

大规模动态发布/订阅系统中的重配置管理

陈金辉¹ 董 彪²

(南京信息工程大学信息与控制学院 南京 210044)¹

(江苏智能传感器网络工程技术研究开发中心 南京 210023)²

摘要 重配置特性使得发布/订阅 P/S(Publish/Subscribe)系统适应高度动态的环境,现有的 P/S 系统通常不能动态地改变既定的操作,适应影响事件代理框架的拓扑结构的变化。提出了一种大规模动态 P/S 系统中重配置管理的新途径——RS3DS(reconfigurable sparse 3 dimensional space)模型。基于 RS3DS 中代理拆分和合并过程,分析了 RS3DS 中订阅表的重配置策略。结果表明 RS3DS 具有低的重配置开销和高的事件交付率。

关键词 发布/订阅,重配置管理,订阅表

中图分类号 TP393 文件标识码 A

Reconfigurable Management for Large Scale Dynamic Publish/ Subscribe Systems

CHEN Jin-hui¹ DONG Biao²

(School of Information & Control, Nanjing University of Information Science & Technology, Nanjing 210044, China)¹

(Jiangsu Research & Development Center of Intelligent Sensor Network Engineering & Technology, Nanjing 210023, China)²

Abstract Reconfigurability makes P/S (Publish/Subscribe) systems amenable to highly dynamic environments. Existing P/S systems are typically not able to rearrange dynamically their operations to adapt to changes which impact the topology of their event broker infrastructure. A new reconfigurable management approach called RS3DS(reconfigurable sparse 3 dimensional space) was proposed in large-scale dynamic P/S systems. A reconfiguration strategy for subscription tables was analyzed, which was based on the notion of composition and decomposition from RS3DS. The simulation results show RS3DS has low reconfiguration overhead and high event delivery rate.

Keywords Publish/Subscribe, Reconfigurable management, Subscription table

1 引言

P/S 范型具有多点通信、异步的特点,使参与通信的发布者和订阅者在控制流、时间和空间 3 方面完全解耦,能够很好地适应大规模、高度动态的计算环境。P/S 系统是一种以 P/S 范型进行交互的中间件。在 P/S 系统中,用事件来表示发布者和订阅者之间所交互的信息。发布者向 P/S 系统发出事件,订阅者向 P/S 系统发送订阅条件,表示对哪些事件感兴趣。本质上,P/S 系统是一个事件代理的网络,事件代理负责事件的匹配和路由。P/S 系统中的路由也称为基于内容的路由 CBR(Content-Based Routing)。可重配置的 P/S 系统是指 P/S 系统支持事件代理框架所对应的拓扑结构的动态重配置。新型高效的、面向大规模的事件代理框架是近年来的主要研究方向,各种类型的 P/S 系统被提出,这些研究大都在事件路由方面具有较高的效率。但是,在面对高度动态变化的网络环境时,现有的 P/S 系统在维护拓扑结构、提高订阅效率等方面尚存在着较多的问题。

重配置管理与网络的拓扑结构紧密相关,不同的拓扑结

构可使用的 CBR 算法差别很大。按照 P/S 系统中的 CBR 算法,重配置管理分为基于静态模式的和基于结构化 P2P 模式的两种方法。静态模式指由多个分布在 Internet 上的代理组成的覆盖网络一旦形成以后,其拓扑结构几乎不会发生变化,本质上是静态的^[1,2]。结构化 P2P 网络,如 CAN、Chord、Pastry 等,具有良好的自组织性、容错性和扩展性,适合作为 P/S 系统的事件路由的基础设施。目前,出现了多种基于结构化 P2P 网络的 P/S 系统,如 Siena^[3]、PSIRP^[4]、PAPs^[5]、Hermes^[6]、P2P-ToPSS^[7]、PUB-2-SUB^[8] 和 StreamHub^[9] 等。这些系统的 CBR 算法都构建在 P2P 路由层之上,常见的有生成树转发和基于聚点的 CBR 算法两类。生成树转发算法与静态网络上基于过滤的算法相似,主要区别在于它们利用网络的路由机制构造事件分发树。例如,Terpstra 等把 Chord 的路由机制同基于内容的过滤结合起来,网络中的每个节点都有一棵以自己为根的生成树,节点在此树上发布事件,但是由于一个节点同时属于所有其他节点以自己为根的生成树,节点需在所有生成树中向根节点传播订阅,因此,在重配置管理时,订阅表的维护成本很高。基于聚点的 CBR 算法由以下 3

