基于对等计算的分布式时空索引模型建立与整体框架研究 *

张 翀, 肖卫东, 杨晓亮
(国防科学技术大学 信息系统工程重点实验室, 长沙 410073)

摘 要: 提出了采用 P2P 技术建立分布式时空索引,从这一理论的基础出发,研究对等环境下分布式时空对象模型和索引的整体框架。模型和架构涵盖了面向历史的索引和面向将来预测的索引技术。模型以空间划分作为时空数据分片的依据,描述了划分的空间、服务器节点以及时空对象之间的关系;整体架构是一种双层结构,满足分布式环境下索引的新需求,架构包含索引查询、建立和维护的基本协议。以三种应用情景为例研究模型的应用范围,最后在自行开发的系统上验证所提出的整体框架的性能。实验表明,所提出的整体框架能够满足分布式环境下对时空数据的存取要求。

关键词: 时空索引; 分布式; P2P; 模型; 框架
中图分类号: TP391 **文献标志码**: A **文章编号**: 1001-3695(2012)03-0961-07
doi:10.3969/j.issn.1001-3695.2012.03.044

Modeling and framework of P2P-based distributed spatio-temporal index

ZHANG Chong, XIAO Wei-dong, YANG Xiao-liang
(*Key Laboratory of Science & Technology on Information Systems Engineering, National University of Defense Technology, Changsha* 410073, *China*)

Abstract: This paper creatively proposed P2P-based spatio-temporal index. It studied how to model this problem and how to build whole framework. And the model and framework covered historical index and future index. The model was based on space division, and described space partition, service node and spatio-temporal objects. The whole framework was two-tier structure, and was able to support new request under distributed environment. It consisted of querying, building and maintaining protocals. And then used 3 cases to show valid of the model. Finally, it studied the performance of the whole framework on the system developed by authors. The results show the frame is able to fullfil the access demond in distributed environment.
Key words: spatio-temporal index; distributed; P2P; model; framework

0 引言

随着各种定位、测绘、遥感等探测感知技术的进步推广应用,大量的时空数据不断被获取并应用于科研、生活以及军事等诸多领域中,与时空相关的应用得以不断发展,同时这也对如何有效合理地管理海量时空数据提出了挑战。

经过研究人员十几年的不断努力与创新,集中式时空索引技术已经被广泛而深入地讨论与研究,形成了若干研究思路与发展方向。然而,分布式环境下的时空索引技术目前研究得非常少,正如 Ilarri 等人[1]在《ACM 计算概览》中强调:"目前多数工作仅考虑了集中式的结构,……,并且未来的工作将会聚焦于分布式环境"。对时空数据的管理越来越向着如下趋势发展:

a)向分布式环境延伸。海量的时空数据对集中式管理提出了严峻的挑战。集中式管理对于数据量小的应用能发挥简单、高效的效果,但面对规模日益庞大的时空数据,集中式管理逐渐显示出弱势(如性能瓶颈),当数据量或访问量达到一定程度时,集中式机制将会出现性能下降,甚至瘫痪,当系统崩溃时,全部应用无法进行,单点失效。

b)强调自治。时空数据的采集单位不同,本组织应该对自己的时空数据具有维护权,而不再是臃肿的集中式控制。集中式控制带来难以准确、全面地控制的问题,庞大的时空数据的管理需要非常繁琐的控制,因此只有本组织对负责的时空数据进行管理才是最佳的选择。

c)大范围共享时空数据。时空应用呈现出全球化趋势,时空数据分布于全球,类似于采用 Internet 将全球的个人计算机互连一样,共享时空数据可以使用户获取空间上更广泛、时间跨度更大的时空信息。

在上述趋势下,时空数据的查询处理将从集中式查询处理过渡到分布式查询处理[2,3]。在这种环境下,如何从分散的 TB、PB 级海量时空数据中进行快速检索? 如何实现让用户感觉地域无差异化的透明式查询处理? 因此,研究分布式环境下的时空索引机制就具有重要作用,而这对时空索引技术提出了新的要求。

a)高效性。所建立的索引机制必须能够高效及时地完成

收稿日期: 2011-09-02; **修回日期**: 2011-10-19 **基金项目**: 国家自然科学基金资助项目(60172012);湖南省自然科学基金资助项目(03JJY3110)

作者简介:张翀(1982-),男,吉林吉林人,博士研究生,主要研究方向为分布式时空索引、对等计算等(early4932@163.com);肖卫东(1968-),男,教授,博导,博士,主要研究方向为信息资源管理、对等计算、信息组织与检索;杨晓亮(1982-),男,硕士研究生,主要研究方向为分布式时空索引.

用户的查询需求。在分布式环境下,时空数据存储于各个分散的服务器中,对某一全局资源的查询势必会遍历若干台服务器,所建立的索引必须要能够合理地组织各个服务器之间的通信策略,寻找一种最佳的遍历路径完成资源的查找。

b)协作性。时空数据分散于不同部门,索引能够支持跨部门建立,各个服务器并不是独立完成作业,而是联合起来协作式地处理。所建立的索引必须能够支持服务器之间跨组织、跨区域、跨系统、松散、可变地完成查询。

c)灵活性。所建立的索引必须具有灵活性。(a)用户查询时无须连接指定的服务器,即服务器之间的组织模式对用户来说是透明的;(b)能够支持服务器组织上的动态变化,即新服务器加入、已有服务器退出对索引的整体结构没有影响。

d)容错性。广域分布式环境中服务器与网络通信的保障性都存在不确定因素。因此,在服务器失效、通信不畅的情况下,所建立的索引应该保持容错性,维护查询处理的正确性,这点在战场环境下显得尤为重要。

