM for Xeon Phi) and greater pressure for energy efficiency (e.g., Qualcomm's use of L0 caches for mobile devices). Yet despite its importance, the memory hierarchy is managed in the same reactive, brute-force manner as it has been for decades.

Advances: In the past year I have published breakthrough advances in memory hierarchy design in the two most prestigious conferences in computer architecture: ISCA [Sembrant14] and MICRO [Sembrant13], as the main advisor. Our work dramatically improves cache energy efficiency (60% better) by eliminating the need for data tags and improves performance (40% lower latency) by replacing today's brute-force hierarchy search with precise data location information. The key innovation in these designs is the introduction of a cache metadata table that allows us to track information about the data in the cache without the overhead of tags. The metadata table eliminates the need for brute-force searches through tags and across cache levels, thereby improving efficiency and performance. This key change to how memory hierarchies are designed is so disruptive that it opens up new solutions to classic problems.

From Reactive to Proactive: This project will leverage our new designs to develop a framework for proactive memory hierarchy management, and use it to tackle several of the central problems in today's memory hierarchies. The framework will build upon the efficiency of our Tag-Less Cache (TLC) [Sembrant12] and the flexibility of our metadata-based data tracking in our Direct-to-Data (D2D) cache [Sembrant13]. Central to this approach is our metadata table, which enables us to dynamically learn how data is used and apply this information proactively.

Proactively managing data is a dramatic departure from today's reactive memory hierarchies, wherein data is simply pushed and pulled through the hierarchy on demand, wasting both energy and time due to poor placement. For example, a proactive hierarchy can understand data behavior and intelligently move data to the right place, thereby significantly reducing energy and improving performance.

Further, our metadata table uniquely allows us to separate the *mechanism of how* to run the cache from the *policy of what* we should do, by providing a level of indirection between the data and metadata. This separation of concerns provides the flexibility required to address a wide range of issues, and allows us to compose new and existing solutions to different problems in the same design.

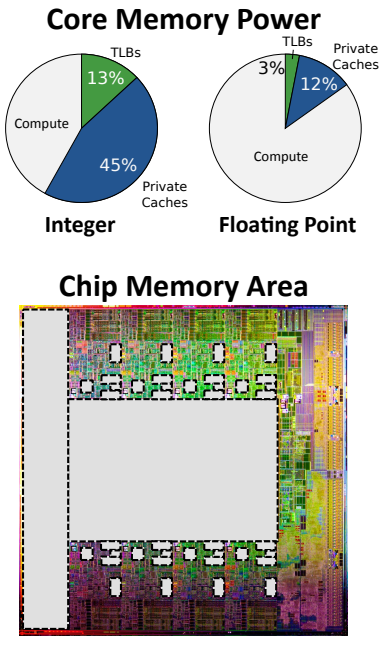


Figure 1: The memory hierarchy cost. *Top: up to 58% of the processor core power is consumed by the memory hierarchy [Sembrant13]. Bottom: 46% of chip area is used for the memory hierarchy [Intel SandyBridge-EP].*

The purpose of this work is to improve memory system performance and energy. **The specific aims** are to leverage our recent advances in cache design and management to develop a flexible framework for proactive memory hierarchy management. This framework will allow us to investigate techniques to improve performance and efficiency. This project has three sub-goals:

1. **Investigate proactive data placement and movement throughout the hierarchy.** Our framework will allow us to explore how we can reduce power and improve performance by minimizing the energy and time spent moving data between cache levels. In particular, we will focus on understanding the dynamic behavior of applications' working sets, how to proactively install and evict data to the appropriate level, and how to track and predict behavior across the hierarchy and over time.
2. **Explore the tradeoff between associativity and hierarchy.** The ability to proactively control where data is placed will allow us to explore trading-off the few levels of highly associative caches we have today for many more levels of lower-associativity caches. The insight is that the size of the cache dictates its latency, and with more cache levels we can have more flexibility in mapping data to the cache with an optimal latency and size. However, this approach will only make sense once we can efficiently move and place data.
3. **Understand the potential of dynamically varying the cache granularity.** In addition to tracking where data is located, the metadata store can also track how it should be handled. One application of this is to seamlessly change the data granularity (block size) throughout the hierarchy and execution of the program. This will allow us to further match the usage and behavior of the data to the metadata and data storage resources available in the hierarchy, thereby improving efficiency and performance.

What is truly unique about this work is that it builds upon the dramatic advances of the TLC and D2D work to create a generic framework that separates the mechanism of how these techniques are implemented from the policies of what to do. This allows us to integrate the best design ideas and build upon the efficiency and performance we have already demonstrated.

Survey of the Field

Memory hierarchies form the foundation of computer systems and have been investigated in great detail. The traditional 2- or 3-level cache hierarchy has been extended to include both larger (100-1000MB DRAM) [Intel Crystalwell, next-generation Intel Xeon Phi] and smaller (4kB and smaller) [Qualcomm Krait] levels. For both of these cases significant work has been done to investigate how to efficiently mange and design them [Jevdjic13, Black-Schaffer08, Kin97]. In addition to adjusting size, many people have investigated changing the cache granularity (e.g., sector and region caches), typically to smaller cache blocks for small caches (to improve utilization) and to larger blocks for larger caches (to reduce overhead) [Kumar12, Zhao13]. Similar approaches have been applied to tracking coherence [Zebchuk07, Zhao13]. On the behavioral side, the difference between spatially reused and temporally reused data has been explored to build separate caches for each type of data [Gonzalez95].

With the universal adoption of multicore processors, a significant body of work has focused on replacement policies that minimize conflicts between cores, primarily for shared cache space, but also for private cache space in the case of inclusive hierarchies where shared data can evict private data [Jaleel10]. Software caching hints have also been investigated to provide developers and compilers the ability to identify accesses that should not be cached or to specify the level within the hierarchy for caching [Sandberg10].

With the looming move from multicore to many-core chips with on-chip networks, there has been a significant amount of work on distributed caches and algorithms for placing and migrating data to minimize the impact of non-uniform access times [see Beckmann13 for a good overview]. The distributed nature of such cache organizations has also been leveraged to achieve the layout and performance benefits of both private and shared caches across cores [Chang06, Intel SandyBridge].

Energy by itself has also been a focus of cache design, with approaches that dynamically change the cache layout [Balasubramonian00], use predictors to reduce the cost of cache accesses [Kaxiras08], and take advantage of the speculative nature of caches to allow lower energy operation [Flautner02].

But despite all of this research there has been very little actual impact. Cache hierarchies today remain simple and brute-force. Software caching hints are only used in very special places (bcopy, malloc, etc.); energy-efficient structures such as L0 caches are rarely used due to latency overheads; multicore-aware replacement policies have only begun to appear 8 years after multicores went mainstream [Intel Haswell]; many-core chips continue to use simple address-based data layouts [Tilera]; and cache hierarchies nearly universally stick to one block size [Intel, AMD, ARM, etc.].

