**一、研究目标、研究内容与技术指标**

**（一）研究目标**

**（存储层）数据保护过程中的系统性能优化**

针对，提出一种基于文件系统的快照方法实现的WOWSnap，通过支持扩展单元和块单元分配策略提供广泛的可移植性，最大限度地减少ROW中连续快照映像之间的I / O处理开销；提供了在一致大小的磁盘部分中维护分配了快照映像的磁盘空间的功能。

**（二）研究内容**

**基于能力感知的主动存储技术**

**2.1 Minimize I/O Overhead**

为了最小化在复制inode和数据块中发生的I / O处理开销，我们选择采用基于ROW的快照方法。此外，在采用ROW方法时，我们使用预先分配的元数据来减少每个瞬时快照的块分配时间。当创建活动文件的inode时，我们还分配一个额外的inode用于以下快照映像。

在获取时间点快照时，此附加inode将成为快照文件的inode。此外，快照inode只是链接指向原始块的指针，而不是简单地复制快照文件的所有关联块，以表示这些数据块在活动文件和快照文件之间共享。通过这种方式，与COW（写入时复制）方法相比，我们可以显著减少块分配的处理时间。

**2.2 Provide Portability提供可移植性**

WOWSnap可以与基于范围的存储结构相结合，其中连续数量的块被分配给文件段，并且可以与基于块的存储结构组合。

当WOWSnap与基于范围的结构（如XFS [1,2]）结合使用时，每个inode（包括快照映像的inode）都包含由三个组件组成的扩展区：起始块地址，描述该数字的块计数 连续分配的块，以及描述属于该范围的数据块的共享的位图标志。

如果位图标志被设置为1，则它将表示最近分配了属于该范围的数据块，因此当前没有其他inode共享这些数据块。 如果该标志设置为0，那么它将意味着属于相关范围的数据块在文件之间共享，因此需要进行可靠的块管理，同时对这些块进行数据块的修改。

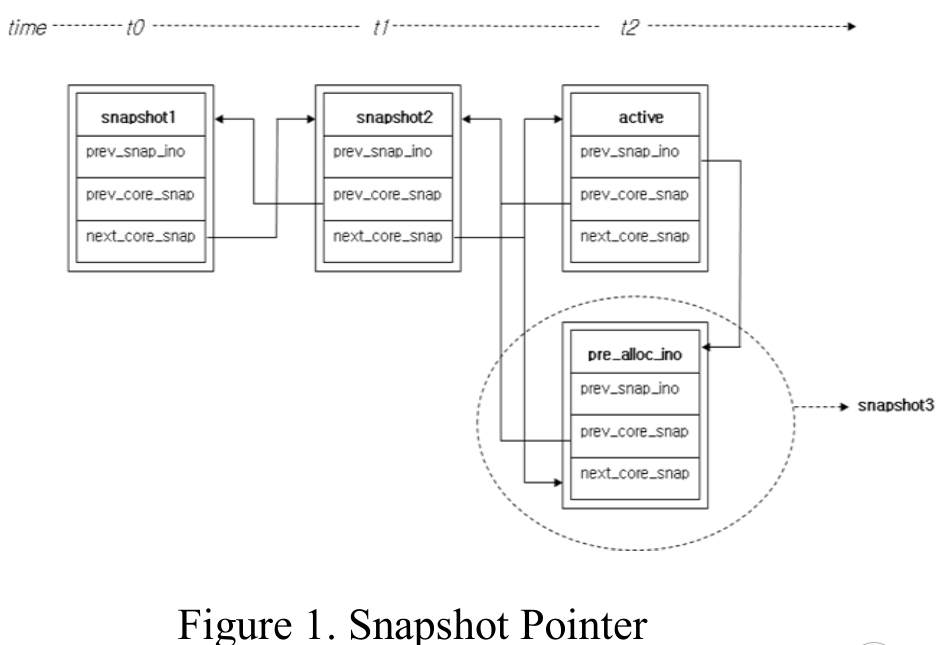
**2.3 Manage Disk Space for Snapshots**

我们开发了一种快照空间算法，使我们能够为快照映像分配的磁盘空间尽可能小。 在WOWSnap中，所有快照映像（包括其活动文件）都按两个指针分组。在每个快照组中，活动文件在前面链接，最旧的快照图像在后面链接。

