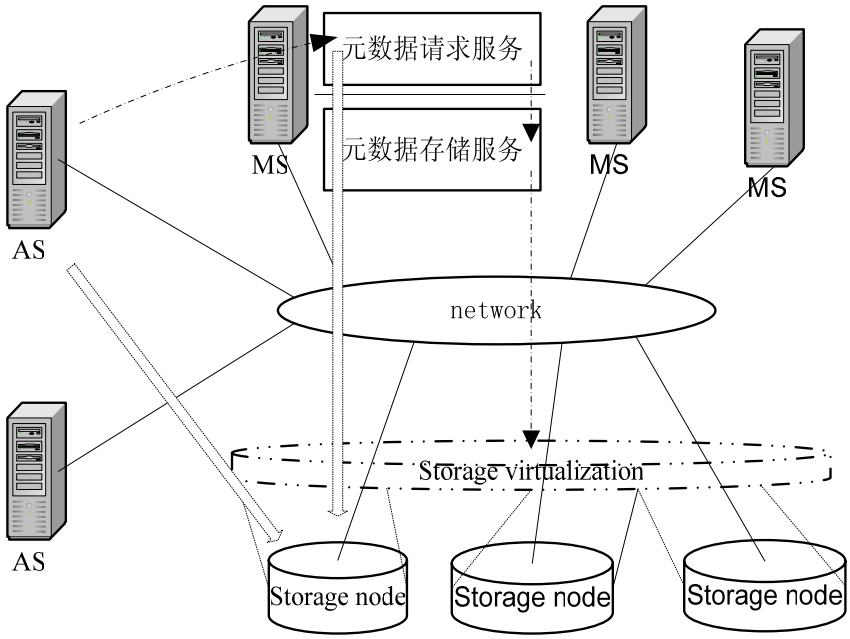

图 2.3 BWMMS 基于集中共享虚拟存储的对称服务器结构

结合图 2.2，存储节点（Storage Node，SN）是提供物理存储资源的物理资源拥有者，它将自己管理的物理存储资源提供给存储虚拟化层（Storage Virtualization Layer, SVL）管理，SVL 是图 2.2 中的逻辑资源提供者，它管理物理存储资源和逻辑存储资源的映射，提供统一的逻辑存储资源空间给逻辑存储资源管理者。元数据服务器（MS）充当逻辑存储资源管理者和逻辑元数据资源拥有者的角色，管理从 SVL 获得的逻辑存储资源，批量的、主动和被动结合的逻辑存储资源分配和释放发生在 MS 和 SVL 之间。AS 是系统的逻辑资源使用者，AS 的元数据请求语义驱动 MS 完成逻辑元数据资源、逻辑存储资源的分配和释放。

在 BWMMS 的集中共享存储架构中，元数据的存储和访问彻底分离。元数据可以存放在任何存储位置，元数据请求也可以分配给任何请求服务器。在此基础上，元数据服务器形成对称的服务器结构，元数据请求可以分布到任何服务器，消除了私有存储资源使用模式中、为平衡请求处理负载而带来的非常困难的元数据存放策略问题，也为灵活的元数据请求分布管理提供坚实的基础。

2.4 本章小结

文件系统元数据服务由元数据存储服务和元数据请求服务构成，元数据存储服务是文件系统元数据服务的基础。元数据存储服务的存储资源组织、管理和使用机制将影响元数据请求服务的有效性，本章对有效的元数据存储服务进行了探讨。

基于业界成熟的虚拟化存储理念，BWMMS 集中组织物理存储资源，并以 64 位的逻辑存储资源标识为系统存储资源扩展提供支持。通过集中化和虚拟化，存储资源能够得到有效利用，存储资源管理的扩展能力得到提高。

BWMMS 提出分布式层次化的存储资源管理机制。层次化将系统存储资源管理功能层次化、模块化，提高各个部分的独立性。分布式的存储资源管理将存储资源管理功能分散，消除存储资源管理的瓶颈。64 位的元数据逻辑资源标识为文件系统扩展提供支持，元数据的动态分配和动态映射机制，为文件系统支持逻辑存储资源的扩展提供基础。

以集中虚拟化存储和分布式层次化存储资源管理机制为基础，BWMMS 提出统一的逻辑文件系统名字空间视图，以完全共享的使用模式管理用户的存储共享。通过完全共享方式，元数据的存储与元数据的访问形成松耦合关系，元数据的存放位置不再限制处理元数据请求的服务器。元数据可以存放在存储资源的任意位置，并将其请求指定给任意服务器处理，服务器间形成对称的结构。共享的存储资源使用模式，有效地消除了私有使用模式中、为平衡请求处理负载而存在的非常困难的元数据放置问题，为灵活的元数据请求服务提供坚实的基础。

第三章 元数据请求分布管理

元数据请求服务构建在元数据存储服务基础上，响应用户的元数据请求。它需要在保证单个元数据请求处理效率的前提下，综合考虑元数据服务器的负载，避免因服务器负载热点而限制元数据服务的扩展。元数据请求服务的核心是如何在元数据服务器间有效分布元数据请求。应用的元数据请求表现出动态变化的特征，如何满足用户动态的元数据服务需求是元数据请求分布管理的关键。