One significant reason for this lack of impact is that these designs are not composable, and since they cannot be put together, it is easy to miss the forest for the trees. For example, much of the benefit found from smaller cache blocks is actually due to the fact that the data should not have been installed in the cache in the first place or it is evicted before it can be reused. (By having smaller cache blocks one sees less pollution.) The work on efficient data movement for DRAM caches and the work to efficiently use small L0 caches are incompatible because neither approach tackles the whole hierarchy. Energy efficient cache designs (until our TLC and D2D work) have continued to use tags and searches. Insights such as the benefit of separating spatial and temporal locality could already be accomplished today, if we could only specify a policy that certain data sets were to be kept in larger (spatial locality) levels of the cache to avoid polluting the smaller (temporal) levels. And throughout these approaches, the policies and mechanisms are typically tightly coupled, thereby making them difficult to integrate with other ideas.

Project Description

The goal of this project is to build upon our Direct-to-Data (D2D) and Tag-less Cache (TLC) work to develop a framework for proactive memory hierarchy management, and exploit this framework to improve energy efficiency and performance. This framework will build upon the energy-efficient baseline design of our TLC work and extend the metadata store from our D2D design to allow more sophisticated management of the memory hierarchy. In particular, it will separate the *policy of what to do* and the *mechanism of how to do it*, thereby making it dramatically easier to combine ideas that tackle problems across the hierarchy.

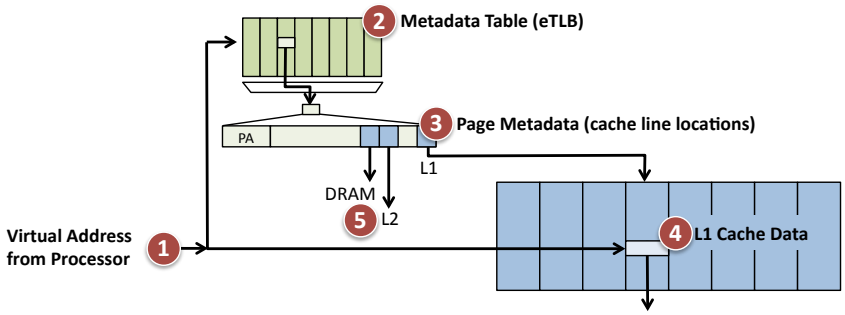


Figure 2: Simplified overview of the TLC and D2D designs. *Metadata is stored in an enhanced TLB (2), which is used to look up per-page metadata (3) for a given data address (1). The metadata indicates where the data is located: for TLC this is the way in the L1 (4) or the location across the whole hierarchy for D2D (5).*

Figure 2 provides a simplified view of the TLC and D2D designs. Specifically, the D2D work provides an enhanced translation lookaside buffer (eTLB) that not only translates from virtual to physical addresses, but also stores cache block metadata on a per-page basis (2). That is, each entry in the meatadata table stores information for all cache lines its page. (Please see the Preliminary Results for more details.) This metadata table allows the design to lookup information about a cache line as part of the translation. In the TLC this capability was used to identify the specific way for the cache block (3), thereby reducing energy over a tag-based search (4). In D2D, this information was used to

accurately identify where in the hierarchy (5) the cache block was located, thereby reducing the latency and energy required to access it.

However, this metadata table can be extended to provide far more generic functionality. For example, it could track where data should be moved on eviction (e.g., bypassing the L2 for streaming data), where data should be installed upon access (e.g., data sets that may not fit in the L1, but still exhibit reuse in the L2), what cache block granularity should be used to store the data (e.g., based on how much of the cache block has not been touched), and information about data sharing (e.g., based on what other cores have accessed the data), to name just a few.

Sub-project 1: Proactive data placement and movement

Data movement in current systems is a reactive brute-force approach wherein the processor requests data and the memory hierarchy searches through each level of the cache until it finds it. And when it does, it copies it down through the whole hierarchy to the processor. This approach is both slow and inefficient. The first sub-project will investigate policies and mechanisms for proactively moving data to the right place in the hierarchy. This will not only reduce the energy wasted moving data around, but it will also reduce the amount of data evicted in the process of moving the data, thereby improving performance.

To proactively move data we need to learn its behavior and be able to use that knowledge in the future. (See Figure 4 for preliminary investigations into data behavior.) The key to this is in using our metadata table to keep track of how data is used, such as where the data was when reused, or how many times it was been used before eviction. The metadata table provides us with a unique ability to learn per-page data behavior at eviction and apply it on demand. Unlike other designs, the indirection of our metadata table allows us to identify what caused data to be installed (e.g., the PC or page causing the original miss) when we see an eviction without a prohibitively high overhead. The value of this is that we can observe the full lifecycle of the data and learn what to do the next time we see that data.

By learning this behavior we can explore policies to appropriately install and evict data. For example, streaming data should typically be evicted directly to DRAM (bypassing the whole cache hierarchy) as it shows little reuse. However, if we detect that the dataset size of the streaming data is small enough to fit in a particular level of the hierarchy (by learning where the data is when it is reused) we can modify this policy to evict it directly to the appropriate level instead. The D2D and TLC work enable this approach by allowing us to efficiently store and update cache metadata and directly access data at different levels of the cache without having to move it through the whole hierarchy.

There are many questions that need to be addressed accomplish this, including: how to identify data behavior accurately and cheaply, whether such identification can be done at a page-granularity as it is in D2D, how to develop a PC-based predictor that can work with page-level data granularity, what policy, placement, and degree of inclusiveness to choose, and how to integrate metadata movement through the hierarchy with data movement.

Sub-project 2: Exploring the tradeoff between associativity and hierarchy

The second sub-project will take advantage of the metadata table's ability to directly locate data to explore tailoring the memory hierarchy more closely to the application behavior. In particular, we will look at trading off today's few cache levels of high associativity for many more levels of lower associativity. For example, we might replace an 8-way L1 with a combination of three smaller, more specialized caches: a 1-way L0, an intermediate 2-way "L0.5", and a smaller 4-way L1. Traditional designs have a high overhead for moving data between levels, making such tradeoff impractical. By building upon D2D work we can access data across multiple levels without this overhead.

The potential here is that we can design each of these caches for a specific latency, such as 1 cycle for the L0, 2 cycles for the "L0.5", and 4 cycles for the L1. If we can proactively put data into the cache that matches its latency requirements, we will see a significant benefit in performance.

Alternatively we could tailor the levels for optimal energy efficiency, or even provide both energy- and latency- optimized hierarchies, with the cache metadata table learning where to place data.

This idea takes the radical approach of significantly increasing the number of levels in the hierarchy, which only makes sense if we both know how to use the levels intelligently and can efficiently locate data within them. To date these requirements have not been fulfilled, leaving the idea of providing a wide range of caches and proactively putting data in the optimal location unexplored.

