

对象存储系统形式化研究^{*}

谭支鹏 冯 丹

(华中科技大学计算机学院教育部信息存储重点实验室 武汉 430074)

摘 要 对象存储系统是近来兴起的存储领域研究热点,它改变了传统的存储模式,用对象接口取代了传统的块接口。作为下一代互联网存储模式的关键技术——对象存储,它的出现使得海量信息存储、数据异构灵活共享、存储系统的可扩展性、信息安全等问题迎刃而解。对象作为对象存储系统的基本单位,它的管理将是系统管理的关键。为了更好地研究对象存储系统对象的管理,本文对对象的形式化做了系统的研究。首先根据对象存储系统的实际给出了对象存储系统中对象的形式化描述;接着我们给出了对象存储系统的形式化描述;最后讨论了对对象存储系统中对象间的关系,并通过这些对象间的关系推导出一种对象存储系统模式。对象存储系统的形式化研究将有助于我们对对象管理做进一步研究。

关键词 对象,对象存储系统,形式化,继承

Study of Formalization for Object Storage System

TAN Zhi-Peng FENG Dan

(Key Laboratory of Data Storage System, Ministry of Education

School of Computer, Huazhong University of Science and Technology, Wuhan 430074)

Abstract Study of Object Storage System is a hotspot, it changes traditional storage pattern by replacing block-based interface with object-based interface. Object storage pattern is a key technology for next generation network storage pattern, it provides a best way to solve mass information storage, data sharing in heterogeneous environment, scalable of storage system, security of information etc.. At the same time, object is the base unit of the object storage system, it is key element for system management. In order to study management of object storage system, the paper gives systemic studying of object formalization. First, the paper gives formalization of object; then, gives formalization of object storage system; last, gives relations of objects and infers an object storage pattern by the relations of objects. Study of formalization for object storage system will help us to study object management in detail.

Keywords Object, Object storage system, Formalization, Inherit

1 引言

随着 Internet 网技术的不断发展以及计算机技术的不断发展,传统的信息存储模式以及存储容量已经无法满足现实应用的需求。理想的存储系统应该具有下列 4 个特征:跨平台数据共享、高性能、可扩展性及安全性^[1]。网络存储一度成为解决这些问题的最好办法,如直接附加存储(DAS)、附网存储(NAS)和存储区域网(SAN)等,但是这些网络存储技术均存在不同程度的缺陷,难以同时具备上述 4 个特征^[2]。因而构造一种基于对象的接口,它具有良好的扩展性能、可提供高速数据传输率且可缓解服务器负载的基于对象的海量存储系统,已经势在必行。

在对象存储系统中,最重要的概念就是对象,对象是对存储系统中客观实体的抽象和描述,每一对象均由描述实体的数据(属性)和施加于该实体之上的相关操作组成。对象存储系统的核心是将数据通路(数据读或写)和控制通路(元数据)分离,并且基于对象存储设备构建系统,每个对象存储设备具有一定的智能,能够自动管理其上的数据分布^[2]。因而对象

存储系统的关键元素即为对象,它是对象存储系统实现管理和运行的基本单位。这些对象通过组合可以形成类对象,同样的类对象也有自己的属性和操作,有关于对象的相关管理方法对它同样实用。因此对对象存储系统的研究关键是对对象管理的研究,以达到对象在系统中最合理、最有效的管理,为整个存储系统性能最优、效率最高打下坚实的基础。对象以及对象存储系统的形式化^[4,5]又是研究对象管理的突破口,因此本文将从对象的概念出发,对对象以及对象存储系统的形式化进行系统化的研究与分析。

2 对象形式化

一个对象通常具有属性、行为和操作接口。对象最显著的特征是类、封装、继承,既然提出了对象存储系统,我们就应该充分应用对象的这些特有的优点,来解决传统的存储系统无法解决的问题,实现存储系统的海量存储、高效存储、安全存储等等。

结合对象存储系统,我们先给出对象存在的几种类型的原子类:

^{*} 本文受国家重点基础研究发展计划项目 973 资助,项目名称:下一代互联网信息存储的组织模式和核心技术研究,项目编号:2004CB318201。
谭支鹏 博士研究生,主要研究兴趣信息存储、信息安全、数据库技术等;冯 丹 教授、博士生导师,主要研究兴趣信息存储、计算机系统结构、磁盘阵列等。