因此,理想的时空数据管理模式应当以广域分布、动态自治为特点,所建立的索引应该具有良好的可伸缩性。而传统的C/S模式或层次分布式的模式不能满足以下要求:a)从系统性能的角度来说,传统的C/S模式、层次分布式模式等存在性能关键节点、伸缩性能差的问题,无法适应海量时空数据的大规模应用;b)从系统互操作性的角度来说,传统模式不适合大规模的协作,无法满足跨组织、跨职能需求,不能很好地支持多部门协同应用;c)从系统灵活性的角度需求,在分布式环境下,已经部署的服务器可能会发生迁移、下线等动态变化,也可能加入新的服务器,传统模式不能也不可能适应动态环境,特别是复杂多变的战场环境。

对等计算(P2P计算)技术作为一种新的网络计算技术,受到工业界和学术界的普遍关注,成为计算机领域研究和应用的重要热点。P2P计算具有节点间对等、节点自治、网络动态、无集中控制和大规模分布以及系统自组织等特点。本文提出采用对等计算技术构建分布式环境下的时空索引,充分满足了分布式环境下对时空索引的要求。但目前研究分布式环境下的时空索引较少,基于对等计算进行研究的甚少,因此,本文从基础理论出发研究该技术的基本模型和整体框架,为的是更好地从根本上建立一套分布式时空索引的机制,形成设计分布式时空索引的基本规范,包括索引结构、查询算法、建立算法和维护算法等。

1　相关工作

Yeh和Cambray是最早采用时间的函数将时间和空间集成到一个数据模型中的,随后又采用行为时间序列[4]的方法对其进行扩展。在该方法中,采用行为函数来描述移动对象的位置变化,尽管他们致力于将时间和空间集成在一起的模型研究,但是没有将该模型集成到数据库中。

MOST[5]模型较好地解决了集中式环境下的移动对象位置表示及将来位置查询问题。在分布式环境中,由多个服务器对移动对象的位置信息进行管理存储。一般在某一服务器的服务区域,指定的移动对象的运动方式与集中式环境下的运动方式可以看做是一样的。因此,分布式环境下的移动对象在某一服务器节点服务区内的运动状态信息也可以由MOST模型来表示。MOST引入了动态属性的概念,它将移动对象的属性分为静态属性和动态属性两类。静态属性是指不随时间变化而发生改变的属性,如汽车的牌照、厂家等。这类属性没有显式更新则被认为是不变,即传统意义上的属性,只有对数据库的相关字段值进行显式的更新时才会改变。与静态属性相对,动态属性是随时间的变化而变化,是不需要显式更新也改变的属性。动态属性是与时间紧密相关的,如行驶中的汽车位置信息等,这类属性即使没有对数据库相关字段进行显式的更新,也会随时间的改变而改变。

Moreira等人[6]在该方法的基础上继续深入研究,他们提出将移动对象的轨迹进行分段表示,在最小偏差的基础上,每段运动轨迹利用线性可变函数来表示。该模型能够描述移动对象完整的历史轨迹信息。可是该模型也存在一个问题,位置属性的精度主要依赖于所采用的数据采集设备,若该设备的精度很低,则会给实际应用的效果带来较大的影响。

2　基于空间划分的分布式时空对象模型

在分布式环境下,时空对象分散管理,所产生的时空数据也是分散存储于诸多服务器上,首先要明确的是数据是如何分片于各个服务器上的。通常有两种做法,即基于对象的划分和基于空间的划分。

2.1　两种划分模式的分析

基于对象的划分是将全局的时空对象划分为多个组,每个节点负责一个组内的时空对象,即一个组内的时空对象产生的时空数据都存储于负责该组的节点上。基于空间的划分是将整个考察空间划分为多个区域,每个节点(服务器)负责一个区域内的时空对象,即时空对象产生的时空数据存储于相应区域内负责的节点中。在不同的应用环境下,这两种划分模式有着各自的优缺点。

在基于移动对象的划分模式中,对于某一个移动对象来说,由于其在不同时刻的位置信息都由同一台服务器来管理,这种模型比较适合于基于轨迹的查询。例如,查询"移动对象 o 在未来5分钟内通过的距离",对于类似这样涉及到某个移动对象的查询,只需要在查询发起后找到管理移动对象 o 的那台服务器,然后由该服务器进行相应处理返回查询结果。而对于基于坐标的查询就要显得复杂得多。例如,查询"未来5分钟内通过区域 P 的所有移动对象",即使 P 区域非常小,在未来5分钟内通过的移动对象也可能是由许多个服务器来管理的,并且预先还不知道是由哪些服务器来管理。对于类似这样与某区域或点相关的查询,一般需要对所有服务器管理的移动对象依次进行扫描来确定有哪些移动对象与查询相关。相反地,在基于空间划分的模式中,由于每一个服务器管理当前位置处于自己管辖区域中的移动对象,使得基于空间划分的模式比较适合基于坐标的查询。例如,查询"未来5分钟内通过区域 P 的所有移动对象",一般来说,只有 P 区域附近服务器管辖区域中的移动对象才有可能在比较短的时间内进入 P 区域,只需知道 P 区域附近有哪些服务器,然后对这些服务器所管辖的移动对象作相应处理并返回查询结果。而对于基于轨迹的查寻,例如查询"移动对象 o 在未来5分钟内通过的距离",首先需要遍历网络中的服务器节点,找出由哪个服务器对 o 的当前信息进行管理,然后再进行相应处理并返回查询结果。