Sub-project 3: Dynamically varying cache granularity

Intelligent granularity control consists of adjusting the cache block size on a per-page basis to match the behavior of the application data. This can both reduce energy (by moving data in smaller blocks) and improve performance (by increasing effective cache capacity through less wasted space). In the context of the metadata table there is a particularly compelling reason to investigate this potential: the metadata table in the D2D design is over-provisioned such that it can track data across all cache levels. We can use this over-provisioning to track less data, but with a finer granularity, thereby leading to better utilization in the smaller cache levels at little additional cost. The opposite argument holds for larger caches: if their data granularity can be increased, we could track more data, but less flexibly. This sub-project will investigate how to tradeoff capacity pressure in the metadata table (fixed number of data blocks tracked) and the data array (fixed number of data bytes stored) to match the behavior of the dataset and the capacity at each level. This ability to seamlessly track granularity across the hierarchy comes nearly for free with the metadata table's indirection.

Our experience with the TLC suggests that one significant challenge will be in handling sparse data efficiently. For the TLC design we introduced micro-pages in order to efficiently use the metadata store for sparse data. It is likely that similar issues will arise when examining different granularities of cache blocks, but the flexibility of the metadata store will likely enable similarly elegant solutions.

Other Directions

There are numerous other directions to take this work, but at this point it is premature to choose which ones are most promising. A few of these include more intelligent cache and data sharing; flexible handling of hardware transactions, including multiple outstanding transactions, nested transactions, and transactions that cannot be handled today due to their working sets being larger than the private caches; cache compression; design for manufacturing, with hard-coded metadata disabling portions of caches and/or specifying ECC requirements; and integration with the operating system to store metadata across context switches for fast cache warming or across launches for accurate phase detection. The reason there are so many potential directions for this work is that it provides a framework for proactive memory hierarchy management, and by exploring different policies and mechanisms we can address just about any issue.

Significance

The memory hierarchy is arguably the most important part of a processor today. While we can fit as many math and logic units as we want on a processor die, to take advantage of them we need the memory hierarchy to feed them with data. The cost, both in terms of time and energy, for moving data far exceeds the cost of computing the results: a double-precision math operation consumes 20pJ of energy and takes 5 cycles, while reading the data for the operation from an on-chip memory takes 25pJ and 4 cycles. Moving the same data across the chip costs 25x as much energy (500pJ) and takes 8x longer (40 cycles), while loading it from DRAM is 400x as expensive in energy and 40x slower [Dally, SC10 "GPU Computing to Exascale and Beyond", Intel SandyBridge memory system performance]. Further, 30-90% of on-chip traffic consists of moving unused data [Zhao13]. Clearly more intelligent memory hierarchies that can reduce the latency, energy, and frequency of moving data are essential for future processors.

However, to reduce latency and energy we need to consider the whole memory hierarchy and how changes in one place affect those in others. This implies that we cannot examine one aspect of the hierarchy in isolation (for example, cache block size, or placement policies) but must find a framework to compose different solutions together to see the overall effect. For example, to understand how to move data intelligently it is not enough to look at a standard 3-level hierarchy: we need a solution that can adapt to cover both smaller (L0) and larger (L4) caches in the future, appropriately adapt to private and shared data and storage, and incorporate novel features such as adjustable cache block sizes and compression.

To understand such a complex design space, we need to be able to explore intelligent policies that adapt to the runtime behavior of an application in the context of the full system. For example, the correct policy for data movement may change depending on whether the data is shared and/or if the hierarchy has a L0 cache with which to bypass the L1 cache. This requires that we be able to compose solutions for multiple issues (e.g., data movement, sharing, cache bypass).

The foundation of our approach is our metadata table, which allows us to store information about data behavior and tie it back to the event that accessed that data. By combining different metadata we can develop policies that look at many aspects of the data's behavior to make the right decision. Of course, to be competitive, we need to also build on a design that offers the best energy efficiency and latency. In short, to put all of these aspects together we need a framework that provides a solid foundation to explore different mechanisms and policies. The combination of our recent TLC and D2D work provides a foundation for this framework by giving us a leading-edge efficient cache design and a rich metadata framework.

Preliminary Results

The problem with memory systems as they are implemented today is that the employ reactive, brute-force approaches for moving and finding data, which leads to significant inefficiencies. The most obvious of these are the universal practices of storing data tags with each cache block and using a level-by-level search to find data across the hierarchy. While these approaches work, they incur significant overheads in terms of energy (constantly reading multiple tags and levels) and latency (having to look in each lower level before finding the data in higher levels). We have addressed these two initial issues in our preliminary TLC and D2D designs.

TLC: Tag-Less Caches for Reducing Dynamic First-Level Cache Energy

Traditional cache designs contain both data arrays to store the data and tag arrays to identify each block in the data array. This approach allows arbitrary data to be stored in the data array as each block has its own tag to identify it. However, caches typically read out all possible matching tags on every access and first-level caches generally read out all data ways in parallel to reduce access latency (e.g., rather than waiting to see which way is correct based on the tags). Since at most one of these ways is used, this results in a waste of nearly 7/8 of the read energy for an 8-way cache.

Our Tag-Less Cache design [Sembrant13], published in MICRO 2013, addresses these issues by extending the translation lookaside buffer (TLB) to include metadata about the location of data along with its traditional virtual-to-physical address translation. This information is stored on a per-page basis, and includes whether each cache line for the page is installed in the cache as well as which way it is located in. As a result, a single lookup in this eTLB now tells the cache where the data is. However, the TLC design goes one step further and allows the designer to flexibly bank the eTLB's SRAM arrays. This means we can afford to first access the eTLB to determine the way, and then directly read only the correct way from the data array. As a result we can reduce the dynamic energy of cache lookups by 60% compared to a standard VIPT cache without hurting performance. (See Figure 3.)