到稿日期:2013-05-06 返修日期:2013-07-28 本文受 2012 年江苏省“青蓝工程”项目,江苏省智能传感器网络工程技术研究开发中心开放基金(ZK11-02-01),中国气象局 2013 年度气象软科学项目([2013]第 68 号)资助。

陈金辉(1970—),女,副教授,主要研究方向为系统分析与集成,E-mail:cjh@nuist.edu.cn;董 彪(1969—),男,博士,副教授,主要研究方向为分布式计算、计算机网络。

个步骤组成:传播订阅到聚点;发布事件到聚点;聚点进行事件匹配并分发事件到感兴趣的订阅者。此类算法的关键在于聚点的选择,困难在于如何解决 DHT 映射精确性与 P/S 系统的数据模型复杂性之间的矛盾。文献[10-12]采取了对属性值进行映射的方法,这些方法的缺点是难以处理订阅中的复杂过滤条件,比如数值范围约束、前缀后缀约束等。文献[13]中提到为域属性集合中的每个属性分配一个聚点,订阅条件存储到离订阅者最近的聚点,但事件要被发送到所有的聚点上,从而产生了对所有的聚点的重配置开销。

本文从 CBR 算法必须保证事件和订阅至少在一个节点相交这样的基本原理出发,提出了一种大规模动态 P/S 系统重配置管理的新途径 RS3DS。在 RS3DS 中,事件、订阅分别在事件分区、订阅分区中路由,任何事件分区和订阅分区有且只有一个交点。为保证 P/S 系统的高效性,事件分区之间、订阅分区之间均不相交。与已有的研究相比较,RS3DS 有如下特点:

1)控制订阅表重配置的开销。RS3DS 中的每一个节点均能扮演一个聚点,围绕该聚点,订阅条件在聚点所在的订阅分区中传播。由于聚点具有随机分布的特性,因此减少了路由的瓶颈,使系统具有较好的可扩展性,能在大规模的网络上进行 CBR 的订阅表重配置。

2)通过调节 RS3DS 中事件、订阅分区的大小,改变节点的订阅、事件负载,来适应具有不同流量特点的应用场景。

2 RS3DS 模型

RS3DS 模型通过事件和订阅在不同的分区中传播来实现 P/S。设 $Q = \{(x, y, z) | x, y, z \text{ 取整数}, 1 \leq x \leq CX \wedge 1 \leq y \leq CY \wedge 1 \leq z \leq CZ, CX, CY, CZ \text{ 为整数}\}$, Q 为三维空间中的节点的集合。订阅分区 S_i 由 Q 中 $z = Z_i$ 的节点组成,事件分区 $\epsilon_{i,k} = \{(x_j, y_k, z) | 1 \leq z \leq CZ\}$ 。可得, $S_i \cap \epsilon_{i,k} = \{(x_j, y_j, z_i)\}$,表明每个订阅分区与事件分区有且只有一个交点,保证事件在每一个订阅分区中的传播。

设 $CX=4, CY=4$ 和 $CZ=4$, Q 中的每个事件分区有 4 个节点,共有 16 个事件分区,每个订阅分区有 16 个节点,共有 4 个订阅分区。下文把订阅分区 S_i 的第一个订阅条件记为 sp_i 。图 1 所示为订阅 sp_i 在 S_i 上的路由。设发出订阅 sp_i 的节点为 $(2, 3, Z_i)$,当 sp_i 在 S_i 中传播时,首先, S_i 中的其余节点先沿着 y 轴方向分别建立起指向节点 $(1, 3, Z_i)$ 、 $(2, 3, Z_i)$ 、 $(3, 3, Z_i)$ 和 $(4, 3, Z_i)$ 的路由。然后,再建立节点 $(1, 3, Z_i)$ 、 $(3, 3, Z_i)$ 和 $(4, 3, Z_i)$ 到 $(2, 3, Z_i)$ 的路由。在图 1 中,用虚线箭头表示指向订阅者的路由。如果有其他节点 $(2, 4, Z_i)$ 、 $(3, 1, Z_i)$ 和 $(4, 4, Z_i)$ 订阅 sp_i ,此时,只需要沿着节点 $(2, 3, Z_i)$ 已经建立的路由的相反方向,建立起节点 $(2, 4, Z_i)$ 到对应节点的路由。不失一般性,上述方法对 S_i 中任何第一个发出订阅的节点均成立。如果其他订阅分区 S_m 中第一次发出订阅的节点是 (x, y, Z_m) ,那么,该订阅在 S_m 中独立地传播,建立起指向 (x, y, Z_m) 的路由。