表面上看,这两种模式对基于坐标的查询和基于轨迹的查询是互补的。然而在现实中,考虑到移动通信的特点,当移动

对象远离其对应服务器时,要保持一定的通信质量,其通信代价必然变大。在基于移动对象划分的模式中,随着时间的推移,移动对象很可能远离其管辖服务器,而它此时很可能距离其他服务器更近;而在基于空间划分的模式中,移动对象的管辖服务器一般也是离空间距离较近的服务器。另外,当考虑系统中有移动对象退出系统空间或有新的移动对象加入时,基于移动对象划分的模式需要对移动对象集合进行重新划分,而在基于空间划分的模式中可以保持原有空间划分不变。综合考虑,本文选择基于空间划分的模式作为分布式时空对象模型的基础。

2.2　模型建立

模型采用基于空间划分的方法,即将待研究的空间进行划分,每个服务器与一个空间进行绑定,并负责管理该空间内的时空对象。该模型是一个四元组$\langle S,\mathrm{Par},\mathrm{PS},O\rangle$。其中:$S$ 为空间,至少由两个坐标属性构成,有边界,容纳时空对象的考察区域;Par 为集合,是对空间的一种分割,$\mathrm{Par}=\{C_1,C_2,\cdots,C_m\}$,每个划分 C_i 称为一个区域,并且 $i\neq j$ 时,$C_i\cap C_j=\varnothing$,$S=\bigcup_{i=1}^{m}C_i$;PS 为服务器集合,$\mathrm{PS}=\{\mathrm{peer}_1,\mathrm{peer}_2,\cdots,\mathrm{peer}_m\}$,每个节点都具有全局唯一标志;$O$ 为时空对象集合,$O=\{o_1,o_2,\cdots,o_q\}$,每个对象都有全局唯一标志 o_{id}。

为进一步将模型描述清楚,首先给出如下定义:

定义 1　标定点。标志一个时空对象具体位置,且区分时空对象在哪个区域的坐标。如点对象直接就是该点的坐标,带状对象的标定点可以选取其中心点。

定义 2　绑定。给定节点 peer_i 和区域 C_j,任意一个对象 o_k,当 o_k 的标定点(X_{o_k},Y_{o_k})属于 C_j 时,o_k 产生的时空数据都存储在 peer_i 中,则称 peer_i 与 C_j 绑定。绑定概念是逻辑上的,如节点既可以物理上设置于区域内,也可以地理上与区域无关。Peer_i 绑定的区域以 $R(\mathrm{peer}_i)$表示。

对模型的描述如下:

a)O 中既有点状对象,也有带状对象,随着时间的推移,点状时空对象仅仅位置发生变化,而带状对象除了位置发生变化外,其形状大小也可能发生变化。

b)O 的基数会随着时间发生变化,既会有新进入空间 S 的对象,也有退出空间 S 的对象。

c)O 中的对象在空间 S 中的运动状态性质有多种,有规律的运动、无规律随机运动以及受限运动等。

d)O 中时空对象向节点进行数据更新方式多样,既可以是动态实时更新,也可以是批量更新;既可以是定时更新,也可以是条件更新。条件更新是指当时空对象的某种属性发生变化时才向节点提交,如当时空对象的速度或运动方向发生变化时,提交当前时空数据。

e)Par 中的区域也会发生变化,区域可以分裂,也可以合并。

f)PS 中的节点也会发生变化。新服务器可以进行部署,通过将某一区域进行分裂产生新区域,新部署的节点负责存储新产生区域内对象的记录。已经部署好的服务器可以移去,这需要将其所对应的区域合并入邻近的区域,并将已经存在的记录交接给合并后负责的服务器进行存储。某一节点失效后,其他节点应该通过网络及时发现,将其管辖区域进行合并。

在上述模型的约束下,容易得出以下两点:

a)对于任意一个对象 o 和某一时刻 t,o 如果产生记录,则该记录一定原始存储于一个服务器中。

b)对给定任意一个对象 o 和一个时间区间$[t_s,t_e]$,o 的记录可能原始存储于多个服务器中。

时空对象在上述模型下不断产生时空数据,针对历史时空索引和将来预测时空索引,以下分别描述相应的时空数据。

2.2.1　面向历史查询的时空数据

a)数据描述。面向历史查询建立时空索引,所需要的时空数据以历史记录形式存储,该形式是一个五元组$\langle o_{\mathrm{id}},x,y,t,\mathrm{attrs}\rangle$,其中:$o_{\mathrm{id}}$代表产生此记录的时空对象的唯一标志;$(x,y)$代表时空对象的标定点坐标;$t$ 代表采集记录的时间;attrs 代表时空对象的其他属性数据,如带状对象的 attrs 可以是形状的边界数据,由一系列点坐标构成,又如运动汽车 attrs 可以代表当前剩余的汽油数量等。

b)更新策略。通过探测手段,可以实时或近实时地获得时空对象的时空数据,但何时向参与组建分布式时空索引的节点提交时空数据,这就需要讨论更新策略问题。由于面向历史的索引所需要的数据为历史数据,因此,更新策略取决于回答历史查询的精确程度。可以采用周期式更新,即每隔一段时间向负责本区域的节点提交一次时空数据。

c)元组的分布。历史记录以水平分片形式分布于各个节点,分片条件为:给定记录$\langle o_{\mathrm{id}},x,y,t,\mathrm{attrs}\rangle$,若$(x,y)$属于 $R(\mathrm{peer}_i)$,则记录存储于 peer_i。当一个对象 o 随时间的变化,其位置坐标移出某个区域进入另一区域时,则分别负责这两个区域的服务器进行交接工作,由新进入区域所在的服务器负责存储记录。

d)元组的存储。对于点状时空对象,建立索引时只需要(x,y,t)信息即可;对于带状对象,还需要提取形状信息加入索引,将 attrs 中具体的形状数据概要为最小限定矩形(minimal bounding rectangle,MBR),MBR 可在索引中代表形状对象相应时刻的大体形状。