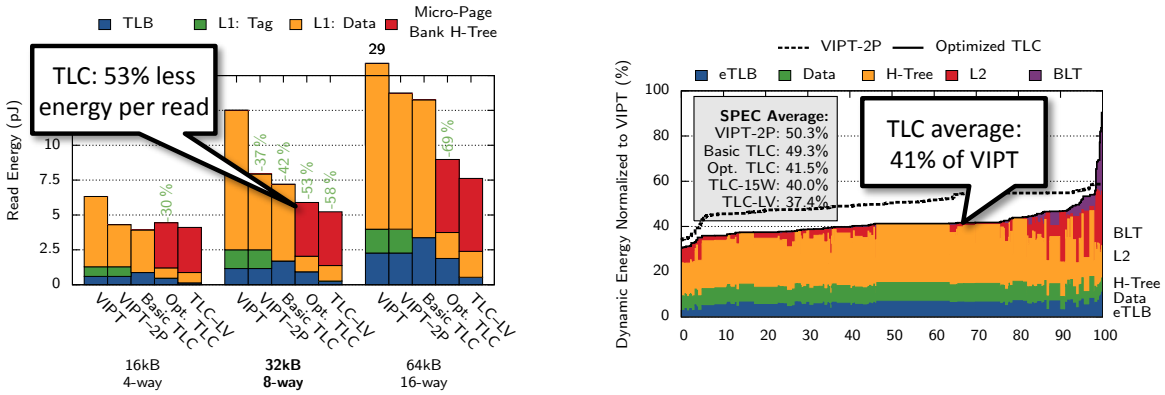


Figure 3: TLC energy improvements. *Per-read energy breakdown (left) and dynamic energy for SPEC (right) of an optimized TLC design ("Opt. TLC") vs. a VIPT and 2-phase VIPT design. The TLC design consumes 41% of the dynamic energy of the baseline VIPT design when running the SPEC benchmarks. From [MICRO13].*

Unfortunately, tracking cache data at a page-level granularity leads to some tradeoffs. In particular, evicting a page from the eTLB implies that its data must to be removed from the cache (as there was no other way to track it) and applications with sparse data access patterns experienced reduced cache capacity due to the number of eTLB entries. To address these problems we developed a technique to dynamically adjusting the page size (micro pages).

D2D: The Direct-to-Data Cache: Navigating the Cache Hierarchy with a Single Lookup

The TLC design extended the TLB to contain metadata about the location of cache lines for each memory page. In our Direct-to-Data design [Sembrant14], to appear in ISCA 2014, we built upon this work to make the metadata even more useful by enabling it to point to data in arbitrary levels of the cache hierarchy. This extension allows us to eliminate the need to search the hierarchy for data that is not in the L1. By tracking cache line location information in the metadata we can correctly send the vast majority (81-99%) of accesses directly to the right level in the hierarchy. Further, this single metadata lookup reduced the access latency to the L2 by 40% by skipping the L1 and L2 tag lookups.

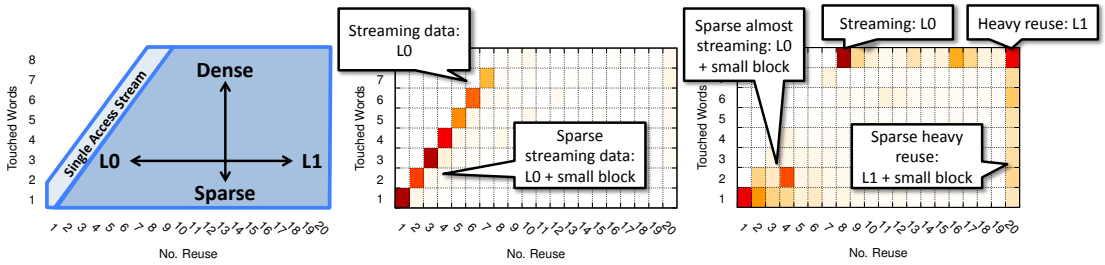
The latency reduction from D2D directly improves the performance of L2-sensitive applications by 5-14%, but does not have a significant effect on applications that fit comfortably in the L1 cache. However, the reduced L2 latency can be traded-off for other design parameters. For example, we demonstrated how one could reduce the reorder buffer size by 25% for L2-sensitivite applications without hurting performance. Reducing the size of complex out-of-order structures has significant implications for the energy efficiency of the processor core itself. Alternatively, one could trade-off the lower L2 latency for a smaller L1 cache without hurting performance, and reduce energy further.

On-going work: application behavior classification

Even with the metadata to tell the cache where the data was located, the D2D design still employed a simple policy of always moving data into the L1 upon access. This incurs costs in both energy (from reading and writing the data) as well as performance (if the moved data evicts other valuable data). In this project we will improve upon this by more intelligently handling data movement and placement. For example, by bypassing the L1 for streaming data that shows no reuse; or only keeping data in the L2 if it has some reuse, but a data set size that is too large for the L1 and would otherwise pollute it; or using reduced cache block sizes for sparsely accessed cache blocks.

To understand the potential for such policies we have undertaken preliminary investigations into how applications reuse data. (See Figure 4.) This investigation counts how many of a cache block's data words are accessed before it is evicted from the L1 (Y-axis "Touched words") and how many times it is reused in the L1 before eviction (X-axis "No. Reuse"). By looking at this data we can graphically understand both whether there is a potential for optimizing placement and how to characterize application behavior at runtime.

A schematic representation of these different behaviors is shown in Figure 4a. Cache blocks where words are only touched once before eviction ("Single Access Streams") are shown on the diagonal left side of the diagram. These types of data accesses should bypass the L1 to avoid pollution (e.g., placed only in an L0 or line buffer). The density of the data within the cache blocks varies from the top to the bottom, with smaller cache block sizes preferred for data that is sparser. The degree of reuse indicates whether the data should bypass the L1 (left, with streaming being the extreme case) or be placed in the L1 for maximum reuse (right).



(a) Schematic (b) gcc at 32G instructions (c) gcc at 12G instructions

Figure 4: Understanding application data behavior. *(a): a schematic representation of how policies should handle data with different sparsity (y-axis) and reuse intensity (x-axis). (b) and (c): examples of widely varying behavior within the same application (gcc) and the policies that should be applied to each data set.*

Our analysis of the data behavior for two significantly different phases of the gcc benchmark is shown in Figures 4b and 4c. The figures are annotated with the policies that should be applied to the different data sets based on their degree of sparsity and reuse. These initial evaluations demonstrate that we should be able to develop polices to proactively move and place data.

National and International Collaboration

I have developed strong research collaborations within Uppsala, including joint publications and co-advised PhD students. These include: performance modeling (Prof. Erik Hagersten [Eklov11b, Eklov13]), energy-efficient runtimes (Prof. Stefanos Kaxiras [Koukos13]), high-performance computing (Prof. Elisabeth Larsson [Ljungkvist11]), and compilers for power efficiency (Dr. Alexandra Jimborean [Jimborean14]). All of these collaborations have resulted in publications. As part of the Uppsala Programming for Multicore Architectures Research Center I routinely interact with colleagues who work on scientific computing, formal verification, and programming languages.

Nationally, I have given invited talks at Chalmers (Prof. Per Stenström), KTH (Prof. Mats Brorsson), and the Swedish Institute for Computer Science. Our D2D and TLC work has generated interest in collaborating with Per Stenström at Chalmers on using our metadata to support efficient hardware transactions and cache compression. I have worked with Ericsson to implement our research in shared resource analysis on their mobile base stations and have had in-depth discussions with SAAB defense systems, ABB, Huawei, and DICE. I have an early-stage startup with Prof. Sverker Janson from SICS and Sameh El-Ansary from Nile University in Egypt, with over 1800 users.

