# 片上内存做普通的内存（PoM）

## 对软件透明

1. J. Sim, A. R. Alameldeen, Z. Chishti, C. Wilkerson, and H. Kim, “Transparent Hardware Management of Stacked DRAM As Part of Memory,” in Proceedings of the 47th Annual IEEE/ACM International Symposiumon Microarchitecture (MICRO), 2014.

这个就是我读的跟我想法最类似的文章！

1. C. Chou, A. Jaleel, and M. K. Qureshi, “CAMEO: A Two-Level Memory Organization with Capacity of Main Memory and Flexibility of Hardware-Managed Cache,” in *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2014

这个也是添加地址转换层，以cacheline为粒度进行转换，每次访问slow memory就进行swap

1. PageSeer: Using Page Walks to Trigger Page Swaps in Hybrid Memory Systems
2. GraphPIM: Enabling Instruction-Level PIM Offloading in Graph Computing Frameworks
3. memif: Towards Programming Heterogeneous Memory Asynchronously

基于用户信息指导片上内存的使用，虽然transparent文章很好，但是数据密集型负载移动的数据块不是2KB为单位，有时大有时小，因此用户指导在这种情况下就比transparent效果好

1. J. H. Ryoo, M. R. Meswani, A. Prodromou, and L. K. John, “Silc-fm: Subblocked interleaved cache-like flat  
   memory organization,” in Proceedings of the 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2017

使用基于锁的子块架构在片上内存和DRAM之间进行swap，最大化整个带宽。 粒度是从64B到2KB，支持两个段interleave数据到一个子块上。

## OS辅助进行数据迁移

# 片上内存做普通的Cache

X. Jiang, N. Madan, L. Zhao, M. Upton, R. Iyer, S. Makineni, D. Newell, Y. Solihin, and R. Balasubramonian, “CHOP: Adaptive filter-based DRAM caching for CMP server platforms,” in *In Proceedings of the 16th International Symposium on High-Performance Computer Architecture (HPCA)*, 2010.

G. H. Loh and M. D. Hill, “Efficiently Enabling Conventional Block Sizes for Very Large Die-stacked DRAM Caches,” in Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2011

组相连cache

J. Sim, G. H. Loh, H. Kim, M. O’Connor, and M. Thottethodi, “A Mostly-Clean DRAM Cache for Effective Hit Speculation and Self-Balancing Dispatch,” in Proceedings of the 45th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2012.

组相连cache

N. Gulur, M. Mehendale, R. Manikantan, and R. Govindarajan, “Bi-Modal DRAM Cache: Improving Hit Rate,  
Hit Latency and Bandwidth,” in Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2014.

## 对软件透明

1. X. Yu, C. J. Hughes, N. Satish, O. Mutlu, and S. Devadas, “Banshee: Bandwidth-efficient DRAM caching via software/hardware cooperation,” CoRR, vol. abs/1704.02677, 2017

该文章也把混合存储做cache

1. A. Prodromou, M. Meswani, N. Jayasena, G. Loh, and D. M. Tullsen, “MemPod: A Clustered Architecture for Efficient and Scalable Migration in Flat Address Space Multi-level Memories,” in Proceedings of the 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2017

使用Majority Element Algorithm (MEA)算法（原本用于数据库的）追踪和预测热页，迁移到片上内存中。

以下摘自：Banshee: bandwidth-efficient DRAM caching via software/hardware cooperation

把片上内存做cache时，优化其替换算法

1. Qreshi, M., and Loh, G. Fundamental Latency Trade-oﬀ in Architecting DRAM Caches: Outperforming Impractical DRAM-Tags with a Simple and Practical Design. In *MICRO* (2012)

Alloy Cache是直接映射DRAM Cache，粒度是cacheline，tag和data在DRAM Cache中毗邻存放，每次读取tag，会同时把data推测load出来。因此cache hit时的latency是访问单个DRAM的时间，cache miss时，需要访问一次片上内存和DRAM，从耗费带宽角度考虑，当发生miss时延迟和带宽会翻倍。

1. Jevdjic, D., et al. Unison Cache: A Scalable and Eﬀective Die-Stacked DRAM Cache. In MICRO (2014)

Unison以粗粒度形式存储数据并支持组相连，并提供way prediction降低hit latency

## Footprint cache

降低带宽压力，只缓存一个页面中，有可能被访问的部分。

[1]Jang, H., et al. Efcient Footprint Caching for Tagless DRAM Caches. In HPCA (2016).