S_i 中陆续到达的新订阅的路由是建立在第一个订阅路由基础之上的,而不需为每个订阅反复建立相应的独立路径。例如,图 1 中的节点 $(3, 2, Z_i)$ 发出订阅 sq , sq 首先经节点 $(3, 3, Z_i)$ 传播到第一个发出订阅的节点 $(2, 3, Z_i)$,再沿着该节点为订阅 sp 生成的路径传播订阅 sq ,在所有经过的节点订阅表

中插入订阅 sq 。

设在事件分区 $\epsilon_{i,2}$ 中发布了与订阅 sp_i 相匹配的事件 e ,事件 e 经节点 $(4, 2, Z_i)$ 进入 S_i ,事件 e 分别沿着 $(4, 3, Z_i) \rightarrow (3, 3, Z_i) \rightarrow (2, 3, Z_i)$ 、 $(4, 3, Z_i) \rightarrow (3, 3, Z_i) \rightarrow (2, 3, Z_i) \rightarrow (2, 4, Z_i)$ 、 $(4, 3, Z_i) \rightarrow (3, 3, Z_i) \rightarrow (3, 2, Z_i) \rightarrow (3, 1, Z_i)$ 、 $(4, 3, Z_i) \rightarrow (4, 4, Z_i)$ 这 4 条路径到达事件 e 的 4 个订阅节点 $(2, 3, Z_i)$ 、 $(2, 4, Z_i)$ 、 $(3, 1, Z_i)$ 和 $(4, 4, Z_i)$ 。在本文的图中,用空心圆圈表示订阅节点,为简明起见,省略节点的 Z_i 值。

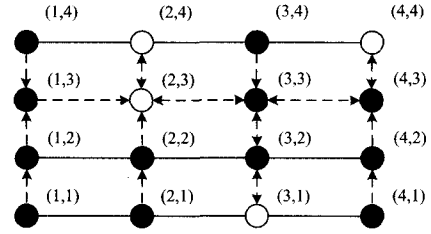


图 1 订阅 sp_i 在 S_i 上的路由

定义 1(RS3DS 模型) 基于三维空间事件、订阅分区之上的稀疏覆盖网称为 RS3DS 模型。

3 RS3DS 重配置管理

3.1 代理的定义

在 RS3DS 中,代理依据其包括的节点来标识其在三维空间中的位置。对于任意一个 S_i ,当第一次有代理进入时,系统把 S_i 中的所有节点分配给该代理,随着系统的运行,新加入的代理动态地将 S_i 划分为多个区域,每个代理负责对应区域内所含节点的订阅传播和事件路由。

定义 2(区域) S_i 的一个区域 $A_{i,j}$ 是指 S_i 中一组节点的集合,对于该集合中的任意一个节点 $a(x_a, y_a, z_i)$,存在该集合中的另一节点 $b(x_b, y_b, z_i)$,使得节点 a 和节点 b 之间的距离 $d = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} = 1$ 。

定义 3(指定节点) 设 S_i 中的节点 a 发布了 sp_i ,从而建立起指向节点 a 的关于 sp_i 的路由信息。对于 S_i 的一个区域 $A_{i,j}$ 和节点 $b \in A_{i,j}$, $A_{i,j}$ 中的其余节点都有到节点 b 的关于 sp_i 的路由,称节点 b 是 $A_{i,j}$ 的指定节点,记为 $b = DN(A_{i,j})$ 。

定义 4(代理) 对于 S_i 和 $A_{i,j}$ ($1 \leq j \leq k, k, j$ 是整数),满足条件: $S_i = \bigcup_{j=1}^k A_{i,j}$ 和 $A_{i,m} \cap A_{i,n} = \{\}$, $m \neq n, 1 \leq m, n \leq k$,称 $A_{i,j}$ 为 S_i 的一个代理,记为 $B_{i,j}$ 。