在元组分布策略的指导下,每个节点对元组具体的存储结构如表 1 所示。

表 1　面向历史查询的元组存储结构

字段名	类型	说明
o_{id}	GUID	时空对象的全局唯一标志
标定点	point	标定点坐标
t	timestamp	产生记录的时刻
MBR	rectangle	带状对象的 MBR
其他属性	…	…
…	…	…

对于带状对象,其 MBR 可能与多个区域相交,需要进行复本存储,策略为:给定形状对象 o,若 o 在时刻 t 产生历史记录,且与 o 的 MBR_o 相交的区域集合$|\{r|r\ \mathrm{overlap}\ \mathrm{MBR}_o\}|>1$,则区域覆盖坐标$(x,y)$的区域 R 所负责的节点存储记录主本,其他区域存储记录复本。

2.2.2　面向将来预测查询的时空数据

面向将来预测查询的时空数据与面向历史查询的时空数据不同。面向历史查询的数据是以离散的历史记录进行存储,所回答的时空查询是关于过去的空间状态情况,而面向将来预测查询的机理是利用当前的时空数据预测未来的空间状态,因此所存储的是关于空间状态的函数。注意,此处所强调的与面向历史索引数据的区别在于对空间状态的查询目的上,但实际存储时,面向将来索引还是以记录形式进行存储。

a)预测模型。它在索引中用来计算将来某时刻的空间状

态。空间状态 A 由三个子属性组成,分别为 A. value、A. updatetime 和 A. function。其中 A. function 是一个关于时间 t 的函数,并且当 $t=0$ 时,该函数值为0。在 A. updatetime 时刻,A 的值为 A. value。从 A. updatetime 至下一次更新空间状态 A 之间的时间段中,当在时刻 A. updatetime + t_0 时,A 的值为 A. value + A. function(t_0)。该预测模型中的空间状态 A 可以指时空对象的位置,也可以指对象的形状、姿态等其他空间属性。为了能够方便描述,本文接下来主要以点状时空对象为例进行建立将来预测时空索引的探讨,那么空间状态 A 即指对象的位置坐标,关于时间 t 的函数为时空对象的运动速度。因此,面向将来预测查询的数据形式如下。

b)数据描述。面向将来预测索引所需的时空数据为六元组$\langle o_{id}, x, y, t, v, \text{attrs}\rangle$。其中:$o_{id}$代表产生此记录的时空对象的唯一标志;$(x,y)$代表时空对象的标定点坐标;$t$ 代表采集记录的时间;v 代表 t 时刻时空对象的速度;attrs 代表时空对象的其他属性数据。

c)更新策略。面向将来预测查询所需的时空数据是当前的空间状态,预测时利用预测模型通过索引筛选结果对象。更新策略与面向历史索引不同,由于预测模型中 A. function 为关于时间 t 的函数,如果还采用周期式更新,既不能准确预测将来的空间状态,还会产生大量无用的时空数据。因此,更新策略为:当时空对象的速度 v 发生变化时,才向负责本区域的节点提交时空数据。

d)元组的分布。元组分布策略与面向历史查询的元组分布策略相同。

e)元组的存储。它与面向历史查询的存储方式类似,但需要存储速度值,在元组分布策略的指导下,每个节点上对元组具体的存储结构如表 2 所示。

表 2　面向将来预测查询的元组存储结构

字段名	类型	说明
o_{id}	GUID	时空对象的全局唯一标志
标定点	point	标定点坐标
t	timestamp	产生记录的时刻
v	real	时空对象的速度信息,二维空间中为两个值,分别是沿 x 轴的速度和沿 y 轴的速度
其他属性	…	…
…	…	…

图 1 显示了模型的一个例子。整个空间被划分为几个不规则的区域,对象 o_1 为点状物,o_1 在七个时刻产生了更新,更新的记录分别由服务器 $peer_2$、$peer_3$、$peer_4$、$peer_5$ 和 $peer_6$ 进行存储;o_2 为带状物体,随着时间变化,o_2 不但位置发生了变化,形状也发生了改变,o_2 在四个时刻进行了更新,记录分别由 $peer_2$、$peer_6$ 和 $peer_7$ 存储。

3　双层的基于对等计算分布式时空索引整体框架

在集中式环境下,时空数据集中存储,对数据建立索引也是集中式结构。在分布式环境下,时空数据分布在各个节点上,以 P2P 技术为支撑,各个节点为对等关系,这时就需要研究在对等环境下分布式时空索引的整体概貌。

3.1　整体架构描述

基于对等计算的分布式时空索引的整体架构规范了索引具体结构设计时的框架,本文接下来对面向历史和面向将来预测索引的设计都遵循这个整体架构。整体架构包括索引的总体结构、索引查询基本协议、索引建立的基本协议和索引维护的基本协议。

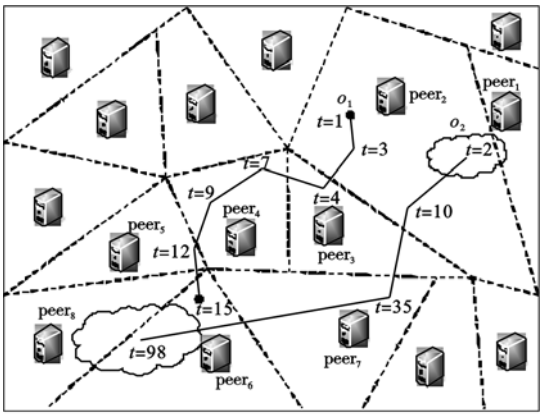

图1　基于空间划分的分布式时空对象模型举例

3.1.1　总体结构

基于对等计算的分布式时空索引整体上是一种双层结构。每个节点首先对本地存储的时空数据建立索引,然后以对等计算为指导思想,通过与其他节点建立拓扑连接完成全局索引的构建。全局时空索引负责查找符合查询条件的节点,本地时空索引负责对本地数据进行全面查找,如图 2 所示。

