# Enforcing Crash Consistency of Evolving Network Analytics in Non-Volatile Main Memory Systems

总体概括：

本文关注动态图的应用，并针对图结构引入NVM进行优化工作。目前图结构在运行的过程中极易发生错误，在使用NVMM的前提下仍存在无法持久性保存的情况。除此之外，NVMM相比于DRAM具有非常严重的写延迟问题，如何设计合理的模式来将图结构应用在NVMM中是非常重要的。

本文提出了NVGraph用来持久化存储动态图结构，并提供错误恢复机制。NVGraph可以针对数据的冷热特征动态的调整内存与NVMM之间的数据分布，并减少NVMM的写操作次数。在图运行过程中至少有一个版本的数据存放在NVMM中以便进行错误恢复，而其他版本数据被分成两个部分分别存储在DRAM与NVMM中。最后该文章将NVGraph与四种图应用进行比较，在仅使用70%DRAM的情况下与其他应用效果相当。

本文贡献：

1文章提出了基于动态图结构的NVGraph数据结构，并用以扩展内存的容量。除此之外使得系统在多版本结构下可以进行错误恢复。

2使用当前数据结构网络的特征与访问特征动态地转移数据以减少NVMM的延迟代价。

3提供了易于编程的接口。并测试NVGraph与其他应用，通过对比得出一些结论。

设计细节：

在图应用中包括了应用定义的数据A，以及运行时所产生的数据R。这些数据均是多版本的大型数组。如图1所示，NVGraph的拓扑结构存储在NVMM中，对于S2版本来说，该版本数据分为两部分，一部分为n-1版本的基础数据以及n版本的增量数据。

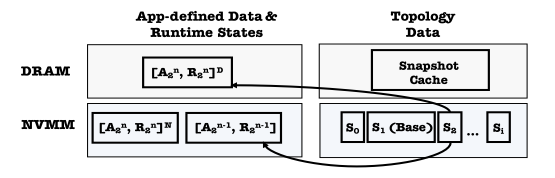


图1：NVGraph的数据结构图

在动态图运行的过程中，我们通过在原来结构的基础上增加动态增量以代替生成全部节点来表示。而运行过程中的数据（最新版本）均在DRAM中生成，上一个保存完成的版本则保存在NVMM中以便进行错误恢复，运行状态图如图2所示。

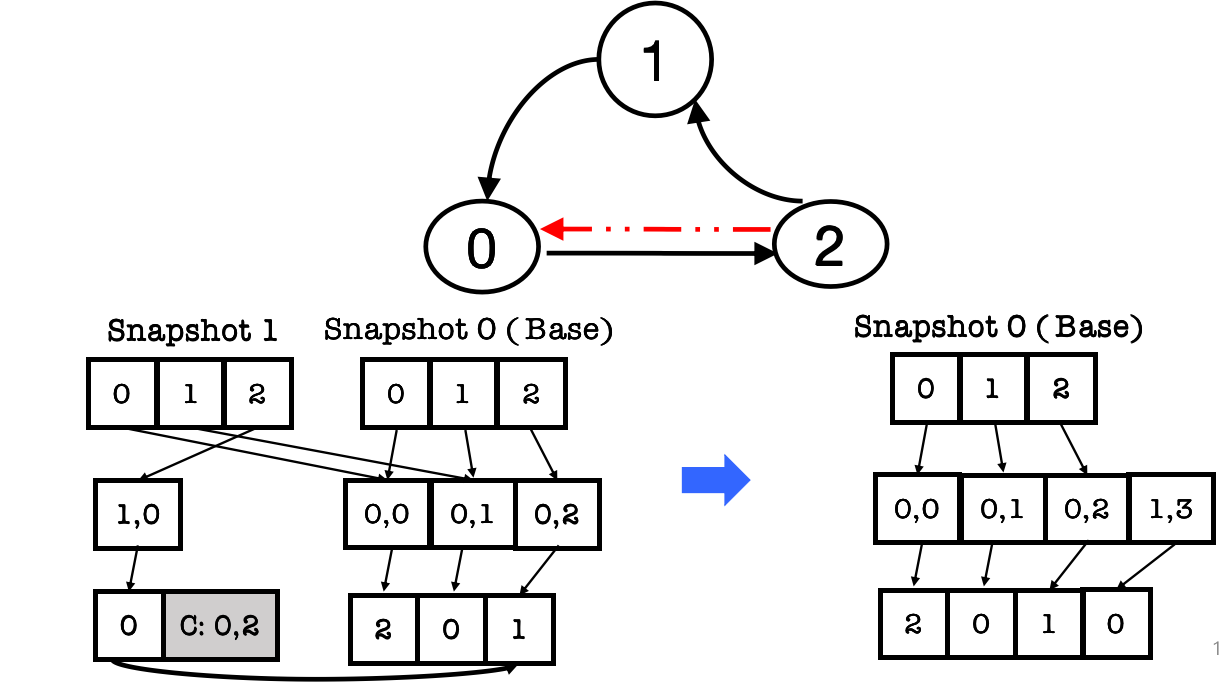


图2：动态图运行时NVGraph的表示形式

NVGraph在运行时包括了插入节点、删除节点、加载数据、垃圾回收、数据持久化、冷热数据迁移、更改基础镜像、数据恢复操作。在插入操作进行时，NVGraph选择产生新的数据表并且将新的所需插入的数据放入到此数据表中；删除操作并不是立刻进行删除，而是将数据表中的数据标记“delete”，并在后期的垃圾回收中进行整体删除；当NVMM中的空间不足时，垃圾回收操作便进行，并清除掉之前被标记的数据以便释放空间；数据持久化操作表示当第n步完成时，NVGraph将Dram中的数据同NVMM中的数据以及两个版本之间共享的数据进行合并，并标记上一个版本“delete”；NVGraph同样会判别冷热数据并有针对性的对不同类型数据进行分配；数据恢复操作发生在应用出现错误时，此时NVGraph根据NVMM中保存的n-1版本的快照来对数据进行恢复。

本文中最重要的地方算是冷热数据的选择。选取热数据的步骤如下：（1）选择节点。使用了一些启发式算法并根据他们的内存访问频率将他们分组，我们发现访问频率最高的前百分之十的节点具有最高的中心特征值，所以我们根据此来识别出来热点，如图3所示。

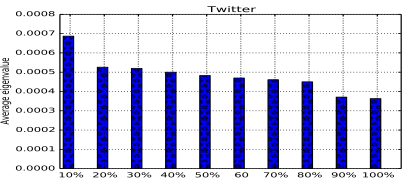


图3：访问频率不同的节点的中心特征值信息

步骤（2）选择边。使用Jaccard系数的倒数作为量化的指标来表示i到j的重要性。

步骤（3）增量更新。NVGraph将上述寻找出来的点和边信息保存在NVMM中，并使用增量的改变减少开销。