定义 5(前驱和后继) 对于 S_i 中的两个代理 $B_{i,m}$ 和 $B_{i,n}$,设 $a = DN(B_{i,m})$ 、 $b = DN(B_{i,n})$,若 b 有指向 a 的关于订阅 sp 的边,则称 $B_{i,m}$ 是 $B_{i,n}$ 的前驱,或者 $B_{i,n}$ 是 $B_{i,m}$ 的后继。

3.2 拆分和合并

定义 6(拆分和合并) 对于代理 $B_{i,l}$ 、 $B_{i,m}$ 和 $B_{i,n}$,满足条件: $B_{i,l} = B_{i,m} \cup B_{i,n}$; $B_{i,m} \cap B_{i,n} = \{\}$,则称 $B_{i,l}$ 拆分为 $B_{i,m}$ 和 $B_{i,n}$, $B_{i,m}$ 和 $B_{i,n}$ 合并为 $B_{i,l}$ 。

性质 1 代理 $B_{i,m}$ 和 $B_{i,n}$ 合并的前提为: $B_{i,m}$ 和 $B_{i,n}$ 具有前驱和后继(或者后继和前驱)关系。

性质 2 设 $B_{i,l}$ 的前驱代理为 $B_{i,p}$ ($B_{i,p}$ 可为空),后继代理集为 $Set(B_{i,c})$,把 $B_{i,l}$ 拆分成 $B_{i,m}$ 和 $B_{i,n}$, $B_{i,m}$ 是 $B_{i,n}$ 的前驱。那么,代理间前驱和后继关系的调整有且仅有以下两种情形:

- 1) $Set(B_{i,c})$ 作为 $B_{i,n}$ 的后继代理;
- 2) $Set(B_{i,c})$ 作为 $B_{i,m}$ 的后继代理。

性质 3 设 $B_{i,m}$ 的前驱代理为 $B_{i,p}$ ($B_{i,p}$ 可为空), 后继代理集为 $Set(B_{i,c1})$, $B_{i,n}$ 的后继代理集为 $Set(B_{i,c2})$ 。 $B_{i,m}$ 是 $B_{i,n}$ 的前驱, $B_{i,m}$ 和 $B_{i,n}$ 合并成 $B_{i,l}$ 后, 代理间前驱和后继关系的调整有且仅有以下两种情形:

- 1) $B_{i,p}$ 、 $Set(B_{i,c2})$ 中的代理分别为 $B_{i,l}$ 前驱、后继代理;
- 2) $B_{i,p}$ 为 $B_{i,l}$ 的前驱代理, $Set(B_{i,c1})$ 和 $Set(B_{i,c2})$ 中的代理为 $B_{i,l}$ 的后继代理。

限于篇幅, 性质 1—性质 3 的证明略。

根据性质 2、性质 3, 代理拆分和合并后, 前驱和后继关系的调整如图 2 所示, 图 2(a)、(b) 中 $a = DN(B_{i,m})$, $b = DN(B_{i,n})$, 图 2(c)、(d) 中 $a = DN(B_{i,l})$ 。

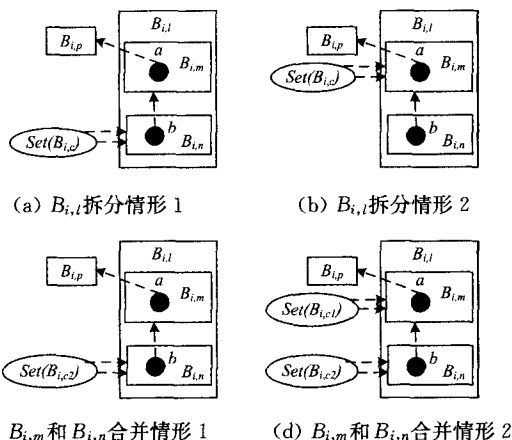


图 2 $B_{i,l}$ 、 $B_{i,m}$ 和 $B_{i,n}$ 拆分和合并后的前驱、后继调整

3.3 代理的重配置

当一个代理加入 RS3DS 时, 新加入的代理向其相邻节点发送包含地址的事件, 该过程对应于代理的拆分。 当一个代理离开 RS3DS 时, 可选择该代理的前驱或者后继代理来接管该代理, 本文使用后继代理来接管该代理, 该过程对应于代理的合并。