本章主要讨论元数据请求分布的有效管理问题。首先介绍元数据访问协议集合，元数据访问的统计特征和相关研究的元数据请求分布管理。然后介绍 BWMMS 的动态元数据请求分布管理的机制和策略，包括请求分布的对象、请求分布的管理机制和协议、请求分布的策略等。最后通过实验评估 BWMMS 的元数据请求分布管理的有效性。

3.1 文件系统元数据访问协议

元数据请求分布管理需要明确分布管理的元数据请求类别，以确定请求分布决策时机，并根据元数据请求的类别加以区别地对待请求的分布。

文件系统元数据访问协议包括文件系统元数据的访问请求和文件元数据的访问请求两大类[Opengroup.org][SMB1987][SUN1989][Callaghan1995]。

文件系统元数据的访问请求包括查询文件系统状态，在文件系统名字空间中查找、创建、移动或删除文件等请求。

查询文件系统状态的请求主要访问文件系统超级块的内容，获取文件系统的资源使用情况，文件系统的统计信息等状态信息。

查找文件的元数据请求用来明确文件系统的名字空间是否存在指定的文件，其格式为（父目录，文件名），期望返回被查询文件名的索引节点。在本地文件系统中，查找文件的元数据请求隐含在文件名解析过程中。在分布式文件系统中，为支持异构的元数据服务器，并减少协议的通信开销，文件查找需要显式地进行，返回被查找文件的索引节点。目录内容读取请求获取目录一定范围内的目录项，其参数格式为（目录，开始位置，读取内容长度）。文件查找和目录内容读取不更改任何元数据，且其访问统计比例很高。

创建文件类元数据请求需要在文件系统名字空间加入新的文件、或者指向已经存在的文件的目录项，包括创建文件、符号连接和硬连接等。文件的创建将在目录下生成指定名字的新文件，请求格式为（父目录，文件名，文件类型）。“文件类型”是 Unix 文件系统支持的普通文件、目录和特殊设备节点等。创建符号连接的参数格式为（父目录，符号连接名，目标文件）。目标文件是被连接的文件名字，它可以是普通文件，也可以是

目录。创建文件和符号连接都需要分配新的索引节点，并在父目录中添加指向该索引节点的目录项。创建硬连接的参数格式为（父目录，硬连接名，目标文件名字）。与文件和符号连接创建不同的是，硬连接创建没有索引节点的分配，且目标文件只能是普通文件。它根据目标文件名获得索引节点号，在父目录中添加指向该索引节点的目录项，并更改目标文件的索引节点。

移动文件的元数据请求将文件（旧文件）从一个目录（源目录）移动到另一个目录（目标目录），同时可能将文件名字改成新的文件名字（新文件）。文件移动请求首先需要在目标目录中添加目录项，指向新文件。如果目标目录中存在与新文件同名的文件（同名文件），需要在添加指向旧文件的目录项之前，处理目标目录与同名文件之间的连接。在目标目录的处理完毕后，在源目录中删除指向旧文件的目录项，并更改旧文件的索引节点。文件移动操作最多将更改 4 个元数据，是最复杂的元数据请求。重命名文件请求是移动文件请求的特殊情况，它完成同一个目录下的文件移动。

删除文件的元数据请求完成目录和文件的连接的删除，其参数格式为（父目录，文件名）。它需要删除父目录中的目录项，更改被删除文件的索引节点。

针对文件的元数据访问请求主要包括读写文件的索引节点、获取文件数据所在的存储设备块号和读写权限等，这些请求仅需要对单个文件的元数据进行访问。

3.2 用户元数据访问特征

根据已有的研究结果，用户的元数据访问的主要特征可以概括为：

1. 聚合元数据请求数目所占比例非常大。

表 3.1 已有研究结果的元数据请求数目比例

	Total	Data	Metadata	Metadata%
NFS	6,680,000	1,640,000	5,040,000	75.45
AFS	1,615,500	285,500	1,330,000	82.33
Sprite	432,000	264,000	168,000	38.89
INS	8,300,000	2,470,000	5,830,000	70.24
RES	3,200,000	374,000	2,826,000	88.31
NT	3,870,000	1,501,000	2,369,000	61.21
CAMPUS	29,900,000	25,220,000	468,000	15.65
EECS	2,300,000	788,000	1,512,000	65.74

如表 3.1 所示(其中 Total，Data 和 Metadata 列的单位是次，表示请求次数)，在软件开发、Web 服务和 E-Mail 服务等应用中[Ousterhout1985][Baker1991][Dahlin1994-1][Gibson1997][Roselli2000][Ellard2003]，元数据请求数目的比例比较高，最多达到近 89%。

2. 从文件系统整体而言，文件系统元数据表现出部分活跃性[1]。

[1] 活跃文件指的是创建完成后，还会被再次访问的文件。