Internationally I have submitted EU proposals with researchers and companies from the UK, Italy, France, Sweden, Belgium, and Switzerland. I am a co-author of the 2011/2012 [Duranton11] and 2013/2014 [Duranton13] roadmaps for the European Network of Excellence on High Performance and Embedded Architecture and Compilation (HiPEAC). As a co-author I was responsible for talking to representatives from a broad range of ICT fields in both academia and industry and synthesizing a set of goals and priorities to encourage the European Commission to allocate funding for ICT. This activity exposed me to a broad range of players across Europe and led to my being part of the ADEPT (Addressing Energy in Parallel Technologies) FP7 project. As part of ADEPT, I work with researchers in Prof. Lieven Eekhout's group in Ghent University to develop fast architecture and modeling technology to predict performance across a range of systems. I currently have one Ph.D. student

funded by ADEPT and am co-advising another who has recently merged hardware virtualization with cycle-accurate simulation to provide a 1000x improvement in simulation speed [Sandberg14].

Other Grants

I was awarded a 2013 Future Research Leaders Grant by the Swedish Foundation for Strategic Research (SSF). This grant provides 10M SEK (1.1M EUR) funding to each of the 20 most promising young science and technology researchers in Sweden. I was the only recipient in information technology. This grant will allow me to explore how our shared resources management tools [Eklov11b, Eklov13, Sandberg13] can be used to optimize heterogeneous execution. This work will further build on my experience developing the OpenCL standard for heterogeneous computing when I worked at Apple.

I am a Co-PI on the 2012 VR framework grant "Efficient Modeling of Heterogeneity in the Era of Dark Silicon" 10M SEK (1.1M EUR). This project seeks to extend our shared resource modeling [Eklov11a, Eklov11b, Sandberg13, etc.] to heterogeneous systems. While such shared resource modeling is valuable for understanding runtime behavior, it does not address hardware design.

I am a Co-PI on the 2012 FP7 ADEPT project 5M SEK. This project seeks to extend our shared resource modeling technologies to predict power and performance on a range of systems.

Independent Line of Research

Since graduating from Stanford University in 2008 I have published more than 20 articles without my PhD advisor. My recent HPCA [Sandberg13], MICRO [Sembrant13], and ISCA [Sembrant14] publications are in the premiere international venues in the field of computer architecture, with acceptance rates of 12-20%. I am the main advisor and last author for the MICRO and ISCA publications. I have received two best paper awards [Eklov11a, Eklov11b] for our work on understanding and modeling the performance effects of shared memory systems and two HiPEAC Paper Awards [Sandberg13, Sembrant13].

Since returning to academia from industry in 2010 I have served on over ten program committees, including CF 2014, ISC 2014, IPDPS 2014, and HPCA 2014. I have been on the grading committee for three Ph.D. theses (Charles University Prague, 2012; University of Gent, 2013, Linköping University, 2014) and two Licentiate theses (KTH, 2013; Chalmers, 2013). I am the main advisor for 4 Ph.D. students, co-advisor for 2 others, and am actively recruiting 3 new students for my Future Research Leaders grant.

I have given keynotes at the Swedish Multicore Day 2013 and the Swedish Multicore Conference 2011, presented our research at the 2011 Ericsson Software Research Day, have given invited talks at Charles University in Prague, Gent University in Belgium, KTH, the Swedish Institute of Computer Science, Chalmers, Linköping, and MDH, and taught GPU computing at the KTH high-performance computing summer school four years in a row.

Type of Employment

This project will cover 80% of a PhD student and 60% of my research time as an assistant professor. This will bring my total research time to 80%, including my Future Research Leaders grant.

References

[Balasubramonian00] R. Balasubramonian, D. Albonesi, A. Buyuktosunoglu, and S. Dwarkadas. “Memory hierarchy reconfiguration for energy and performance in general-purpose processor architectures.” International Symposium on Micorarchitecture (MICRO), 2000.

[Beckmann13] N. Beckmann and D. Sanchez. “Jigsaw: Scalable Software-Defined Caches.” International Conference on Parallel Architectures and Compilation Techniques (PACT), 2013.

[Black-Schaffer08] **D. Black-Schaffer**, J. Balfour, V. Parikh, J. Park, and W. J. Dally. “Hierarchical Instruction Register Organization.” Computer Architecture Letters (CAL), 2008.

[Chang06] J. Chang and G. S. Sohi. “Cooperative Caching for Chip Multiprocessors.” International Symposium on Computer Architecture (ISCA), 2006.

[Duranton11] M. Duranton, **D. Black-Schaffer**, S. Yehia, and K. De Bosschere. “Computing Systems: Research Challenges Ahead. The HiPEAC Vision 2011/2012.” High Performance and Embedded Architecture and Compilation Network of Excellence (HiPEAC), 2011.

[Duranton13] M. Duranton, **D. Black-Schaffer**, K. De Bosschere, and J. Maebe. “The HiPEAC Vision for Advanced Computing in Horizon 2020.” High Performance and Embedded Architecture and Compilation Network of Excellence (HiPEAC), 2013.

[Eklov11a] D. Eklov, **D. Black-Schaffer**, and E. Hagersten. “Fast Modeling of Cache Contention in Multicore Systems.” Conference on High Performance and Embedded Architectures and Compilation (HiPEAC), 2011. *(best paper)*

[Eklov11b] D. Eklov, **D. Black-Schaffer**, and E. Hagersten. “Cache Pirating: Measuring the Curse of the Shared Cache.” International Conference on Parallel Processing (ICPP), 2011. *(best paper)*

[Eklov13] D. Eklov, N. Nikoleris, **D. Black-Schaffer**, and E. Hagersten. “Bandwidth Bandit: Quantitative Characterization of Memory Contention.” International Symposium on Code Generation and Optimization (CGO), 2013.

[Flautner02] K. Flautner, N. Kim, S. Martin, D. Blaauw, and T. Mudge. “Drowsy Caches: Simple Techniques for Reducing Leakage Power.” International Symposium on Computer Architecture (ISCA), 2002.

[Gonzalez95] A. Gonzalez, C. Aliagas, and M. Valero. “A Data Cache with Multiple Caching Strategies Tuned to Different Types of Locality.” International Conference on Supercomputing (ICS), 1995.

[Jaleel10] A. Jaleel, K. Theobald, S. Steely, and J. Emer. “High Performance Cache Replacement Using Re-Reference Interval Prediction (RRIP).” International Symposium on Computer Architecture (ISCA), 2014.

[Jevdjic13] D. Jevdjic, S. Volos, and B. Falsafi. “Die-stacked DRAM caches for servers: hit ratio, latency, or bandwidth? Have it all with the footprint cache.” International Symposium on Computer Architecture (ISCA), 2013.

[Jimborean14] A. Jimborean, K. Koukos, V. Spiliopoulos, **D. Black-Schaffer**, and S. Kaxiras. “Fix the code. Don’t tweak the hardware: A new compiler approach to Voltage-Frequency scaling.” International Conference on Code Generation and Optimization (CGO), 2014.