1)本地时空索引　本地时空索引以集中式形式组织,每个节点对本地时空数据建立相同的索引结构,针对面向历史的查询可以建立面向历史查询的索引,如 3DR-树、HR-树等,也可以建立面向将来预测查询的索引回答将来查询,如 TPR-树、B^x-树等。本地时空索引通常以树状结构进行,树的节点为一块固定大小的磁盘区域,连接节点之间的边为节点指向其他节点的指针。

2)全局时空索引　全局时空索引实质上是由各个节点建立起来的一种覆盖网(overlay)。覆盖网可看做是一个图(graph),图的顶点就是参与组建全局索引的节点,边是各个节点建立的网络连接。覆盖网中各个节点地位平等,不存在中心。建立网络连接的目的在于对其他节点所存储的时空数据进行了解,这样在实施查询时就可以将查询有选择性地投递,减少了网络开销,提高了查询速度,发挥了索引的作用。

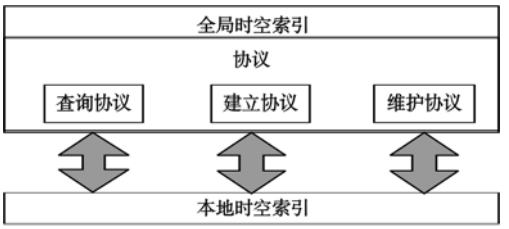

图2　双层索引结构

从总体上看,全局时空索引是一个图,这个图具体的形成过程是由各个节点通过建立连接并保存为路由表而形成的。一个路由表是通过在内存中建立链表形成的,每个链表项为一个二元组〈时空范围,节点〉。其中,时空范围表示对应节点中所存储的时空数据涵盖的范围。路由表在进行全局时空查询时具有重要作用,查询条件通过与时空范围进行比较,那些与查询条件无交集的时空范围所对应的节点将不会被投递。

全局时空索引的设计遵循 P2P 计算的规律,它的平均扇出度决定了全局时空查询的效率,也决定了全局时空索引的维护开销。平均扇出度提高,查询速度将会提高,当形成全连通的结构时,网络内任何一个节点到其他节点只需 1 跳,查询效

率达到最高。扇出度的提高同时也带来了索引维护的开销,因为 P2P 计算中每个节点都要不停地维护路由表,度数的提高意味着路由表的规模扩大,将增加维护开销。

3.1.2　索引查询基本协议

基于对等计算的分布式时空索引的查询基本协议按照先全局索引再本地索引的基本流程进行。通过将查询条件与路由表进行比较,选择可能存在查询结果的节点,再将查询投递至这些候选节点,然后利用本地时空索引在本地查找获得最终结果。协议的基本流程如下:

a)节点 $peer_p$ 发起时空查询(spatio-temporal query,STQ),$peer_p$ 将 STQ 与本身路由表中各个表项的时空范围进行比较,将与 STQ 有时空范围交集的节点选定为候选集 CP;

b)$peer_p$ 将查询投递至 CP 中的所有节点;

c)收到查询消息的节点执行与 $peer_p$ 相同的操作,并且利用本地时空索引进行时空对象查找,并将结果返回至 $peer_p$。

3.1.3　索引建立基本协议

基于对等计算的分布式时空索引的建立过程包括各个节点首先在本地建立集中式的时空索引结构,然后根据全局时空索引的结构与相应的节点建立拓扑连接,再将连接保存为路由表的形式,从而形成具有一定结构的全局索引。基本协议可以分为如下几步:

a)各个节点首先建立本地时空索引;

b)节点 $peer_p$ 启动建立全局索引的协议;

c)某一个节点 $peer_q$ 根据全局索引的结构设计与 $peer_p$ 建立连接,$peer_p$ 将自己所辖的时空范围与 $peer_q$ 交换,建立路由表;

d)其余节点从已经加入全局索引协议的节点中随机挑选一个,与之建立连接,遵照加入协议,参与组建全局索引;

e)所有节点进入全局索引后,索引建立过程完毕。

3.1.4　索引维护基本协议

P2P 计算是一个完全动态的环境,节点在本地自治,并且可以自主地加入和退出,网络环境的不稳定可以使某些节点失效。退出与失效的区别在于:节点退出是“有意识”的退出,要遵守节点退出协议,而失效是“无意识”的,其他节点需要先检测到这种情况再进行处理。时空数据同时也在不断产生,数据的更新同样需要索引的配合。因此,基于对等计算的分布式时空索引的维护包括:

a)节点加入。新节点可以自主地加入全局索引。节点加入协议规定了新节点如何加入全局索引并维护既定的结构,基本流程如下:

(a)新节点 $peer_p$ 通过某种方式与已经加入全局索引的某个节点 $peer_q$ 进行联系;

(b)$peer_q$ 根据全局索引的结构以及 $peer_p$ 上所辖时空范围或将要管辖的时空范围并依据全局索引的查询协议,将 $peer_p$ 的加入请求路由至相应的网络邻居 $peer_r$;

(c)$peer_p$ 与 $peer_r$ 建立连接,交换时空范围并保存路由表,新节点加入完毕。

b)节点退出。节点的退出需要发出退出请求,具体过程如下:

(a)节点 $peer_p$ 将要退出全局索引,$peer_p$ 向自己的某个或某几个邻居发出退出消息;

(b)$peer_p$ 的邻居收到退出请求后,将 $peer_p$ 的时空数据进行交接;