(1)原子设备对象(Device Object),记为 Φ 。它是对象存储系统中具体的一个个存储设备,这些设备可以是磁盘、磁盘阵列还可以是 SAN 或者 ANS。它的属性描述了该存储设备 SOSC 的特性,如总容量、最多可以存储的对象数、以及与数据完整性有关的属性(如该存储设备 SOSC 是作为 RAID 来实现,描述 RAID 的级别)。

(2)原子分区对象(Partition Object),记为 Ψ 。它包含一组用户对象,这些用户对象拥有共同的属性,分区对象的数据区只含用户对象 ID 列表,分区对象的属性包括分区内的用户对象数目,分区内用户对象占用的空间等。

(3)原子集合对象(Collection Object),记为 $\mathcal{L}$,它用于实现用户对象的快速检索,一个分区对象可以包含零个或多个集合对象,一个用户对象可以属于零个或多个集合对象。

(4)原子用户对象(User Object),记为 δ 。它是指需要在对象存储设备上存储的各类用户信息,可能是文件对象信息、数据库对象信息、流式文件对象等等,所有对象以 Partition-ID 和 User ID 标识。

定义 1 若 T 为原子类, $t \in T$ 且 $t \in \Phi \cup \Psi \cup \mathcal{L} \cup \delta$, 则 T 表示所有可能的对象组成的集合称为类 t 的域或直接简称为类,类 t 中的元素都称为原子对象。

在实际的对象存储系统中,设备对象、分区对象、集合对象、用户对象它们往往并不是以原子形态存在,相反多数情况下它们是以某种形式集合存在的,这是因为对象存储系统的需求多种多样,尤其是在网络对象存储系统中。对象存储系统中的对象一定存在着多种多样的引用关系,通常主要包括简单引用、依赖排它聚合引用、独立排它聚合引用、依赖共享聚合引用、独立共享聚合引用等 5 种引用关系。

我们也可以符合 BNF 范式的要求来给出对象的形式化定义。

定义 2 $\langle \text{对象} \rangle ::= \langle \text{原子对象} \rangle | \langle \text{类对象} \rangle | \langle \text{复杂对象} \rangle$
 $\langle \text{原子对象} \rangle ::= \langle \text{原子属性} \rangle | \langle \text{原子操作} \rangle$
 $\langle \text{类对象} \rangle ::= \{ \langle \text{原子属性} \rangle \} | \{ \langle \text{类属性} \rangle \} | \{ \langle \text{原子操作} \rangle \} | \{ \langle \text{类操作} \rangle \}$
 $\langle \text{复杂对象} \rangle ::= \{ \{ \langle \text{原子属性} \rangle \} \} | \{ \{ \langle \text{类属性} \rangle \} \} | \{ \{ \langle \text{聚合属性} \rangle \} \} | \{ \{ \langle \text{原子操作} \rangle \} \} | \{ \{ \langle \text{类操作} \rangle \} \} | \{ \{ \langle \text{复杂操作} \rangle \} \}$
 $\langle \text{原子属性} \rangle ::= \langle \text{对象标识} \rangle | \langle \text{对象大小} \rangle | \langle \text{对象指针} \rangle | \langle \text{对象状态} \rangle | \langle \text{对象访问频率} \rangle | \dots$
 $\langle \text{类属性} \rangle ::= \langle \text{类引用关系} \rangle | \langle \text{原子对象} \rangle$
 $\langle \text{聚合属性} \rangle ::= \langle \text{聚合引用关系} \rangle | \langle \text{聚合对象} \rangle$
 $\langle \text{聚合引用关系} \rangle ::= \langle \text{依赖排它聚合引用} \rangle | \langle \text{独立排它聚合引用} \rangle | \langle \text{依赖共享聚合引用} \rangle | \langle \text{独立共享聚合引用} \rangle$
 $\langle \text{聚合对象} \rangle ::= \langle \text{原子对象} \rangle$
 $\langle \text{聚合对象层次} \rangle ::= \langle \text{复杂对象} \rangle | \langle \text{类对象} \rangle$
 $\langle \text{聚合类层次} \rangle ::= \langle \text{聚合类} \rangle | \langle \text{组成类} \rangle$
 在定义中, $\{ \}$ 表示出现零次或多次, $|$ 表示“或”关系。