[Kaxiras08] S. Kaxiras and M. Martonosi. “Computer Architecture Techniques for Power-Efficiency.” 2008

[Kin97] J. Kin, M. Gupta, and W. H. Mangione-Smith. “The Filter Cache: an energy efficient memory structure.” International Symposium on Microarchitecture (MICRO), 1997.

[Koukos13] K. Koukos, **D. Black-Schaffer**, V. Spiliopoulos, and S. Kaxiras. “Towards more efficient execution: A decoupled access-execute approach.” International Conference on Supercomputing (ICS), 2013.

[Kumar12] S. Kumar, H. Zhao, A. Shriraman, E. Matthews, S. Dwarkadas, and L. Shannon. “Amoeba-cache: Adaptive blocks for eliminating waste in the memory hierarchy.” International Symposium on Microarchitecture (MICRO), 2012.

[Ljungkvist11] K. Ljungvist, M. Tillenius, **D. Black-Schafer**, S. Holmgren, M. Karlsson, and E. Larsson. “Using Hardware Transactional Memory for High-Performance Computing.” International Symposium on Parallel & Distributed Processing (IPDPS), workshop, 2011.

[Sandberg10] A. Sandberg, D. Eklov, and E. Hagersten. “Reducing Cache Pollution Through detection and Elimination of Non-Temporal Memory Accesses.” International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2010.

[Sandberg13] A. Sandberg, A. Sembrant, E. Hagersten, and **D. Black-Schaffer**. “Modeling Performance Variation Due to Cache Sharing.” International Symposium on High Performance Computer Architecture (HPCA), 2013.

[Sandberg14] A. Sandberg, E. Hagersten, and **D. Black-Schaffer**. “Full Speed Ahead: Detailed Architectural Simulation at Near-Native Speed.” Technical Report. Uppsala University, Department of Information Technology, nr. 2014-005, 2014.

[Sembrant13] A. Sembrant, E. Hagersten, and **D. Black-Schaffer**. “TLC: A Tag-less Cache Design for Reducing Dynamic First Level Cache Energy.” International Conference on Microarchitecture (MICRO), 2013.

[Sembrant14] A. Sembrant, E. Hagersten, and **D. Black-Schaffer**. “The Direct-to-Data (D2D) Cache: Navigating the Cache Hierarchy with a Single Lookup.” International Symposium on Computer Architectures (ISCA), 2014. *(to appear)*

[Zebchuk07] J. Zebchuk, E. Safi, and A. Moshovos. “A Framework for Coarse- Grain Optimizations in the On-Chip Memory Hierarchy,” in International Symposium on Microarchitecture (MICRO), 2007.

[Zhao13] H. Zaho, A. Shriraman, S. Kumar, and S. Dwarkadas. “Protozoa: adaptive granularity cache coherence.” International Symposium on Computer Architecture (ISCA), 2013.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-35904-115258-21

Name of applicant
Black-Schaffer, David

Date of birth
780405-2632

Title of research programme
Proactive Memory Hierarchy Management

Appendix B

Curriculum vitae

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Appendix B – CV/Scientific Qualifications

Higher Education Qualifications

- B.A. in Engineering, 2000, **Dartmouth College**, USA. Electrical engineering.
- M.S. in Electrical Engineering, 2003, **Stanford University**, USA. Robotics and vision.

Doctoral Degree

Stanford University, USA, 6/2008, Electrical Engineering/Computer Architecture
Thesis: "Block Parallel Programming for Real-time Applications on Multi-core Processors"
Advisor: William J. Dally (Stanford), Co-advisors: Mark Horowitz (Stanford), Anwar Ghuloum (Intel)

Postdoctoral Position

Uppsala University, Department of Information Technology, 9/2009-5/2010
Project: CoDeR-MP: understanding the impact of multi-core technology on real-time systems.

Docent Level

Docent in Computer Science, Uppsala University, expected 6/2014.

Present Position

Associate Professor (lektor), Uppsala University, Department of Information Technology 5/2014-present.

Previous Positions

- **Assistant Professor** (biträdande lektor), Uppsala University, Department of Information Technology. 5/2010-5/2014. (70% research)
- **Postdoc**, Uppsala University, Department of Information Technology. 9/2009-5/2010.
- **Software Engineer** on the OpenCL team at Apple, Inc. Responsible for designing, developing, implementing, and testing the first implementation of the OpenCL standard for heterogeneous parallel computation on CPUs and GPUs. 6/2008-7/2009.

Supervision (with expected PhD date and funding source)

Completed:

- David Eklöv (co-advisor, PhD 12/2012, SSF CoDeR-MP)

On-going:

- **Andreas Sembrant** (main advisor, Lic 12/2012, PhD expected 12/2015, SSF CoDeR-MP)
- **Germán Ceballos** (main advisor, PhD expected 5/2018, VR UPMARC)
- **Moncef Mechri** (main advisor, PhD expected 9/2018, FP7 ADEPT)
- **Ricardo Alves** (main advisor, PhD expected 9/2018, VR Framework Grant)
- Andreas Sandberg (co-advisor, PhD expected 6/2014, SSF CoDeR-MP)
- Muneeb Khan (co-advisor, PhD expected 9/2014, VR UPMARC)

Deductible Time

Parental Leave	2008-05-01	2008-05-26	100%	0.8 month
Parental Leave	2009-08-22	2009-09-24	100%	1 month
Parental Leave	2011-09-01	2011-11-30	20%	0.6 months
Parental Leave	2011-12-01	2011-12-31	30%	0.3 months
Parental Leave	2012-01-01	2012-06-30	80%	4.8 months
Parental Leave	2012-08-01	2012-08-31	60%	0.6 months
Parental Leave	*2008-06-01*	*2012-09-01*	*Total*	*8.1 months*

Other

PhD Grading Committees:

- J. Wang, Linköping University, Sweden, 5/2014
- V. Babka, Charles University, Prague, 9/2012
- K. Van Craeynest, University of Ghent, Belgium, 4/2013
- A. Pobdobas, KTH, Sweden, 5/2013 (Licentiate)
- A. Bardizbanyan, Chalmers University, Sweden, 10/2013 (Licentiate)

Invited Talks: HiPEAC ADEPT Workshop (2014); Keynote, Swedish Association for Distance Education (2013); Keynote, Swedish Institute of Computer Science Multicore Day (2013); University of Ghent, Dept. of Electronics and Information Systems (2013); Charles University, Prague, Dept. of Distributed and Dependable Systems (2012); Keynote, Swedish Workshop on Multicore Computing (2011); Ericsson Software Research Day (2011); KTH (2011); Chalmers (2010)

