## Unimem: Runtime Data Management on Non-Volatile Memory-based Heterogeneous Main Memory

总体概括：

NVM因其可扩展等特性逐渐被用来代替主存。由于NVM相对于DRAM来说带宽较低、延迟较高，所以NVM常与DRAM搭配使用来创建异构的Memory系统。本文就针对这种混合Memory系统设计了一套轻量级数据自动部署方案。该方案使用了在线的profile技术并减少了不必要的数据移动。

本文贡献：

本文做了如下三点贡献：

1，文章研究了多节点上存储大量数据的HPC应用上NVM与DRAM的性能差。

2，设计了轻量级的运行时系统并在不借助修改硬件的基础下进行对数据位置的管理。

3，使用六种HPC工作负载与一个生产代码来对Unimem进行测试，最终该系统能够成功降低两种介质的性能差别，并与现阶段最优的方案比较。

前期测试：

文章使用了Quartz作为nvm的模拟器，并总结了三个发现。

1：DRAM-only设备与NVM-only性能差别比较大，如图1所示。

2：一个好的数据放置策略可以降低DRAM与NVM之间的性能差。

3：不同类型的数据对带宽与延迟敏感度不同，如图2所示。Class c与class d则表现不同的影响效果。

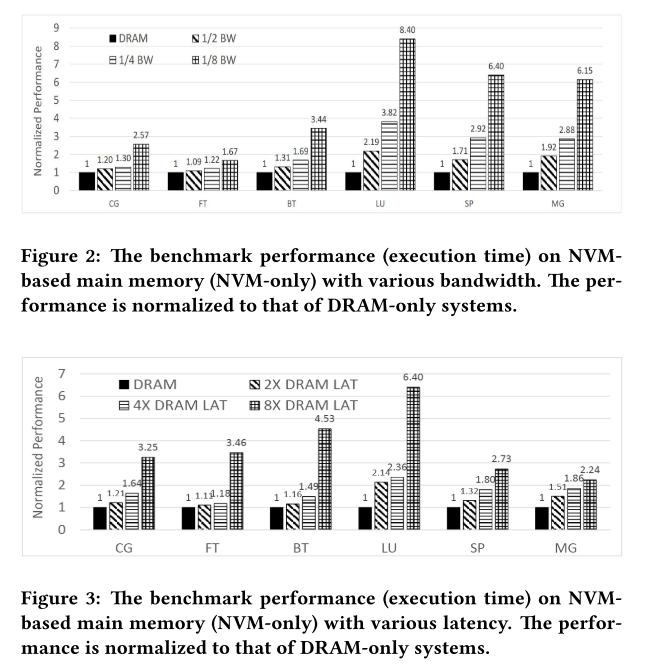


图1：NVM在不同延迟与带宽下的性能测试

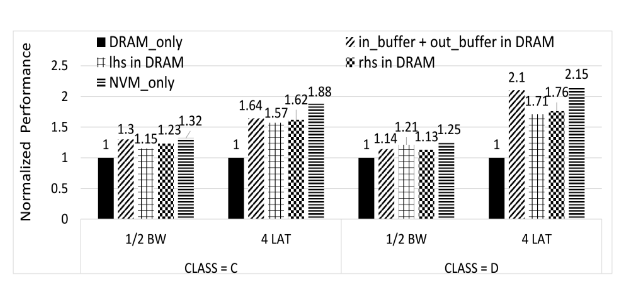


图2：NVM上不同数据放置模式对性能的影响

设计思路：

本文设计了Unimem系统用于管理数据的放置，并基于各个阶段进行对数据样本的采集。本文在整个工作流中设计了三个步骤：对各个阶段的样本采集、性能建模以及放置策略的设计。下面简单的对着三个阶段进行介绍。

Phase Profiling：这一步收集Memory各个阶段的访问信息，包括访问频率等。

Performance Modeling：在第一步中我们收集了各个阶段的对Memory的访问频次，本阶段用上述信息来计算数据是带宽敏感还是延迟敏感。在本节的计算结果分为了三种情况：计算出的带宽>=80%最大带宽就是带宽敏感；小于10%则是延迟敏感；处于中间为均敏感，于是取两者最大值即可。之后计算性能受益。除此之外，文章还计算了数据移动的开销（基于数据大小以及内存拷贝带宽）

数据移动决策：基于阶段并对阶段内的各个数据赋予权重w。根据权重w与剩余空间我们可以用0-1背包选择最优集合使得w最大。

优化方案：

1 各个iteration之间虽然是相似的，但是同样会有一些区别。这里实时监控性能，如果有10%的下跌那么就重新进行profile操作。

2 开始的时候为了减少数据移动的开销，于是将一些数据有选择性的放到dram中。这里使用编译器的方案去实施。

3 对大文件进行分割操作。