对象作为对象存储系统中的一个最基本的存储单位,根据对象的固有特征,我们用三元组的方法来表示对象存储系统中一个对象的最简单存在形式:

定义 3 对象 $O_i = (ID, \text{Attribute}, \text{Inherit})$, $i > 0$, $O_1, O_2, \dots, O_n$ 是有限个对象元。其中 ID 为对象的标识符,Attribute 为对象的属性集,由属性集我们可以知道对象的类别对象信息,inherit 为对象的继承关系集,通过继承关系可以形成大对象以及聚合对象。

3 对象存储系统的形式化定义

给出对象的形式化定义后,下面我们给出对象存储系统的形式化定义。

定义 3 一个对象存储系统我们用八元组来表示 $OS = \langle T, O, A, M, I, ID, f_1, f_2 \rangle$, 其中 T, O, A, M, I, ID 都是可数无限集:

(1) T 称为对象存储系统的类型集,它包括原子对象集和组合对象集两种类型的多种形式, T 上存在一个函数 $f: T \rightarrow T$ 。

(2) O 称为对象存储系统的对象集。其上存在一线序关系,而且有特定元素 NULL 表示无对象,有特殊对象集 AT-OM,称为原子对象集,其中任意元素属于原子类型集。

(3) A 称为对象存储系统的属性集,对象存储系统原子对象和聚合对象所具有的属性集合。

(4) M 称为对象存储系统的方法集,对象存储系统中对象所有操作方法的集合。

(5) I 称为对象存储系统的继承关系集,对存储系统对象之间存在的继承关系集合,并且存在一个一一映射关系 π ,它反映对象间的继承关系。

(6) ID 称为对象存储系统的对象标识符集,用来标识对象存储系统中一个个的对象。

(7) 现在如果令 $A' = A \cup I \cup M$, $O' = ID \cup O \cup T$, $D = O' \times A' \times O'$, $O^* = O' \cup O' \cup \dots$, 那么函数 $f_1: 2^D \times O \times ID \times O^* \rightarrow 2^D$ 称为对象存储系统的执行函数,函数 $f_2: 2^D \times O \times ID \times O^* \rightarrow O^*$ 称为对象存储系统的响应函数。

为了支持系统的继承关系,我们设定每个对象的所有直接父类型确定一个优先关系,用这个优先关系来处理系统中可能会出现冲突。这个映射关系为 $\pi()$,它属于继承关系集 I 。假设 $(t_1, \pi(n_1), t_2)$ 和 $(t_1, \pi(n_2), t_3)$ 同时出现,那么当 $n_1 < n_2$ 时,表明 t_2 和 t_3 都是 t_1 的直接父类,且 t_2 的优先于 t_3 。

4 对象间关系的形式化

对象存储系统中对象间的关系就是指对象间的继承关系。继承性是对象的一个非常重要而且作用巨大的特征。通过对象的继承关系我们很容易地形成能存储海量信息的对象存储系统,同时由于这种继承关系通过指针保持对象之间的引用关系,可以灵活实现对象在存储系中的分布存储,方便实现信息的共享、安全控制等等。

在对象存储系统中,我们定义对象的继承关系有单继承和多继承。所谓的单继承是指一个对象仅仅只继承了一个对象相关属性;而多继承是指一个对象同时继承了多个对象的相关属性。在这里我们将对对象存储系统中存在的所有继承关系,以及由继承关系形成的聚合对象进行形式化描述。

定义 4 单继承定义为 $O_i O_k [(1 \leq j \leq N, 1 \leq k \leq N) \wedge (j \neq k) \wedge (C_j \cap C_k = \emptyset)] \rightarrow O_i O_k [(C_j \subseteq C_k) \vee (C_k \subseteq C_j)]$, 即一个类的中的对象继承了另一个类中对象的相关属性和操作,其中 N 表示有 N 个类, C_i 表示类 i 例样的集合, C_j 表示类 j 例样的集合。