Program Committees: International Symposium on High-Performance Computer Architecture (HPCA) 2015; GPGPU 2014; International Conference on Supercomputing extended review committee (ICS-2014); Computing Frontiers (2014); Programming Issues for Multi-Core Computers (MULTIPROG-2014), Cluster Journal special issue on Unconventional Cluster Architectures and Applications (2013); International Conference on Advanced Parallel Processing Technology (APPT 2013); International Conference on Information Communication Technology (ICT-EurAsia 2013); 6th Workshop on Programming Issues for Heterogeneous Multicores (MULTIPROG-2013); 26th IEEE International Parallel & Distributed Programming Symposium (IPDPS 2011); 7th Annual Workshop on Modeling, Benchmarking and Simulation (MoBS 2011)

Best Paper Awards:

- David Eklöv, Nikos Nikoleris, David Black-Schaffer, and Erik Hagersten. "Cache Pirating: Measuring The Curse of the Shared Cache." International Conference on Parallel Processing (ICPP), September 2011
- David Eklöv, Erik Hagersten, and David Black-Schaffer. "Fast Modeling of Shared Caches in Multicore Systems." International Conference on High-Performance and Embedded Architectures and Compilers (HiPEAC), January 2011

Grants (awarded):

- PI: Swedish Foundation for Strategic Research, **Future Research Leaders** 5. *System-level Intermediate Representation for Program Optimization*. **10M SEK**, 2013-2018 (only recipient in ICT across Sweden)
- Co-PI: **VR Framework Project** (2012-5332). *Efficient Modeling of Heterogeneity in the Era of Dark Silicon.* **10M SEK**, 2012-2017
- Co-PI: **FP7 "Addressing Energy in Parallel Technologies." 5M SEK**, 2013-2016.
- PI: Uppsala science and technology fund for pedagogical renewal (flipped classroom), 70k SEK, 2012
- PI: Uppsala university pedagogical development (flipped classroom teaching), 99kSEK, 2013.

Industrial and Academic Networks:

- European Network of Excellence on High Performance and Embedded Architecture and Compilation (HiPEAC): co-author of 2011/2012 and 2013/2014 roadmaps.
- Swedish representative to the Open European Network for High Performance Computing on Complex Environments. Co-organized 2013 summer school.
- Ericsson: MSc project supervision to implement our sensitivity technology on their next-generation base station hardware (2012-2014), invited presentation at Ericsson Software Research Day 2011.

Teaching (invited courses and awards):

- Uppsala Engineering and Science Student Union, Pedagogical Prize, for integrating online and in-class teaching, 2012.
- KTH PDC Center for High Performance Computing, Summer School, Invited Lecturer, 2011-2014.

Publications after returning to academia from industry in 9/2009: 21 (6/year, after deducting parental leave)

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-35904-115258-21

Name of applicant
Black-Schaffer, David

Date of birth
780405-2632

Title of research programme
Proactive Memory Hierarchy Management

Appendix C

List of publications

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Appendix C – List of Publications

Peer-reviewed articles

William Dally, James Balfour, **David Black-Schaffer**, James Chen, R. Curtis Harting, Vishal Parikh, JongSoo Park, and David Sheffield. "Efficient Embedded Computing." IEEE Computer, July 2008.

*** David Black-Schaffer**, James Balfour, William Dally, Vishal Parikh, and JongSoo Park. "Hierarchical Instruction Register Organization." Computer Architecture Letters, vol. 7, 2008.

James Balfour, William Dally, **David Black-Schaffer**, Vishal Parikh, and JongSoo Park. "An Energy-Efficient Processor Architecture for Embedded Systems." Computer Architecture Letters, vol. 7, 2008.

Peer-reviewed conference contributions

2014

***** Andreas Sembrant, Erik Hagersten, and **David Black-Schaffer**. "The Direct-to-Data Cache: Navigating the Cache Hierarchy with a Single Lookup." International Symposium on Computer Architecture (**ISCA**), 2014. *To appear*

Alexandra Jimborean, Konstantinos Koukos, Vasileios Spiliopouls, **David Black-Schaffer**, and Stefanos Kaxiras. "Fix the code. Don't tweak the hardware: A new compiler approach to Voltage-Frequency scaling." International Conference on Code Generation and Optimization (**CGO**), 2014.

2013

***** Andreas Sembrant, Erik Hagersten, and **David Black-Schaffer**. "TLC: A Tag-less Cache Design for Reducing Dynamic First Level Cache Energy." International Symposium on Microarchitecture (**MICRO**), 2013.

Konstantinos Koukos, **David Black-Schaffer**, Vasileios Spiliopoulos, and Stefanos Kaxiras. "Towards more efficient execution: A decoupled access-execute approach." International Conference on Supercomputing (**ICS**), June 2013.

***** Andreas Sandberg, Andreas Sembrant, Erik Hagersten, and **David Black-Schaffer**. "Modeling Performance Variation Due to Cache Sharing." 19th International Symposium on High Performance Computer Architecture (**HPCA**), January 2013.

Konstantinos Koukos, **David Black-Schaffer**, Vasileios Spiliopoulos, and Stefanos Kaxiras. "Towards Power Efficiency on Task-Based, Decoupled Access-Execute Models." Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures (**PARMA**), January 2013.

David Eklöv, **David Black-Schaffer**, and Erik Hagersten. "Bandwidth Bandit: Quantitative Characterization of Memory Contention." International Symposium on Code Generation and Optimization (**CGO**), January 2013.

Germán Ceballos and **David Black-Schaffer**. "Shared Resource Sensitivity in Task-Based Runtime Systems." 6th Swedish Multicore Workshop, 2013.

2012

Andreas Sembrant, **David Black-Schaffer** and Erik Hagersten. "Phase Behavior in Serial and Parallel Applications." IEEE International Symposium on Workload Characterization (**IISWC**), 2012

Andreas Sandberg, **David Black-Schaffer**, and Erik Hagersten. "Efficient Techniques for Predicting Cache Sharing and Throughput." The 21st International Conference on Parallel Architectures and Compilation Techniques (**PACT**), 2012.

***** Andreas Sembrant, **David Black-Schaffer** and Erik Hagersten. "Phase Guided Profiling for Fast Cache Modeling." International Symposium on Code Generation and Optimization (**CGO**), April 2012.

2011

Karl Ljungkvist, Martin Tillenius, **David Black-Schaffer**, Sverker Holmgren, Martin Karlsson, Elisabeth Larsson. "Using Hardware Transactional Memory for High-Performance Computing." International Symposium on Parallel & Distributed Processing, Workshops and PhD Forum (**IPDPSW**). 2011.

David Eklov, Nikos Nikoleris, **David Black-Schaffer** and Erik Hagersten. "Cache Pirating: Measuring the Curse of the Shared Cache." In Proceeding of the 40th International Conference on Parallel Processing (**ICPP**), September 2011.
Best paper award.

David Eklov, **David Black-Schaffer** and Erik Hagersten. "Fast Modeling of Shared Caches in Multicore Systems." In Proceedings of the 6th International Conference on High Performance and Embedded Architecture and Compilation (**HiPEAC**), January 2011.
Best paper award.