[2]Jevdjic, D., et al. Die-Stacked DRAM Caches for Servers: Hit Ratio, Latency, or Bandwidth? Have It All with Footprint Cache. In ISCA (2013)

## OS辅助进行数据迁移

1. Meswani, M., et al. Heterogeneous Memory Architectures: A HW/SW Approach for Mixing Die-stacked and Oﬀ-package Memories. In HPCA (2015)

OS定期对页面进行排序，把热页放到片上内存中，并更新所有相关PTE，flush TLB， flush cache，因为开销大，因此remapp操作粒度是100ms-1s，这导致有可能丧失一部分局部性，并且在remapp过程中程序要停止运行。

1. Lee, Y., et al. A Fully Associative, Tagless DRAM Cache. In *ISCA* (2015).

全相连cache，与上述1类似，但是使用了硬件管理的TLB一致性机制，TDC在主存里维护一个TLB目录结构并当系统中任意TLB中的entry被插入或删除时更新上述TLB目录。这导致额外的设计复杂性，并且TLB目录可能导致多核情况下可扩展性不好，并且TDC没有讨论地址的一致性。

# 把片上内存同时用作cache和PoM

1. CHAMELEON: A Dynamically Reconfigurable Heterogeneous Memory System

本文首先分析把片上内存只用作cache会导致整体内存容量下降，影响性能，然后分析只用作PoM引发的Swap操作会导致严重带宽问题

1. Die-Stacked DRAM: Memory, Cache, or MemCache?

一部分做cache一部分做普通内存

# 其他优化

1. C. Chou, A. Jaleel, and M. Qureshi, “BATMAN: Techniques for Maximizing System Bandwidth of Memory Systems with Stacked-DRAM,” in the International Symposium on Memory Systems, 2017

优化swap，使得片上内存系统带宽最大化

1. D. Knyaginin, V. Papaefstathiou, and P. Stenstrom, “ProFess: A Probabilistic Hybrid Main Memory Management Framework for High Performance and Fairness,” in 2018 IEEE International Symposium on High Performance Computer Architecture, Feb 2018