(c)交接完毕后,$peer_p$ 的邻居更新路由表,仍然是维护原来既定的全局索引结构,$peer_p$ 退出完毕。

c)节点失效。节点失效后,其他节点通过心跳协议不断检测,发现节点不在网络中后,启动类似退出协议的程序,最终仍然是维护原来既定的全局索引结构。

d)数据更新。数据更新的目的是使全局索引的查询保持一致性。时空对象不断向节点提交数据,节点除了在本地更新索引后,还要通知其邻居节点时空范围的变化,具体过程如下:

(a)节点 $peer_p$ 更新了时空数据,$peer_p$ 根据本地索引的更新程序进行本地时空索引的更新;

(b)若更新的为带状数据,且与其他节点有交叠,$peer_p$ 还要更新其他节点上该数据的复本;

(c)若时空数据的更新引起节点 $peer_p$ 的时空范围发生变化,$peer_p$ 还要通知其他节点更新路由表。

3.2　整体架构的分析

整体架构的提出从总体上规范了基于对等计算分布式时空索引的设计原则。整体架构是一套设计框架,接下来的索引的具体设计都从该框架派生。总的来看,整体架构的优点体现在以下几个方面:

a)合理性。海量时空信息的分布式存储意味着需要高度的自治性,而基于对等计算的分布式时空索引的双层结构满足了这种自治性的需求。随着时空数据不断更新,本地时空索引保证了数据在本地可以自主维护。本地时空索引的建立为快速查找本地时空信息提供了工具。

全局时空索引保证了从全局上查找时空对象的完整性。时空数据分散于各个节点中,各节点将自身所辖时空范围作为路由提示信息进行消息转发,这就将原来无序地分散在各处的时空信息有序地组织起来,从而能够提高查询效率。

b)完整性。基于对等计算的分布式时空索引整体架构规定了总体框架、索引的查找、建立和维护基本协议,成体系地建立起一套基于对等计算的总体布局。

索引作为一种数据结构,在定义时就需要结构上的描述,同时,利用这种结构进行查询需要给出查询算法,在数据动态更新时还需要给出索引的动态维护算法。整体框架从这几个方面都对本文所提的基于对等计算的分布式时空索引进行了定义和总体描述,说明了整体架构的完整性。

c)可扩展性。研究表明,资源信息服务必须具备很强的可扩展性。多个资源能够按照一定的拓扑结构和组织形式进行聚合,从而消除系统单点失效的可能性和性能上的瓶颈。在整体架构中,其可扩展性很强。首先,索引查找协议采用了节点转发的方式,通过一定的转发策略可以避免大量扩散搜索信息的产生,从而使该协议具有很强的可扩展性;其次,从拓扑结构上看,必要时可以通过增加节点的个数来提高网络的容量,本地索引可令节点互不干扰地在本地完成数据更新和查询任务,系统的吞吐量较高。

4　模型用例研究

基于空间划分的分布式时空对象模型可以指导如何管理分布式时空对象,接下来以三个用例进行应用研究。

4.1　信息化联合作战战场监控

未来的数字化战场是一种分布式环境,信息化条件下,网络信息系统将陆、海、空、天、电构成多维一体的战场,夺取战场

信息优势可以借助卫星、无人机、地面传感器等设备采集战场的实时信息,快速有效地实现信息的共享,获取战场信息资源作为作战决策中的重要支撑。单兵、车辆、飞机等作战单位都可以理解为时空对象,对战场的实时监控离不开管理这些时空对象的时空信息。如图 3 所示,战场中各参战军种、各级指挥机构及单兵之间都能实现信息的高度共享。

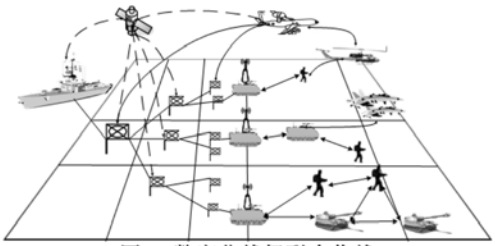

图3　数字化战场联合作战

以本文所提的模型为指导,以每支参战部队所辖的地域为依据进行空间划分。划分后的区域内的作战单元即为该划分内的时空对象,时空对象实时向管辖自身的指挥中心报告位置及状态(时空数据),进入友邻部队辖区内的时空对象向该辖区的中心进行报告。通过这样对战场时空数据的管理,就形成了联合作战和联合指挥。

假设在某次联合作战中有三支参战力量,分别为陆军 A、海军 B、空军 C。A 主要侦查地面战场的情报,监控地面战场态势的变化,其侦查手段和方式主要有侦查分队、无人机和地面传感器。B 主要侦查海面战场的情报,监控海面战场态势的变化,其侦查手段和方式主要有卫星、地面雷达。C 主要侦查空中战场的情报,监控空中战场态势的变化,其侦查手段和方式主要有预警机、地面雷达。三支参战力量所侦查的情报通过分布式时空查询处理系统进行管理。

由上述描述可将模型进一步具体化为:

a) S 为整个战场;

b) Par 中的区域即为 A、B 和 C 初始既定负责的区域;

c) 在 A、B、C 的指挥中心分别建立服务器节点;

d) 时空对象为参战的装甲车、舰船、飞机。

例如,在 15:00 时刻 A 可以通过分布式时空查询处理系统查询此时距海岸线 40 海里范围内有多少敌方舰艇,并可以预测在 15:30 时刻有多少敌方舰艇;同时 A 也可以查询此时有多少敌方飞机向自己所在阵地飞来,多长时间之后可能对自己构成威胁。B 可以查询在敌方某区域内有多少阵地防空力量和机动防空力量,并可以预测敌方机动防空力量的动向。

4.2　传感器网中分布式时空查询

随着通信和传感器技术的迅速发展,世界范围的传感器网(world-wide sensor Web,WSW)的设想即将成为现实。通过建立统一的时空信息基础设施,分散的传感器网络(sensor network)将组成一个整体的时空传感器网(spatio-temporal sensor Web,SSW)。整个时空传感器网的基础架构如图 4 所示。

