

基于矩阵操作的必然性 QoS 约束副本放置方法

付伟, 叶清, 吴晓平

(海军工程大学 信息安全系, 武汉 430033)

摘要 如何提供严格有保障的数据服务质量, 为每个数据访问请求提供满足必然性 QoS 约束的数据服务, 是数据网格成功的关键. 在副本放置模型的基础上, 给出了 QoS 约束形式化的描述方法, 并得出必然性 QoS 约束满足的形式化定义. 针对必然性 QoS 约束的副本放置问题. 在矩阵表示的基础上, 利用矩阵操作给出一种有效的副本放置算法. 分析和实验表明, 该算法能够很好地解决各种规模条件下必然性 QoS 约束的副本放置问题, 并具有稳定性和可扩展性的特点, 在多种网络拓扑和访问模式下均能获得满足必然性 QoS 约束的副本策略.

关键词 数据网格; 必然性 QoS; 副本放置

Placing replicas under certain QoS restriction using matrix operations

FU Wei, YE Qing, WU Xiao-ping

(Department of Information Security System, Naval University of Engineering, Wuhan 430033, China)

Abstract It is critical for data grid to provide strictly QoS guarantee. Certain-QoS-Restriction replication service guaranteed strict QoS satisfaction for every individual user, which is very important in some grid applications. A graph-based replica placement model was built, and a formula description method was presented. With the help of self-designed matrix operations, a novel algorithm was proposed. Analysis and experiments show that the algorithm is feasible in arbitrary problem scales and can obtain approximate-optimal replica policy under different simulating conditions.

Keywords data grid; certain QoS restriction; replica placement

1 引言

数据网格^[1](data grid)将高端计算技术与高性能网络技术、分布式存储技术和数据管理技术结合起来,为大规模应用提供广域环境下海量数据的管理功能.如何提供 QoS 有保障的数据服务,是数据网格相关应用成功的关键^[2].服务质量(QoS)描述了一个服务满足客户需求的能力,是服务好坏的定量度量.数据服务的 QoS 可以从响应时间、吞吐量、可靠性、可用性、安全性和并发处理能力等等方面来描述^[3-4],而这些不同方面对于特定系统而言具有不同的权重,可用不同的加权值加以区分.副本(replica)技术在网格系统中的不同位置分散存储数据的拷贝,所有用户向“距离自己最近”的拷贝发起数据请求,可有效缩短响应时间和避免远距离的数据传输^[5-6],从而提高数据服务 QoS.

很多文献对如何保证分布式系统的 QoS 做了深入研究,但是它们主要从系统全局的角度出发,将服务的某些总体指标作为 QoS 评价的依据,如上面提到的请求响应时间、访问延迟时间等等.从本质上来看,这些指标实际上是一种大概率事件的保证.例如,假设某系统的 QoS 要求“请求响应时间 $\leq 3s$ ”,实际上是要求每次访问的请求响应时间有很大概率(如 $> 95\%$)不超过 3 秒钟,它并不能保证所有访问都能在 3 秒钟之内得到响应.我们将这种传统的 QoS 约束称为或然性服务质量约束 PQR(PQR, probable QoS restriction),它反映的是服务的整体状况,对服务的部署、评估和优化具有很高的参考价值^[7].

收稿日期: 2011-07-25

资助项目: 国家自然科学基金(61100042); 海军工程大学自然科学基金(HGDQNJJ023)

作者简介: 付伟(1978-),男,博士,讲师,研究方向:分布式计算,信息安全;叶清(1978-),男,博士,讲师,研究方向:信息安全,网络安全;吴晓平(1961-),男,博士生导师,研究方向:应用数学,信息安全.

然而, 随着网格应用领域的不断扩展, 各种应用对于服务质量的要求正在提高, 出现一类对 QoS 要求更为严格的应用. 例如, 在股票市场上, 股东需要实时获得股票信息. 当股票信息发生变化时, 每个股东都必须在第一时间内获取到最新的股票价格. 如果因为股票信息系统的原因导致某个股东在经过一段时间延迟之后才获得所需数据, 这将使其错过股票交易的最佳时机, 从而可能蒙受重大经济损失. 又如, 在大规模联合作战中, 分布在战场各地的指挥员都必须在允许的时间范围之内获得最新的战场情报. 情报分发系统必须将作战指令按照严格的时间要求传送给每个指挥员, 任何一个指挥员不能及时获得指令都将影响作战效果, 甚至因为协同作战的失败而导致整个战局的溃败. 在金融、实时通信、数据分发和多媒体在线游戏等领域中均可以找到同样的应用需求 [8].