Marcus Holm, Martin Tillenius, and **David Black-Schaffer**. "A simple model for tuning tasks." In Proceedings of the Swedish Workshop on Multi-core Computing, 2011.

Andreas Sandberg, **David Black-Schaffer**, and Erik Hagersten. "A simple statistical cache sharing model for multicore." In Proceedings of the Swedish Workshop on Multi-core Computing, 2011.

2010

David Black-Schaffer and William J. Dally. "Block-Parallel Programming for Real-Time Embedded Applications." In Proceedings of the 2010 39th International Conference on Parallel Processing (**ICPP**), September 2010.

David Eklov, **David Black-Schaffer** and Erik Hagersten. "StatCC: A Statistical Cache Contention Model." In the Proceedings of 19th International Conference on Parallel Architectures and Compilation Techniques (**PACT**), September 2010.

Earlier

JongSoo Park, Sung-Boem Park, James D. Balfour, **David Black-Schaffer**, Christos Kozyrakis and William J. Dally. "Register Pointer Architecture for Efficient Embedded Processors." In Proceedings of the Conference on Design Automation and Test in Europe (**DATE**), April 2007.

Kristof Richmond, **David Black-Schaffer**, and Stephen Rock. "Automatic Determination of Vision Lock on the Seafloor in the Presence of Dust." In Proceedings of the Unmanned Untethered Submersible Technology Conference (**UUST**), Aug 2003.

Review articles, book chapters, books

Erik Hagersten, David Eklöv, and **David Black-Schaffer**. "Efficient Cache Modeling with Sparse Data." Chapter in "Processor and System-on-Chip Simulation", editors Olivier Temam and Rainer Leupes. Springer, 2010.

Patents

Provisional patent for Tag-less Cache and Direct-to-Data designs, submitted 2013.

Open access computer programs developed

Cache Pirate and Bandwidth Bandit tools (developed by thesis and PhD students).

Popular-scientific articles and/or presentations

Marc Duranton, **David Black-Schaffer**, Koen De Bosschere, and Jonas Maebe. "The HiPEAC Vision for Advanced Computing in Horizon 2020." High Performance and Embedded Architecture and Compilation Network of Excellence (**HiPEAC**). March 2013.

David Black-Schaffer. "GPUs: The Hype, The Reality, and the Future." Keynote at the Swedish Institute for Computer Science Multicore Day, 2013.

David Black-Schaffer. "Flipped classroom teaching in an introductory IT course." Seminar keynote in "Överlever högskolan globala nätkurser?" (Will universities survive global online education?), Royal Institute of Technology, Stockholm, 2013

David Black-Schaffer. "Resource Sharing in Multicore Processors." Presentation at Ericsson Software Research Day. 2011.

Marc Duranton, **David Black-Schaffer**, Sami Yehia, Koen De Bosschere. "Computing Systems: Research Challenges Ahead. The HiPEAC Vision 2011/2012." High Performance and Embedded Architecture and Compilation Network of Excellence (**HiPEAC**). November 2011.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-35904-115258-21

Name of applicant
Black-Schaffer, David

Date of birth
780405-2632

Title of research programme

Appendix N

Budget and research resources

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Appendix N – Budget

Requested VR Funding

"The indirect costs of the Department of Information Technology are for 2014 calculated as a flat rate of 28,5 % on direct running costs (excl. depreciation of equipment and premises)."

This proposal requests funding for one PhD student (80% for five years) and for 60% of my time, bringing my total research time to 80% with the additional 20% from my Future Research Leaders grant. This represents approximately 80% of the funding for this project, as the current PhD student will continue working in this area for the remaining two years of his degree.

In addition to these staff costs, the proposal requests 75k SEK for computer hardware for experiments, costs for office space (25k SEK/year for a shared PhD office and 10k SEK/year towards a faculty office), travel (15k SEK/year for the PhD student and 25k SEK/year for myself) and 10k SEK for initial computer equipment for the PhD student.

All budget values in kSEK, including overhead and employee taxes where appropriate:

	Y1 (2014)	Y2 (2015)	Y3 (2016)	Y4 (2017)	Y5 (2018)
Salaries					
Ph.D. Student (VR: 80%)	484	513	534	577	626
Dr. Black-Schaffer (VR: 60% 2015-2018)	663	680	697	714	732
Other costs					
Research hardware (no overhead)	75				
Premises	35	35	35	35	35
Travel (15k PhD, 25k Black-Schaffer)	51	51	51	51	51
PhD laptop/setup	13				
VR Costs	**1321**	**1279**	**1317**	**1378**	**1444**

Total VR Funding: 6 739 kSEK

Other Funding

Type of grant	Applied or Granted	Funding source	Grant holder/project leader	Grant period	Total amount in thousands
Project Grant Future Research Leader	Granted	SSF	David Black-Schaffer	2013-2018	10,000
Framework Grant Modeling of Heterogeneous Systems	Granted	VR	Stefanos Kaxiras	2012-2017	10,000
Project Grant ADEPT	Granted	EU	Erik Hagersten	2012-2015	5,000

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-35904-115258-21

Dnr
2014 -

Name of applicant
Black-Schaffer, David

Date of birth
780405-2632

Reg date
2014-04-08 22:19:33

Project title
Proactive Memory Hierarchy Management

NT

Research grants NT April 9, 2014
Project Grant Junior Researchers
*Övrigt generellt
Uppsala universitet

NT-2

Appendix S

Signatures

2015	2016	2017	2018	2019	
1321	1279	1317	1377	1444	2015-01-01 -- 2019-01-01

Applicant | **Date**

Head of department at host University | **Clarifi cation of signature** | **Telephone**

A signature on the application is required not only from the applicant but also from the authorised representative of the administrating organisation/institution or equivalent (normally the head of the department or establishment where the research is to be conducted). The signature confirms that the department can accommodate the proposed research, position or equipment; that the costing in the application is approved for the department's part, that any proposed experimentation on human or animal subjects has been reported, and that the applicant has reported any secondary occupations and commercial ties (s)he may have, and nothing inconsistent with good research practice has thereby emerged. The applicant must have discussed these conditions with the representative of the administrating organisation/institution or equivalent before the latter approves and signs the application.

NOTE the exceptions!
Please note that a different procedure applies for the following types of grants:

- Infrastructure- Operation Grants
- Infrastructure- Grants for Expensive Equipment
- Infrastructure- Grants for Large Databases
- Framework Grant SIMSAM
- International Recruitment

For these types of grants the vice-chancellor, instead of the head of department, must sign Appendix S. Thereby (s)he confirms the commitments of the administrative organization. The details are specified in the Call for Proposals, published at www.vr.se.

Vetenskapsrådets noteringar
Kod

2014-35904-115258-21

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se