系统结构主要由以下四部分组成:

a) 时空传感器网层(spatio-temporal sensor Web layer,SSL)。理论上讲,系统是由多个 SSW 层构成,每个 SSW 层针对不同的需求提供不同的服务,如 SSW 的气温层可以提供气温数据。每个层都是由相同类型的 SSW 节点组成。

b) 时空传感器网层节点(spatio-temporal sensor Web layer node,SLN)。系统由多个 SLN 组成,每个 SLN 以 P2P 的方式进行网络连接,并提供时空信息查询。每个 SLN 存储着不同位置的时空信息,通过分布式索引结构对信息进行索引。当用户进行查询时,通过建立的索引结构查询信息,以提高查询效率。

c) 时空传感器节点(spatio-temporal sensor node,SSN)。SSN 是传感器节点和 SLN 之间的网关,在 SSW 中扮演数据提供者的角色。每个 SSN 从传感器中收集原始数据,并发送给一个或多个 SLN。

d) 系统客户端。客户端可以给用户提供二维或者三维的地图,用户通过客户端可以查询自己所需的相关信息。

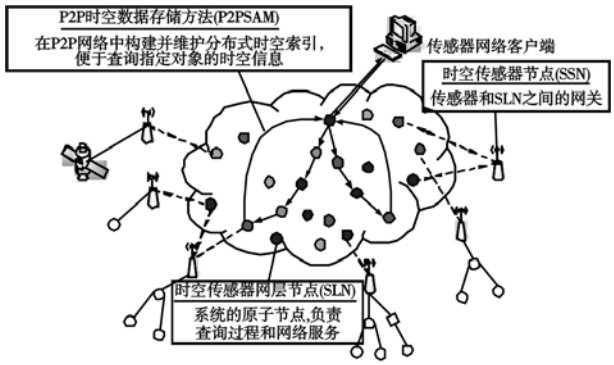

图4　时空传感网示例

在时空传感器网中,传感器将采集到的时空信息存储于基站,不同的基站经底层网络连通相互通信,通过引入基于 P2P 的时空查询处理机制,可以极大地方便用户进行全局时空传感器网的时空查询。例如,用户可以通过系统的客户端查询某块区域在某时刻的温度、湿度、光照情况等。

模型在此处应用的具体化为:S 为整个感知的地域;Par 中的区域为各个单独的传感器网络,每个传感器网络负责一个区域;各个传感器网络的基站建立时空服务节点;各个传感器网络采集的数据随时间而变化,不断存储至服务器中。

4.3　物联网

物联网[7](Internet of things)是通过射频识别(RFID)、红外感应器、全球定位系统、激光扫描器等信息传感设备,按约定的协议把任何物品与互联网连接起来进行信息交换和通信,以实现智能化识别、定位、跟踪、监控和管理的一种网络,其结构如图 5 所示。

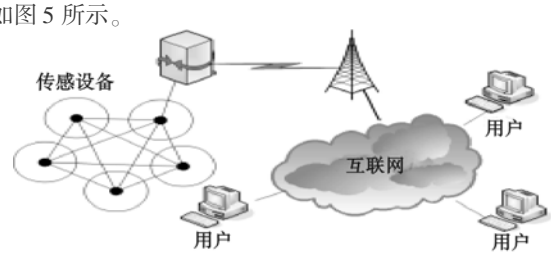

图5　物联网示例

在物联网中,各传感器节点分布式地部署在各地,采集的信息也分布式地存储在相应的存储节点上。由于信息的分布式存储,当用户进行信息查询时,查询效率很低,因此需要构建分布式时空索引,以提高查询效率。当一个传感器采集到信息之后,采用分布式时空查询处理系统将信息索引,从而提高查询效率。

模型在此处应用的具体化为:S 为整个要感知的空间;Par 中的区域为各个服务提供商所管辖的空间;每个服务提供商建立自己的服务器节点;物品位置和属性信息随时间一起构成时空数据,经由传感器感知存储至服务器。用户可以查询在某次物流运输过程中,某货物在指定的时间范围内经过哪些中转站。

5　整体框架验证

本文在笔者开发的系统——基于对等计算的分布式时空查询处理系统[8]上对总体框架进行了验证。以总体框架为依据,分别实现了面向历史查询的索引和面向将来预测查询的索引。

数据采用 GSTD[9]时空数据生成方法,该方法在时空索引的实验中被广泛采用。基于 GSTD 方法生成了实验数据集。基态数据为中心点呈 $\mu=0.5$、$\sigma=0.1$ 的高斯分布,边长呈(0,0.1)均匀分布的 N_d 个矩形数据;时间采样点数量为 100,且均匀分布;中心点的变化量(Δ_{center})呈(-0.2,0.2)均匀分布,边长的变化量(Δ_{extent})呈(-0.05,0.05)均匀分布。随着时间的推进,数据的整体变化为从待考察空间的中心向四周扩散。图 6 展示了 $N_d=1\ 000$ 时,时刻 t 分别为 0、0.25、0.5、0.75 和 1 时的所有数据。

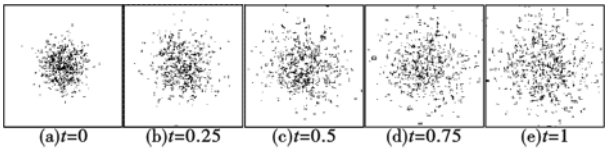

图6　数据集

实验如下:

a)响应时间。将本文的系统与集中式的 HR 树[10]和 TPR* 树[11]进行查询响应时间的对比。设定查询空间窗口的选择率为 0.05,节点数量为 1 000,不断增加数据规模 N_d 从 1 000 至 100 000,测试响应时间。实验结果如图 7 所示。