以上这些应用场景的共同之处在于: 每个用户请求的 QoS 需要“无一例外”、100% 地得到满足. 与 PQR 相对应地, 我们称之为必然性服务质量约束 (CQR, certain QoS restriction). 可以这样认为: PQR 与 QoS 指标的平均值有关, 而 CQR 则与 QoS 指标的峰值有关.

本文针对数据网格系统中的 CQR 问题, 研究通过副本放置技术保证数据服务的 CQR. 主要贡献是:

- 建立了基于图论的副本放置模型, 给出 CQR 的形式化描述方法和 CQR 副本放置问题的形式化定义;
- 提出基于矩阵操作的 CQR 副本放置算法, 将 QoS 请求差额作为启发信息, 求解获得满足所有用户 CQR 的副本放置策略;
- 设计和开发了一个副本策略测试与验证平台, 并进行了实验验证与对比.

2 系统模型与问题定义

2.1 副本放置模型定义

在数据网格系统中, 对于某个数据资源而言, 数据服务器可以划分为两类: 保存了该数据的服务器和未保存的服务器. 前者称为副本节点, 可以直接满足客户端的数据请求; 而后者称为非副本节点, 它们的数据请求需要向副本节点转发. 将数据网格系统中的数据服务器抽象成一个无向连接图 $G=(V,E)$. V 为数据服务器节点集合; E 为节点之间的链路集合, 边的长度表示该边所连接的节点对之间的通信开销. 任意两个节点之间都可直接或者间接地建立连接; 一个节点可以读取或者复制另一个节点的数据; 节点也可以修改本地的数据内容. 特别地, 包含原始数据的节点称为源节点.

2.2 CQR 形式化描述方法

为了便于描述, 下面给出符号定义如表 1 所示.

表 1 符号定义

符号	说明
S	数据服务, 可访问源节点或者任一副本节点获得
Q, q	QoS 属性集合为 Q , 属性个数 $ Q = m$; 其中单个属性为 $q, q \in Q$
λ_q	QoS 属性 q 所对应的加权值
B_q, b	q 的取值约束范围 (bound) 集合为 B_q ; 其中单个约束为 $b, b \in B_q$
op	属性约束操作, $op \in \{\subseteq, \leq, \geq, <, >, =, \neq\}$
C, c	客户端请求集合为 C ; 其中单个请求为 $c, c \in C$
N	节点集合, 节点个数 $ N = n$
R, r	副本节点集合, 副本个数 $ R = w$; 其中单个副本节点为 $r, r \in R$

在此基础上, 给出几个基本定义:

定义 1 QoS 属性满足: 对于某个 $c \in C$, 对于 QoS 属性 q 的满足定义为:

$$QoS_Satisfy(c, q) = \begin{cases} True, & \text{if } q \text{ op } B_q \\ False, & \text{else} \end{cases} \quad (1)$$

对于一次客户请求 c 获得的服务质量 q , 如果 q 满足对应的 QoS 需求约束 B_q , 则称 c 获得 QoS 属性 q 的满足. 反之, 对于未获得 B_q 满足的 c , 客户获得的 QoS 必然不满足 QoS 约束, 即存在差额. 这里给出 QoS 请求差额的定义:

定义 2 QoS 请求差额: QoS 请求差额定义为:

$$QoS_Gap(c, q) = \begin{cases} 0, & \text{if } QoS_Satisfy(c, q) \\ |B_q - q|, & \text{else} \end{cases} \quad (2)$$

定义 3 必然性约束 CQR: $\forall c \in C, \forall q \in Q$, 服务 S 的 CQR 定义为:

$$QoS_Certain(S) = \sum_{q \in Q} \sum_{c \in C} \lambda_q QoS_Gap(c, q) \quad (3)$$

即必然性约束是指所有客户请求的各个属性的 QoS 请求差额之加权和。

定义 4 必然性约束满足: 如果 $QoS_Certain(S) = 0$, 则称服务 S 是被必然性 QoS 约束满足的. 显然, 如果一个服务是被 CQR 满足的, 那么对于任何一个请求, 其所有 QoS 属性都应是满足的.