3.4 订阅表的重配置

当代理离开 RS3DS 时, 涉及订阅表的重配置。 本文用 $PPSet(B_{i,l})$ 表示 $B_{i,l}$ 的前驱和后继代理的集合。

定义 7(相邻节点集) 与 $B_{i,m}$ 中的节点直接相邻的 $PPSet(B_{i,l})$ 中节点的集合, 称为代理 $B_{i,m}$ 的相邻节点集, 记为 $NNode(B_{i,m})$ 。

定义 8(对应节点) 设 $x_1, x_2 \in NNode(B_{i,m})$, 对于订阅 p , x_1 有指向 $B_{i,m}$ 的路由, x_2 有远离 $B_{i,m}$ 的路由, 则称 x_2 为 x_1 的关于 $B_{i,m}$ 和 p 的对应节点, 记为 $CN(B_{i,m}, p) = (x_1, x_2)$ 。

定义 9(订阅表的重配置) 当 $B_{i,m}$ 离开 RS3DS 后, 设 $x_1 \in NNode(B_{i,m})$ 和 $NNode(B_{i,m})$ 中节点订阅了 p , 基于对应节点的订阅表重配置分为两种情形:

1) 若存在 x_2 , 使得 $CN(B_{i,m}, p) = (x_1, x_2)$, 则在 $P(x_1, b)$ 和 $P(b, x_2)$ 上建立关于 p 的路由(除节点 x_1 和 b)。 其中 $P(x, y)$ 表示在 S_i 中第一个订阅节点建立的订阅路由之上的从 x 到 y 的路径。

2) 若不存在 x_2 , 使得 $CN(B_{i,m}, p) = (x_1, x_2)$, 则在 S_i 中取消订阅 p 。

定义 10(订阅表的正确性) 对于任意的事件 e , 订阅表保证订阅者接收到其订阅的事件 e 。

定理 1 对于 S_i , 基于对应节点的订阅表重配置保证 S_i 中订阅表的正确性。

证明: 根据性质 3 及订阅者位置, 订阅表重配置分 5 种情形, 如表 1 所列。

表 1 订阅表重配置的情形

订阅者	需修改的路径	对应节点
$B_{i,n}$	$P(df, da), P(da, db)$	(df, db)
$B_{i,n}$	$P(dk, da), P(da, db)$	(dk, db)
经 $B_{i,p}$	$P(db, da), P(da, df)$	(db, df)
经 $B_{i,p}$	$P(dk, da), P(dm, df)$	(dk, df)
经 $Set(B_{i,c1})$	$P(df, da), P(da, dk)$	$(df, dk1)$
经 $Set(B_{i,c1})$	$P(db, da), P(da, dk)$	$(db, dk1)$
经 $Set(B_{i,c2})$	$P(df, da), P(da, db)$	(df, db)
经 $Set(B_{i,c2})$	$P(dk, da), P(da, db)$	(dk, db)
只有前 $B_{i,m}$	无路径	无对应节点

表 1 中 $B_{i,k}$ 为 $Set(B_{i,c1})$ 中的任意代理, $df = DN(B_{i,p})$, $dk = DN(B_{i,k})$, $dk_1 = DN(B_{i,c1})$ 。 经验证: 基于对应节点的订阅表重配置满足定义 10。

图 3 描述了代理离开 RS3DS 后, 相关节点的订阅表重配置过程。 虚线框表示代理, 当代理离开 S_i 后, 相应区域内的节点用斜线填充。 图 3(a) 是 S_i 的一种初始配置, 节点 $(2, 3, Z_i)$ 、 $(2, 4, Z_i)$ 、 $(3, 1, Z_i)$ 订阅 sp_i , 节点 $(1, 1, Z_i)$ 发布事件 e , 当 e 与 sp_i 匹配时, 事件路由仅涉及 $B_{i,1}$ 、 $B_{i,2}$ 两个代理。 图 3(b)–(f) 为一些代理离开 S_i 后, 相关节点的订阅表重配置的过程。