定义 5 多重继承定义为 $O_i O_j O_k [(O_i \in (C_j \cap C_k)) \wedge (1 \leq j \leq N, 1 \leq k \leq N) \wedge (k \neq j)] \leftrightarrow O_i O_j O_k \{ [O_i \subseteq C_j] \wedge [O_i \subseteq C_k] \wedge (1 \leq k, j \leq N) \wedge (k \neq j) \}$, 即类 i 中的对象继承了类 j 和类 k 中对象的相关属性和操作。其中 N 表示有 N 个类, C_i

表示类 i 例样的集合, C_j 表示类 j 例样的集合, C_k 表示类 k 例样的集合。

由于继承关系在对象存储系统中我们形成了各种不同类型的聚合对象,对这些继承关系的形式化定义如下:

定义 6 由单继承构成的含有聚合对象的对象存储系统,其结构是一个树 $R=(O, E)$, 其中:

- (1) O 是包括 O_ϕ (ϕ 类结点), O_ψ (ψ 类结点), O_δ (δ 类结点), O_ε (ε 类结点), O_α (α 类结点) 这些类的节点集;
- (2) $O_\phi, O_\psi, O_\delta, O_\varepsilon$ 是树的叶结点;
- (3) O_α 结点是有有一个或多个子结点的类结点;
- (4) E 是边的集合。

定义 7 由多重继承或多重继承与单继承共同构成的含有聚合对象的对象存储系统,其结构是一个图 $N=(C', E')$, 其中:

- (1) O' 是包括 O_ϕ (ϕ 类结点), O_ψ (ψ 类结点), O_δ (δ 类结点), O_ε (ε 类结点), O_α (α 类结点) 这些类的节点集;
- (2) $O_\phi, O_\psi, O_\delta, O_\varepsilon$ 是底层结点;
- (3) O_α 结点是有有一个或多个子结点的类结点;
- (4) E' 是边的集合。

定义 8 对象经过封装后成为类,类的继承关系我们可以形式化定义如下,设 A, B 为两个类, $A < B$ 称 A 继承 B 的性质:

- (1) 对原子类, $A < A$;
- (2) 对具有属性集 S_a, S_b 的元组类 A 及 $B, A < B \leftrightarrow \{P_x[x \in S_a][x \in S_b \wedge A.x < B.x]\}$;
- (3) 对具有操作行为集 O_a, O_b 的元组类 A 及 $B, A < B \leftrightarrow \{P_x[x \in O_a][x \in O_b \wedge A.x < B.x]\}$;
- (4) 对聚合类 A 和 $B, A < B \leftrightarrow \{P_a[a \in A]b[b \in B \wedge a < b]\}$;

有了对象、聚合对象、类以及它们之间的继承关系后我们可以得到对象存储系统的一种模式,如果 $S \in 2^D$, 且 S 满足以下条件,则我们可以称 S 就是一种对象存储系统模式:

- (1) 对任意的 $(t_1, a, t_2) \in S$ 都有 $t_1 \in T-B$, 且不存在 $t_3 \in T$ 使得 $t_1 = f(t_3)$;
- (2) 对任意的 $(t_1, a, t_2) \in S$ 都有 $a \in A \cup I \cup M$, 如果 $a \in M$, 则有 $t_2 \in ID$, 否则 $t_2 \in T$;
- (3) 对任意的 $(t_1, a_1, t_2) \in S$, 如果 $t \notin B \cup P$ 且不存在 $t_3 \in T$ 使得 $t_2 = f(t_3)$, 则必存在 $(t_2, a_2, t_2) \in S$, 如果存在 t_3 使得 $t_2 = f(t_3)$, 且 $t_3 \notin B$, 则必存在 $(t_3, a_2, t_2) \in S$;
- (4) 如果 $(t_1, a, t_2) \in S$, 则不存在 $t'_2 \neq t_2$ 使得 $(t_1, a, t'_2) \in S$;
- (5) 现在来定义函数 $F(S, t) = \{t' | (t, t') \in R(S)^+\}$ 其中定义域为: $2^D \times T$, 值域为: 2^T ; $R(S)^+ = \{(t_1, t_2) | \exists n(n \in N \wedge t_1, \pi(a), t_2) \in S)\}$, 是传递闭包, 对任意的 $(t_1, a, t_2) \in S$ 都有 $t_1 \notin F(S, t_1)$ 。