2.3 问题的形式化定义

数据的副本放置问题是副本管理的一个重要研究内容, 传统研究主要针对 PQR, 并未考虑 CQR. 为了适应某些数据网格应用中的 CQR 需求, 在 2.2 节形式化描述方法基础上, 本文给出如下定义:

定义 5 CQR 满足的副本放置问题: 给定一个数据网格系统 $G = (V, E)$ 及数据服务 S , 必然性 QoS 满足的副本放置问题是指从节点集合 V 中找到一个副本节点子集 R , 使得 V 中所有节点获得 S 的服务质量满足必然 QoS 约束, 即 $QoS_Certain(S) = 0$.

3 基于矩阵操作的副本放置方法

3.1 算法思想

可以证明, CQR 满足的副本放置问题可以规约为一个最小集合覆盖的问题^[9], 因此可知 CQR 满足的副本放置问题必然也是一个 NP 完全问题. 本文提出一种启发式算法, 逐步减小数据服务的 QoS 请求差额, 直至 CQR 为 0; 同时考察每个候选的非副本节点, 判断其成为副本节点之后所引入的副本开销. 在每次循环中, 选择能减少最多服务 QoS 请求差额、同时引入开销最小的节点作为新增副本节点, 直至 QoS 请求差额减少为 0. 在基于图论的抽象模型 $G=(V,E)$ 基础上, 为了简化算法的描述并提高算法效率, 引入矩阵表示, 并自定义了 3 种矩阵操作, 可有效加速计算过程.

3.2 矩阵表示与矩阵操作

假设系统中包含 n 个节点, 每个节点的 QoS 需求包括 m 个元素. 给出如下矩阵表示:

1. 质量约束矩阵 (bound matrix): $BM = (b_{i,j})_{n \times m}$. 其中元素 $b_{i,j}$ 表示节点 i 对数据服务的第 j 个 QoS 质量的约束.
2. 服务质量矩阵 (QoS matrix): $QM^j = (q_{i,k}^j)_{n \times m}$. 其中元素 $q_{i,k}^j$ 表示节点 j 提供数据服务时, 节点 i 可获得的第 k 个 QoS 质量. 服务质量矩阵描述了单个节点 j 提供独立数据服务时的服务能力.
3. 加权向量 (weight vector): $WV = [\lambda_{q_1} \lambda_{q_2} \cdots \lambda_{q_m}]^T$. 其中 λ_{q_i} 表示第 i 个方面的 QoS 约束在所有 QoS 约束中所占的比重.

为了衡量某个节点是否适合为另一个节点提供数据服务, 给出服务质量距离的概念.

定义 6 服务质量距离 (QoS distance): 服务质量距离定义为一个节点 i 访问另一个节点 j 提供的数据服务时所获得的 QoS 的加权代数和, 即服务质量距离 (QoS distance):

$$d_{i,j} = QM^j[i] \times WV = \sum_{1 \leq k \leq m} q_{i,k}^j \times \lambda_k \quad (4)$$

4. 服务质量距离矩阵 (QoS matrix): $DM = (d_{i,k})_{n \times n}$.

显然, 所有节点对之间的服务质量距离构成一个 $n \times n$ 的矩阵 DM , 其中所有节点对之间的服务好坏程度被归一化为统一的服务质量距离. 在本文的算法中主要根据该距离给出启发式信息, 目标就是使得每个节点都能在 QoS 约束范围之内找到某个副本提供的数据服务. 为了有效利用矩阵计算可并行化的优势, 定义 3 种矩阵操作:

操作 1 矩阵最小并: 对于矩阵 $X = (x_{i,j})$, $Y = (y_{i,j})$, 则 X 和 Y 的矩阵最小并记为: $Z = X \oplus Y = (z_{i,j})$, 其中 $z_{i,j}$ 为 $x_{i,j}$ 和 $y_{i,j}$ 的较小值. 显然矩阵最小并具有对称性和可交换性.