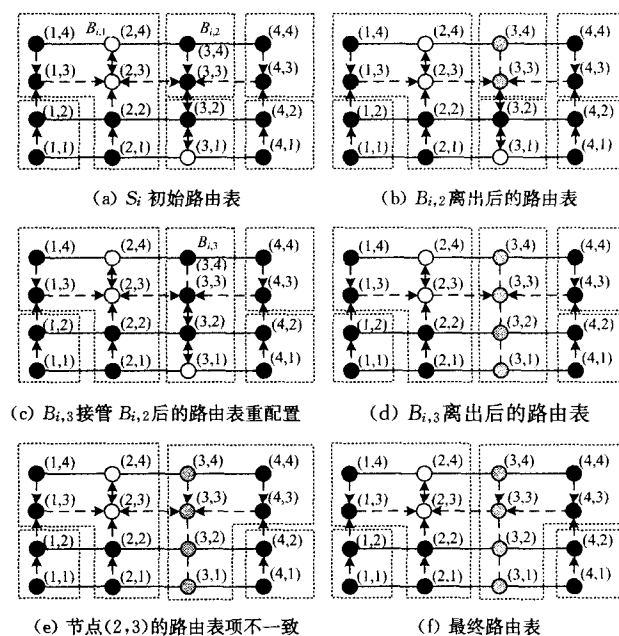


图 3 订阅表重配置过程

4 仿真实验

使用 PeerSim 构建仿真环境, 通过代理周期性离开, 评估订阅表重配置性能。 实验测试 50 次, 求其平均值作为实验值。

定义事件交付率 $Div(e) = T_{scale}(e) / T_{exp}(e)$, 其中 $T_{scale}(e)$ 、 $T_{exp}(e)$ 表示 RS3DS 中实际传送的事件数、预期传送的事件数。 参数设置为: $CX = 16$, $CY = 16$ 和 $CZ = 256$, 则每个事件分区有 256 个节点, 共有 256 个事件分区, 每个订阅分区有 256 个节点, 共有 256 个订阅分区, RS3DS 中的节点总数为 65536。 图 4 中横坐标表示重配置周期, 纵坐标表示事件交付率。

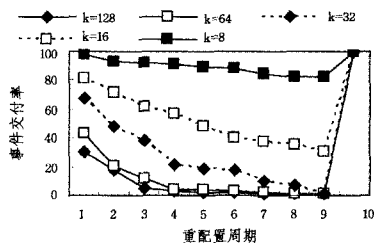


图4 代理离开时的事件交付率

在一个重配置周期, k 个代理离开订阅分区, k 从 8 变化到 128, k 占订阅分区中代理的比例从 3% 到 50%。从图 4 可以看出: 当离开的代理所占比例较小时, 有较高的 $Div(e)$, 当没有代理离开时, 又可保证事件的交付。这表明订阅表的重配置策略是可行的、高效的。

结束语 建立了在三维空间中订阅和事件在不同的分区中传播的 RS3DS 模型。提出了基于对应节点的订阅表重配置策略。仿真实验表明: RS3DS 具有低的重配置开销和高的事件交付率, 提高了大规模动态 P/S 系统的重配置效率。