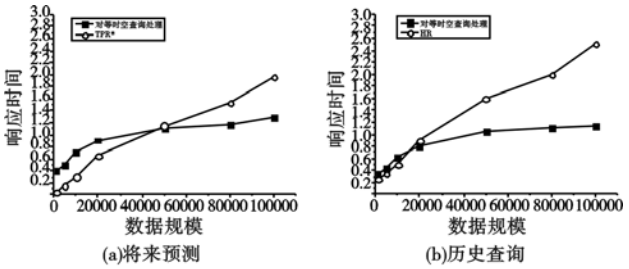

图7　响应时间对比

b)查询代价。设定查询空间窗口的选择率为 0.05,数据规模 N_d 为 5 000,增加节点数量从 250 至 10 000,测试查询引起的消息数量。实验结果如图 8 所示。

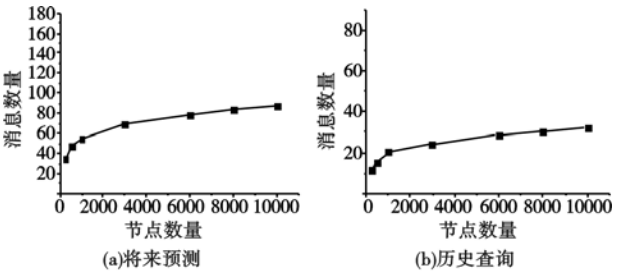

图8　查询代价对比

c)窗口查询。设定数据规模 N_d 为 5 000,节点数量为 1 000,查询空间窗口的选择率从 0.001 变化至 0.07,测试查询引起的消息数量。实验结果如图 9 所示。

由图 7 可知,随着数据规模的不断增大,无论是将来预测查询还是历史查询,基于对等计算的时空查询处理技术的响应时间要低于对应的 TPR* 树和 HR 树,并且响应时间与数据规模的变化成对数关系,说明基于对等计算的时空查询处理技术具有较好的可扩展性。

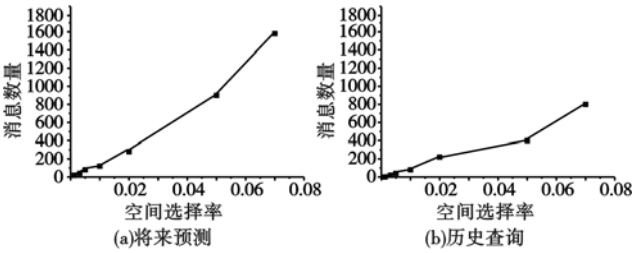

图9　响应时间对比

由图 8 可知,从网络规模的角度度量,基于对等计算的时空查询处理技术具有良好的可扩展性,随着网络规模不断扩大,查询所引起的消息数量成对数增长。

由图 9 可知,随着空间选择率的不断增大,将来预测查询所引起的消息数量增长得更快,这是因为将来预测查询中需要扩大查询窗口,这会引起更多的消息查询。

6　结束语

本文提出了采用 P2P 技术组织服务器建立索引,并从基础出发研究相应的分布式时空数据模型和索引建立的整体架构。以三个用例在自行开发的系统上验证所提的架构,结果表明整体架构合理,效果明显。下一步工作将会以此为基础,设计并实现性能优良的索引结构和算法。

参考文献:

[1] ILARRI S, MENA E, ILLARRABEBDI A. Location-dependent query processing: where we are and where we are heading [J]. ACM Computing Surveys, 2010, 42(3): 1-73.

[2] HERNANDEZ C, RODRIGUEZ M A, MARIN M. A P2P meta-index for spatio-temporal moving object databases[C]//Proc of the 13th International Conference on Database Systems for Advanced Applications. Berlin: Springer-Verlag, 2008: 653-660.

[3] LANGE R, DÜRR F, ROTHERMEL K. Scalable processing of trajectory-based queries in space-partitioned moving objects databases [C]//Proc of ACM Workshop on Advances in GIS. New York: ACM Press, 2008: 31.

[4] YEH T S, De CAMBRAY B. Time as a geometric dimension for modeling the evolution of entities: a 3D approach[C]//Proc of International Conference on Integrating GIS and Environmental Modeling. 1993.

[5] SISTLA A P, WOLFSON O, CHAMBERLAIN S. Modeling and querying moving objects[C]//Proc of the 13th Intenratinal Conference on Data Engineering. Piscataway, NJ: IEEE Press, 1997: 422-432.

[6] MOREIRA J, RIBEIRO C, SAGLIO J M. Representation and manipulation of moving points: an extended data model for location estimation [J]. Cartography and Geographic Information Systems, 1999, 26(2): 109-123.

[7] International Telecommunication Union (ITU). ITU Internet Reports: the Internet of things[R]. Geneva, Switzerland: ITU, 2005.

[8] 杨晓亮,张翀,汤大权,等. 基于对等计算的分布式时空查询处理系统的设计及应用研究[J]. 计算机应用研究, 2011, 28(4): 1342-1347.

[9] THEODORIDIS Y, SILVA J R O, NASCIMENTO M A. On the generation of spatiotemporal datasets[C]//Proc of the 6th International Symposium on Spatial Databases. Berlin: Springer-Verlag, 1999: 147-164.

[10] NASCIMENTO M, SILVA J R O. Towards historical R-trees[C]//Proc of ACM Symposium on Applied Computing. 1998: 235-240.

[11] TAO Yu-fei, PAPADIAS D J S. The TPR*-tree: an optimized spatio-temporal access method for predictive queries[C]//Proc of the 29th International Conference on Very Large Data Bases. San Fransisco, CA: Morgan Kaufmann, 2003: 790-801.