操作 2 矩阵正向差: 对于矩阵 $X = (x_{i,j})$, $Y = (y_{i,j})$, 则 X 和 Y 的矩阵正向差记为: $Z = X \ominus Y = (z_{i,j})$, 其中:

$$z_i^j = \begin{cases} 0, & \text{if } x_{i,j} \text{ op } y_{i,j}, \\ |x_{i,j} - y_{i,j}|, & \text{else.} \end{cases}$$

操作 3 矩阵和: 对于矩阵 $X = (x_{i,j})$, 其矩阵元素和记为: $\sum(X) = \sum_{\forall i} \sum_{\forall j} x_{i,j}$. 矩阵和是矩阵中所有元素的代数和.

5. 访问向量 (access vector): $\mathbf{AV} = [a_1 \ a_2 \ \dots \ a_n]^T$. 其中元素 a_i 表示当前副本集合中与节点 i 之间服务质量距离的最小值.

由于每个节点都会访问与其服务质量距离最小的副本, 因此利用矩阵最小并及其对称性和可交换性, 可以得到如下结论:

定理 1 设副本节点集合为 $R = \{r_k | k = 1, 2, \dots, w\}$, 则该副本系统所提供服务的访问向量为: $\mathbf{AV} = ((\mathbf{QM}^{r_1}) \oplus (\mathbf{QM}^{r_2}) \dots \oplus (\mathbf{QM}^{r_w})) \times \mathbf{WV}$

实际上 \mathbf{AV} 的各个元素应当是 \mathbf{QM} 对应位置上访问距离最短 (即元素最小) 的数值, 即为对应矩阵最小并的结果, 因此易知定理 1 的正确性.

引理 1 在当前副本集合中增加一个新的副本节点 k , 则新的访问矩阵为:

$$\mathbf{AV}(k) = \mathbf{AV} \oplus (\mathbf{QM}^k \times \mathbf{WV}) \quad (5)$$

由于 k 节点从非副本节点变成副本节点, 因此 k 节点本身和 k 附近节点的 QoS 性能可以得到提升, 对应的 QoS 请求差额必然减少, 从而使得 CQR 也减小. 当其逐渐减小为 0 时, 当前副本集合即为所求满足 CQR 的副本策略.

定理 2 数据服务 S 的 CQR 等于访问矩阵与质量约束矩阵的正向差的矩阵和, 即:

$$QoS_Certain(S) = \sum((\mathbf{AM} \ominus \mathbf{BM}) \times \mathbf{WV}) \quad (6)$$

证明 令 $\mathbf{M} = (\mathbf{AM} \ominus \mathbf{BM}) = (m_i^j)$. 根据定义:

$$m_i^j = \begin{cases} 0, & \text{if } a_{i,j} \text{ op } b_{i,j} \\ |a_{i,j} - b_{i,j}|, & \text{else} \end{cases}$$

由定义 1 及定义 2 可知, 实际上 B_q 即为 $b_{i,j}$, q 即为 $a_{i,j}$, 于是: $\sum((\mathbf{BM} \ominus \mathbf{QM}) \times \mathbf{WV}) = \sum(\lambda_i \times m_i^j) = \sum(QoS_Gap(i, j) \times [\lambda_1 \ \lambda_2 \ \dots \ \lambda_n]^T) = \sum_{\forall i} \sum_{\forall j} \lambda_i QoS_Gap(i, j)$, 结合定义 3 知命题成立, 证毕.

3.3 算法描述及复杂性分析

根据以上形式化描述方法和定理, 提出一种基于矩阵运算的 CQR 满足的副本放置算法. 该算法以各个节点的 QoS 约束和单独数据服务能力作为输入, 获得的输出是一个符合 CQR 满足的副本节点集合. 算法的流程描述如图 1 所示.

\mathbf{DM} 的构造需要 $n \times n$ 次循环, 每次需要 m 次求和操作, 因此需要 $n^2 m$ 次操作. 在最差情况下, 所有节点将加入副本集合, 此时 While 循环将执行 $n - 1$ 次, 而 For 循环执行 $n - 2$ 次, For 循环执行体中计算 $\mathbf{AV}(i)$ 需要执行 m 次, 计算 $QoS_Certain(S)^i$ 需要执行 $2n + m$ 次, 共需要 $(n - 1) \times (n - 2) \times (m + 2n + m) = 2(m + n)(n - 1)(n - 2)$ 次操作. 因此, 算法的总体算法复杂度为 $\Theta((m + n)n^2)$. 考虑到一般情况下 $n \gg m$, 所以最坏情况下算法复杂度为 $\Theta(n^3)$.