如果 S 是一种对象存储系统的模式, 那么有 $T(S) = \{t | (t, a, t') \in S\}$ 称为聚合对象的类型集。根据对象存储系统模式的定义, 在这里我们依然可以给出方法和属性的继承关系。根据假设, 显然有集合 $F(t_1, S)$ 是 S 中 t_1 类型的直接或者是间接超类型构成的集合, 在模式 S 中的复杂类型 t 都可以从它的超类型来继承方法和属性, 用两个函数 $Att(S, t)$ 和 $Met(S, t)$ 分别来定义一个类型的属性和方法的集合:

$$Att(S, t) = \{a | a \in A_1 \wedge ((t, a, t') \in S \vee \exists t_1((t_1, a, t_2) \in$$

$S \wedge t_1 \in F(S, t))\}$, 定义域为: $2^D \times T$, 值域为: 2^A 。

$Met(S, t) = \{m | m \in ID \wedge ((t, m, id) \in S \vee \exists t_1((t_1, m, id) \in S \wedge t_1 \in F(S, t))\}$, 定义域为: $2^D \times T$, 值域为: 2^M 。

结束语 随着高性能计算、网络技术的蓬勃发展, 现在社会已经基本上完全进入了计算机信息时代, 我们的工作和生活几乎都不能离开计算机了, 由此产生的数据信息急剧增长, 传统的硬盘存储已经越来越无法满足信息存储的需要。现在人们已经开始重视信息存储系统建设的解决方案, 计算机时代也随之进入以数据存储为中心的存储时代, 以数据存储为代表的第三次计算机产业已经来临。事实上, 近几年存储技术的发展也很快, 如 RAID 技术以及网络 RAID 技术进一步完善, 逐步成为主流的海量信息存储解决方案。同时顺应应用的需求, 还出现了网络存储技术的研究, 如 NAS、SAN 等, 也已经有一些相关的产品应用于生产一线, 取得了巨大的经济效益。然而在异构平台共享、系统性能、信息安全、系统的可扩展性等方面人们提出了更高的要求, 为了更好地来解决这些问题, 研究者们开始了对对象存储的研究。为了系统地深入研究对象存储系统, 我们在深刻认识对象技术以及对象特征的基础上, 并结合对象存储系统的实际, 对对象存储系统的进行了形式化研究, 这对我们进一步地分析和研究对象存储系统将起到很大的帮助作用。

参考文献

- 1 Braam P J. The Lustre Storage architecture[EB/OL]. <http://www.lustre.org/docs/lustre.pdf>, 2003-03
- 2 Mesnier M, Mellon C, Gregory I, et al. Seagate Research, Object-Based Storage. IEEE Communications Magazine, 2003, 84~90
- 3 Chen DDS. The entity-relational model towards a unified view of data. ACM TODS, 1986, 1(1): 255~265
- 4 Atkinson M, et al. The object-oriented system manifesto. In: Proc DOOD'97, MA, USA, 1998. 823~828
- 5 Khoshafian S, Coperland G. Object identity. In: Proc OOPSLA'86, 1986. 406~416
- 6 Zanio lo C, et al. Object-oriented database systems and knowledge systems. In: Proc from the Nine Int'l Workshop, CA, USA, 1994. 27~31
- 7 Kim W, et al. Operations and implementation of complex objects. IEEE Trans on Software Engineering, 1988, 14(7): 211~218
- 8 Abiteboul S, Hull R. New approach towards an object-oriented database system; [Tech Rep]. 502. MIT 1997
- 9 Wu JK. An object-oriented approach to picture database. In: Proc of ACM 2 SIGMOD96, New York, 1996. 450~453
- 10 Kiwi W. Introduction to object-oriented Database. MA: MIT Press, 1990
- 11 Mariani J A. Oggetto, an object oriented database layered on a triple store. The computer journal, 1992, 35(2): 108~118
- 12 Deux O, et al. The story of O2. IEEE Trans on Knowledge and Data Engineering, 1990, 2(1): 208~232
- 13 Woelk D, Kim W, Luther W. An object-oriented approach to multimedia database. In: Proc of ACM 2 SIGMOD96, New York, 1996. 268~275
- 14 尹七春, 柳诚飞, 邵志清. 面向对象数据模型的形式化描述. 计算机学报, 1995, 18(7)