构建活动文件的快照组使我们能够来回跟踪链接，以查找文件之间共享的数据块。当WOWSnap在遍历快照组时发现损坏或备份的快照映像时，快照空间算法可以检查属于这些映像的数据块的共享。它将关联的inode与快照组取消链接，从而将快照映像保存在一致大小的磁盘部分中。

**（三）关键技术**

**3.1 Snapshot Pointer**



创建活动文件的inode时，还会预先分配其他inode，并使用prev\_snap\_ino将其与活动inode链接。 如果采用下一个快照，则不需要分配和复制活动文件的inode，因为在预分配了额外的inode时已经执行了复制inode属性。

此时唯一要执行的是调整两个指针prev\_core\_snap和next\_core\_snap，并设置范围的位图值以指示数据块的共享。在WOWSnap中，快照守护程序定期发出快照调用以检查快照完整性。

**3.2 Overall Structure**

WOWSnap为每个扩展区结构分配一个位图值，以指示数据块的共享，如图2（a）所示。 如果位图值为零，则表示不能修改范围的数据块，因为这些块必须与其他文件共享。 否则，可以修改范围的块。

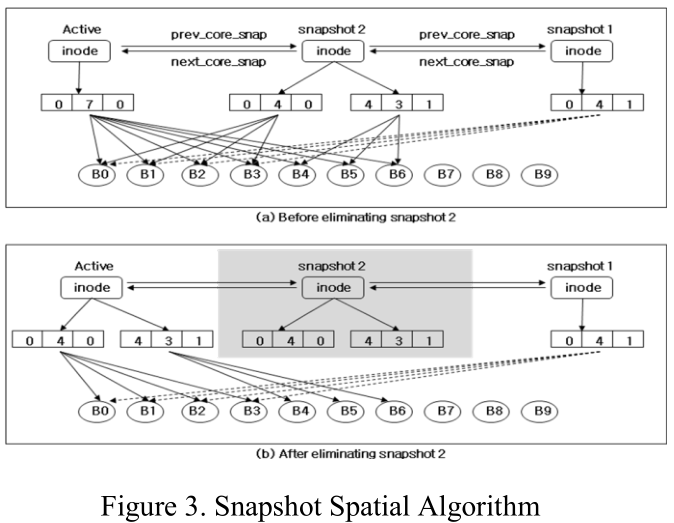
图2（a）显示了快照结构的示例。 该文件由五个数据块B0到B4组成，其范围包括三个组件：起始块号，块计数和位图值。 位图值的初始值为1，表示当前没有其他文件共享属于该文件的数据块。

更改位图值会对数据修改和删除产生显着的性能影响，因为通过检查当前位图值，WOWSnap可以轻松确定是否可以更新数据块。如果由于多个文件之间的共享而发现无法修改数据块，则它会分配新数据块以接收最新值。此外，与数据块关联的范围被拆分以分隔新块。图2（c）显示了在拍摄第一个快照后发生更新过程的示例。

在图2（c）中，更新需要修改两个块B3和B4。由于这两个块在活动文件和第一个快照之间共享，因此分配了两个新块B5和B6，并对这两个块执行更新过程。此外，为了反映更新的新块分配，创建了一个额外的范围，并且其位图值最初设置为1，因为当前没有其他文件与活动文件共享这些新块。

**3.3 Snapshot Spatial Algorithm**

在WOWSnap中，我们提供快照空间算法，以持续维护快照历史记录。空间算法由快照守护程序定期执行。守护程序负责删除已损坏或备份到其他磁盘的快照。



假设快照守护程序发现第二个快照映像快照2已损坏。 空间算法的消除过程要求守护进程遍历prev\_core\_snap和next\_core\_snap，以确定是否可以解除分配属于快照2的数据块。

如果位图值为零，则无法修改属于此范围的块，因为这些块与prev\_core\_snap链接的其他文件共享。另一方面，在位图值为1的情况下，属于该范围的块不与前面的快照共享。但是，这些数据块可能与链接到next\_core\_snap的其他后续快照共享。因此，在解除分配属于快照2的数据块之前，应检查链接到next\_core\_snap的快照映像。