4 实验结果与分析

为了模拟真实的网络环境, 采用著名的网络拓扑生成器 BRITE^[10] 产生具有不同数目节点的网络结构. BRITE 先将节点随机放置在一个长为 HS、宽为 LS 的矩形区域中, 然后在节点之间根据一个几率 p 随机地添加边, 最后为这些边赋予延迟和带宽等属性. 在下面的实验中, $HS = 10 * n$, $LS = n$, $p = 0.45$. 在 BRITE 产生的网络拓扑基础上, 抓取点和边的信息构成邻接矩阵 \mathbf{AM} , 并采用 Floyd-Warshall 算法从 \mathbf{AM} 计算得到距离矩阵 \mathbf{DM} . D 为整个网络拓扑的直径, 即所有节点之间的最远距离. Floyd-Warshall 算法的复杂度为 $\Theta(n^3)$, 在 n 较大的时候会带来很大计算开销. 但是数据网络的拓扑结构一般比较稳定, 且各节点之间的服务关系也比较稳定, 因此一次计算获得的 \mathbf{DM} 在较长的时间窗口内有效. 只有新的节点加入、或者旧节点退

算法 1: 满足 CQR 的副本放置算法 (\mathbf{BM} , \mathbf{QM})

```

{
  初始化副本集合: RepSet={src}; //src 为源节点编号
  For 每一个节点 i, 执行如下循环: //构建 DM
  {
    For 每一个节点 j, 执行如下循环:
       $d_{i,j} = \mathbf{QM}^i[i] \times \mathbf{WV} = \sum_{1 \leq j \leq m} q_{i,j}^k \times \lambda_j$ ; //根据 (4) 式计算
    }
  }
   $\mathbf{AV} = \mathbf{DM}[\text{src}]$ ; //初始化访问向量
   $QoS\_Certain(S) = \sum((\mathbf{AM} \ominus \mathbf{BM}) \times \mathbf{WV})$ ; //根据 (6) 式初始化 CQR
  While ( $QoS\_Certain(S) > 0$ ), 执行如下循环:
  {
    For 每一个非副本节点 k, 执行如下循环:
    {
       $\mathbf{AV}^k = \mathbf{AV} \oplus (\mathbf{QM}^k \times \mathbf{WV})$ ;
       $QoS\_Certain(S)^k = \sum((\mathbf{AV}^k \ominus \mathbf{BM}) \times \mathbf{WV})$ ;
      计算节点 Profit ( $k$ ) =  $QoS\_Certain(S) - QoS\_Certain(S)^k$ ;
    }
    从中选择 Profit 值最大的节点 cand,  $RepSet = RepSet \cup \{cand\}$ ; //加入副本集合
     $\mathbf{AV} = \mathbf{AV} \oplus (\mathbf{QM}^{cand} \times \mathbf{WV})$ ; //更新访问向量
     $QoS\_Certain(S) = \sum((\mathbf{AV} \ominus \mathbf{BM}) \times \mathbf{WV})$ ; //更新 CQR
  }
  返回 RepSet;
}

```

图 1 满足 CQR 的副本放置算法伪码

出引起网络拓扑变化时,数据网格管理服务才需要重新获取系统信息并重新计算 DM .

在试验中,将节点之间的几何空间距离作为 QoS 距离进行模拟,将 CQR 简化为节点之间距离与网络直径 D 之间的比值 q/D . 实验将研究在包含不同节点数目的数据网格拓扑中算法的性质,讨论不同 CQR(q/D) 和访问模式 (ξ) 对保障个体 QoS 所需的副本数量和访问总体开销的变化情况. 在实际操作中并不直接选取不同 q/D 和 ξ 进行多组实验,每组实验重复 10 次. 为了降低偶然性因素的影响,每次随机选择不同的节点作为数据源节点,最后计算平均值作为最终结果.

实验一:在保持 $q/D = 0.2$ 不变的情况下,研究副本节点数目占节点总数百分比的变化情况,测试结果见图 2. 可以看到,随着 n 的增加即网络规模的扩大,算法选取的副本节点数比例数反而逐渐减小. 这是因为每个副本节点将有机会服务于更多的节点. 本实验证明了算法具有很好的可扩展性.