提出基于成本的机制，决定swap方式，考虑到了不同程序之间的公平性。

3． 已有综述文章：A Survey Of Techniques for Architecting DRAM Caches

# 新的想法

1. 数据压缩技术是不是可以用在片上内存与DRAM之间的数据交换操作？毕竟以4K为粒度时进行数据传输，其中很多数据都没有被修改过。
2. 既可以按照页面（或大块memory）为粒度，也可以按照cacheline为粒度，大页面为粒度时，可以由三个寄存器标示一段的映射，Addr\_Fast, Size, Addr\_Slow ，Cacheline为粒度时，如何维护映射表？
3. 更精确统计冷热数据的方法，不是以页面而是以cacheline为粒度
4. 更快查到映射表的办法，从L2 Cache中就开始查映射关系，而不是到了Fast memory才开始查。把一级目录表放到L2中
5. 那些用于节约目录存储空间的办法（例如pointer-based），能不能用到我的idea？
6. 能不能为了加速我的idea，在一致性协议中加一个状态，例如MOESI基础上加一个X？例如Intel的Forward状态
7. 对于需要swap的地址，每个地址存储两个数据，一个是swap前的，一个是swap后的，例如Addr\_Fast: Data1, Data2, Addr\_Slow: Data2, Data1，这样在需要进行数据swap时，可以更快的完成？
8. 从目录一致性协议的角度，对片上内存提出优化想法
9. 当片上内存做普通内存并添加额外地址层使用时，粒度要么是2K，要么是4M，不可变，这导致有可能出现大量false sharing，并且没有考虑到多核多MC情况下，不同的MC所管理的内存数据是interleave的。
   1. 因此可以结合目录一致性协议设计一种灵活粒度的数据迁移操作，类似于可变粒度的预取。为了减少带宽浪费，也可以用数据压缩算法对数据进行压缩后传输。
   2. 可以根据LLC的miss类型，决定进行swap的粒度，例如conflict miss是4K，capacity miss是4M等。
10. **把预取和片上内存结合起来，自适应预取器，根据混合存储决定预取的方法及粒度**
11. 在片上内存做cache时，用一个buffer把修改数据的命令全部copy一份，当需要进行替换的时候，把上述命令再写一遍到DRAM中，而不需要数据来回拷贝，降低带宽浪费。
12. 把LDS访存模式跟片上内存结合起来，识别LDS后区别片上内存和普通内存进行区别对待？
13. 把NUMA环境下的remote cache prefetch算法用到片上内存预取上
14. 把张老师的博士文章想法（Impulse MC）用在混合内存
15. 虽然有人（Transparent Hardware Management of Stacked DRAM as Part of Memory.pptx）已经把我的想法用硬件实现了，但是我可以把地址翻译暴露给软件，让软件管理。
16. VA到PA的地址remap后，需要对remap的页面进行cache flush，否则可能导致错误，例如原始情况下：VA0->PA0，重映射后（需要把PA0拷贝到PA1）VA0->PA1，此时需要把PA0的数据flush掉，因为以后别的虚拟地址可能映射到该PA0页面，例如VA1->PA1。可以考虑不flush cache。
    1. 具体的，维护一致性协议的目录结构会对缓存里当前存储的地址进行记录，当发生上述remap时，在目录里标记所有访问PA1的请求全部转换成访问PA0
17. 把阿里巴巴公开的数据中心Alibaba Cluster Data数据跑在混合存储上，看看有啥特点需要改进？
18. 把QoS管理和混合存储结合起来，对于batch程序和latency程序使用片上内存时进行区分。
19. 把主动LLC写回和LLC替换算法结合起来进行混合存储带宽管理
20. DRAM Cache和普通DRAM之间的粒度如果可以变化，则会大大减少remap swap导致的带宽浪费
21. 把cache partition的想法用在片上内存
22. Hit/miss aware的llc替换算法
23. DRAM Cache一半用于普通cache，一半用来压缩。因为dram cache的访问时间和dram的时间差别较大 中间的压缩层相当于另一个cache
24. 实现所有pom cache结构，进行横行评估，发表IISWC
25. 统计Banshee中，被swap的页面里，有多少数据是热的，其实很可能没多少是热的，只swap其中热数据即可。解决办法是用footprint cache + 一个大的Dram Cache的写buffer，只缓存修改的数据。并且在Dram Cache中有一部分是压缩的，有一部分是不压缩的，减少swap操作的带宽开销。

# 相关知识积累

1. 各大厂商片上内存的数据，应用自[Banshee: Bandwidth-Eﬀicient DRAM Caching via Sofware/Hardware Cooperation]：Bandwidth/FLOPS2 ratio for Nvidia’s P100 (Pascal) system (0.14 B/FLOP) [4], Nvidia’s V100 (Volta) system (0.12 B/FLOP) [6], and Intel’s Knights Landing system (0.12 B/FLOP) [56]
   1. P100 [4] has 16GB in-package High Bandwidth Memory (HBM) with a bandwidth of 732 GB/s. Its cores provide 5300 GFLOPS for double-precision operations.
   2. V100 [6] has 16 GB of in-package HBM2 with a bandwidth of 900 GB/s. Its cores provide 7500 GFLOPS for double-precision operations.
   3. Intel Xeon Phi 7290 [56] has 16 GB in-package DRAM with bandwidth of 400+GB/s. Its cores provide 3456 GFLOPS for double-precision operations
2. 片上内存做Cache的工作对比，引自同上
3. 文章[CHOP: Adaptive Filter-Based DRAM Caching for CMP Server Platforms]中提到，把片上内存做Cache时，和普通DRAM之间进行数据替换导致的带宽浪费，可能比直接把片上内存做普通内存使用更严重。
4. 把片上内存做cache相关文章对比：