参考文献

- [1] Jafarpour H, Hore B, Mehrotra S, et al. CCD: Efficient customized content dissemination in distributed publish/subscribe[C]// Proceedings of the 10th ACM/IFIP/USENIX international conference on middleware. New York: Springer-Verlag, 2009: 62-82
- [2] Tarkoma S. Dynamic content-based channels: meeting in the middle[C]// Proceedings of the 2nd international conference on distributed event-based systems. New York: ACM Press, 2008: 47-58
- [3] Keeney J, Roblek D, Jones D, et al. Extending Siena to support more expressive and flexible subscriptions[C]// Proceedings of the 2nd international conference on distributed event-based systems. New York: ACM Press, 2008: 35-46
- [4] Fotiou N, Trossen D, Polyzos G C. Illustrating a publish-subscribe internet architecture[J]. Telecommunication Systems, 2012, 51(4): 233-245
- [5] Ahmed N, Linderman M, Bryant J. PAPA S: peer assisted publish and subscribe[C]// Proceedings of the 7th Workshop on Middleware for Next Generation Internet Computing. New York: ACM Press, 2012: 7
- [6] Borsje J, Levering L, Frasinca F. Hermes: A semantic Web based news decision support system[C]// Proceedings of the 2008 ACM symposium on applied computing. New York: ACM Press, 2008: 16-20
- [7] Middleware Systems Research Group at the University of Toronto. ToPSS-Toronto publish/subscribe system project [EB/OL]. <http://www.eecg.toronto.edu/~jacob-sen/topss.html>, 2009-11-2
- [8] Tran D A, Pham C. Enabling content-based publish/subscribe services in cooperative P2P networks[J]. Computer Networks, 2010, 54(11): 1739-1749
- [9] Barazzutti R, Felber P, Fetzer C, et al. StreamHub: a massively parallel architecture for high-performance content-based publish/subscribe[C]// Proceedings of the 7th ACM international conference on Distributed event-based systems. New York: ACM Press, 2013: 63-74
- [10] Baldoni R, Beraldi R, Quema V. TERA: topic-based event routing for peer-to-peer architectures[C]// Proceedings of the 2007 inaugural international conference on distributed event-based systems. New York: ACM Press, 2007: 2-13
- [11] Pujol-Ahullo J, Garcia-Lopez P, Gomez-Skarmeta A F. Towards a lightweight content-based publish/subscribe services for peer-to-peer systems[J]. International Journal of Grid and Utility Computing, 2009, 1(3): 239-251
- [12] Garcia-Sanchez A J, Garcia-Sanchez F, Pavon-Marino P, et al. An optimised publish/subscribe middleware for peer-to-peer systems[J]. International Journal of Communication Networks and Distributed Systems, 2008, 1(2): 195-215
- [13] Zhu Y W, Hu Y M, Ferry: A P2P-based architecture for content-based publish/subscribe services[J]. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(5): 672-685
- (上接第 119 页)
- [11] Wang Jun, Zhu Hui-jun, Li Dong. eRAID: Conserving Energy in Conventional Disk-Based RAID System[J]. IEEE Transactions on Computers, 2008, 57(3): 359-374
- [12] Weddle C, Oldham M, Qian Jin, et al. PARAID: A gear-shifting power-aware raid [J]. ACM Trans. Storage, 2007, 3(3): 13
- [13] Zhu Qing-bo, Shankar A, Zhou Yuan-yuan. PB-LRU: A Self-Tuning Power Aware Storage Cache Replacement Algorithm for Conserving Disk Energy[C]// Proceedings of the 18th annual international conference on Supercomputing. Malo: ACM Press, 2004: 79-88
- [14] Pinheiro E, Bianchini R. Energy Conservation Techniques for Disk Array-Based Servers[C]// 2004 International Conference on Supercomputing. Saint-Malo: Association for Computing Machinery, 2004: 68-78
- [15] Yao Xiao-yu, Wang Jun. RIMAC: a novel redundancy-based hierarchical cache architecture for energy efficient, high performance storage systems [J]. SIGOPS Oper. Syst. Rev, 2006, 40(4): 249-262
- [16] Satoshi H, Hanh L H, Haruo Y. A power saving storage method that considers individual disk rotation[C]// DASFAA 2012. Busan: Springer Verlag, 2012: 138-149
- [17] Zhu Qing-bo, Chen Zhi-feng, Tan Lin, et al. Hibernator: helping disk arrays sleep through the winter [J]. SIGOPS Oper. Syst. Rev., 2005, 39(5): 177-190
- [18] Colarelli D, Grunwald D. Massive arrays of idle disks for storage archives[C]// Proceedings of the 2002 ACM/IEEE conference on Supercomputing. Baltimore: IEEE Computer Society Press, 2002: 1-11
- [19] 郝明哲. 视频点播系统关键技术研究[D]. 天津: 天津大学, 2006
- [20] 胡乐尧, 冯美禄, 赵继梅. 多媒体文件点播频率的分析和预测[J]. 南阳师范学院学报, 2007, 6(6): 38-40
- [21] 《2010 年中国网民网络视频应用研究报告》核心数据[EB/OL]. <http://it.people.com.cn/GB/119390/118340/212787/212790/13763203.html>, 2013-5
- [22] 乐天, 译. 用 MAID 2.0 降低存储费用[N]. 计算机世界, 2009-01-12(037)
- [23] Schulz G. MAID 2.0: Energy Savings without Performance Compromises [J]. the StorageIO Group, 2008