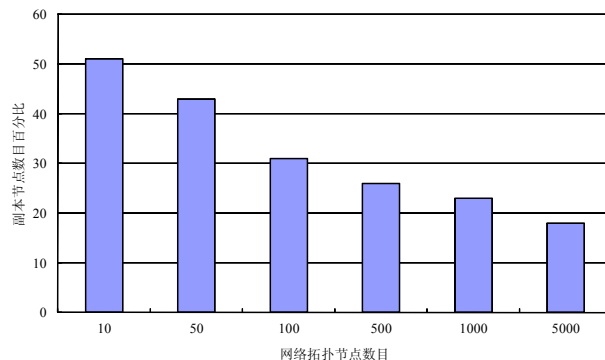


图 2 保持 q/D 一定时副本节点占全部节点百分比

实验二:在不同节点规模情况下,研究 q/D 对副本节点数目的影响,测试结果见图 3. 可以看到, q/D 对于副本节点的数目影响非常大. 随着 q/D 比例的增加,保障 CQR 所需的副本节点数目迅速减小. 当 $q/D = 0.5$ 时,只需要少数几个节点即可以满足个体服务质量.

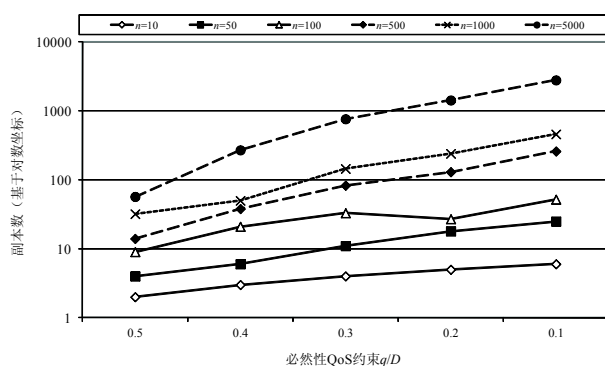


图 3 $\xi = 0.2$, q/D 与副本节点数目之间的关系

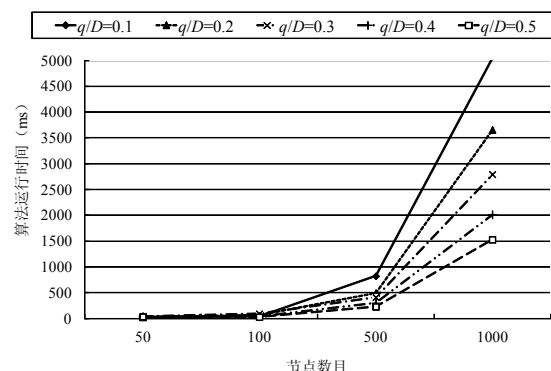


图 4 算法运行速度与节点数目的变化关系

实验三:在保持 q/D 不变的情况下,研究节点数目对于算法性能的影响,如图 4 所示. 实验中,固定 $\xi = 0.1$, 节点数目从 50 逐渐增加到 1000. 可以看到,算法的运行时间随着节点数目的增长而迅速增长. 例如,在节点数目为 1000, $q = 0.1$ 时,也仅为 5.053s. 另外还可以看到,在相同节点数目的情况下, q/D 比值越大,那么所需的副本节点数目越少,因此运行时间反而越短.

5 相关工作

作为副本技术的一个重要研究点,研究人员关于副本放置 (replica placement) 问题已经进行了广泛而深入的探讨. Wang 等^[11]将副本放置问题与负载均衡结合起来考虑,选择使访问负载分布最优的节点集作为放置策略. 但是所有这些研究都从全局的角度出发,重视对于数据网格总体性能的优化,只能支持粗粒度的 QoS 约束,缺乏更细粒度的控制. 最近的一些文献逐渐关注到这个问题. Xu 等^[9]提出一种 QoS 感知的副本放置问题,包括副本已知的放置和副本未知的放置,并证明了副本已知的放置问题是一个 NP 难问题. 文章同时提出 l -greedy-insert 和 l -greedy-delete 两种启发式算法,算法选择的副本集保证所有请求能在指定的时间内访问到数据的某个副本. 在此基础上,Jeon 等^[12]给出该问题的另一个 NP 难证明,并提出两个分布式

算法. 另外, Liu 等^[13] 提出另一种启发式算法解决副本已知的放置问题, 可用最小的代价获得副本的放置位置. 肖依等^[14] 从“分而治之”的思想出发, 重点解决超大规模网络拓扑条件下的 QoS 感知副本放置问题, 并采用副本环机制管理副本一致性更新. 与上述研究相比, 本文的研究更加注重从形式化的角度进行定义和研究, 并不具体规定 QoS 约束和副本开销的具体内容, 因此具有更为广泛的应用前景.

6 结束语

在数据网络中进行副本放置是实现副本管理的重要功能, 提供有 CQR 保障的数据服务是数据网格相关应用成功的关键. 本文在对 CQR 进行形式化定义和副本放置模型的基础上, 形式化界定了 CQR 满足的副本放置问题的内涵. 通过定义一系列的矩阵表示与矩阵操作, 实现了一种基于矩阵操作的副本放置算法. 实验与分析表明, 该算法具有可扩展性和稳定性等特点.

参考文献

- [1] Chervenak A, Foster I, Kesselman C, et al. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets[J]. *Journal of Network and Computer Applications*, 2000, 23: 187–200.
- [2] Lin Y F, Liu P F, Wu J J. Optimal placement of replicas in data grid environments with locality assurance[C]// *International Conference on Parallel and Distributed Systems*, Minneapolis, Minnesota: IEEE Computer Society Press, 2006: 465–474.
- [3] 杨胜文, 史美林. 一种支持 QoS 约束的 Web 服务发现模型 [J]. *计算机学报*, 2005, 28(4): 589–594.
Yang S W, Shi M L. A mode for web service discovery with QoS constraints[J]. *Chinese Journal of Computers*, 2005, 28(4): 589–594.
- [4] 胡春明, 怀进鹏, 沃天宇, 等. 一种支持端到端 QoS 的服务网格体系结构 [J]. *软件学报*, 2006, 6: 1448–1458.
Hu C M, Huai J P, Wo T Y, et al. A service oriented grid architecture with end-to-end QoS[J]. *Chinese Journal of Software*, 2006, 6: 1448–1458.
- [5] Mansouri Y, Monsefi R. Optimal number of replicas with QoS assurance in data grid environment[C]// *Proceedings of the 2nd Asia International Conference on Modelling & Simulation*, Kuala Lumpur: IEEE Computer Society Press, 2008: 168–173.
- [6] Rodolakis G, Siachalou S, Georgiadis L. Replicated server placement with QoS constraints[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2006, 17(10): 1151–1162.
- [7] Loukopoulos T, Lampsas P, Ahmad I. Continuous replica placement schemes in distributed systems[C]// *Proceedings of the 19th Annual International Conference on Supercomputing*, Boston, USA: IEEE Computer Society Press, 2005: 284–292.
- [8] Knutsson B, Lu H H, Xu W, et al. Peer-to-peer support for massively multiplayer games[C]// *Proceedings of INFOCOM 2004*, Hongkong: IEEE Computer Society, 2004: 1–12.
- [9] Tang X Y, Xu J. QoS-aware replica placement for content distribution[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2005, 16(10): 921–932.
- [10] Medina A, Lakhina A, Matta I, et al. BRITe: An approach to universal topology generation[C]// *Proceedings of the 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Cincinnati, USA: IEEE Computer Society Press, 2001: 346–353.
- [11] Wang H K, Liu P F, Wu J J. A QoS-aware heuristic algorithm for replica placement[C]// *Proceedings of the Grid Computing 2006*, Barcelona: IEEE Computer Society Press, 2006: 96–103.
- [12] Jeon W J, Gupta I, Nahrstedt K. QoS-aware object replication in overlay networks[C]// *Proceedings of the Global Telecommunications Conference*, San Francisco: IEEE Computer Society Press, 2006: 1–5.
- [13] Liu P F, Wu J J. Optimal replica placement strategy for hierarchical data grid systems[C]// *Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid*, Singapore: IEEE Computer Society Press, 2006: 1–4.
- [14] 肖依, 付伟, 卢锡城. 数据网格中服务质量感知的副本放置方法 [J]. *中国科学 F 辑*, 2009, 39(10): 1063–1071.
Xiao N, Fu W, Lu X C. QoS-awared replica placement techniques in data grid applications[J]. *Science China, Information Science*, 2010, 53(8): 1487–1496.