



(12)发明专利

(10)授权公告号 CN 104521200 B

(45)授权公告日 2017.09.15

(21)申请号 201380036596.4

(22)申请日 2013.08.29

(65)同一申请的已公布的文献号
申请公布号 CN 104521200 A

(43)申请公布日 2015.04.15

(30)优先权数据
61/694,658 2012.08.29 US
13/839,823 2013.03.15 US

(85)PCT国际申请进入国家阶段日
2015.01.09

(86)PCT国际申请的申请数据
PCT/US2013/057370 2013.08.29

(87)PCT国际申请的公布数据
W02014/036310 EN 2014.03.06

(73)专利权人 甲骨文国际公司
地址 美国加利福尼亚

(72)发明人 B·博格丹斯基 B·D·约翰森

(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038

代理人 袁玥

(51)Int.Cl.
H04L 12/933(2006.01)

(56)对比文件
US 2005/0071709 A1,2005.03.31,
CN 102164081 A,2011.08.24,
Wei Lin Guay et al.“vFtree - A Fat-tree Routing Algorithm using Virtual Lanes to Alleviate Congestion”.《2011 IEEE International Parallel & Distributed Processing Symposium》.2011,第197-208页.
Nishanth Dandapanthula.“InfiniBand Network Analysis and Monitoring using OpenSM”.《The Ohio State University》.2011,正文第1.2、2.2-2.3、3.4.2节.

审查员 马旗超

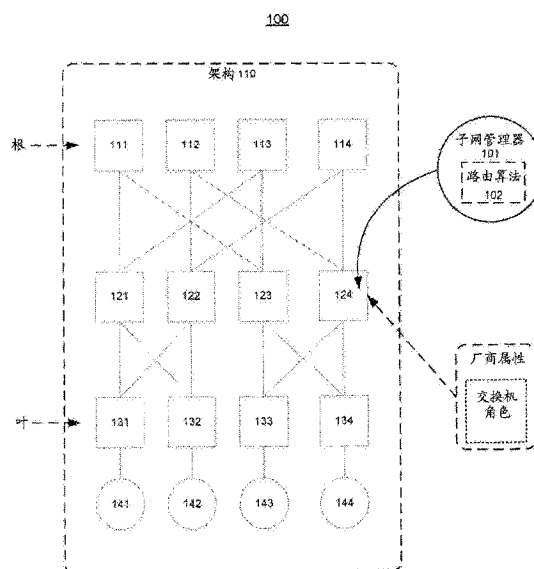
权利要求书3页 说明书6页 附图5页

(54)发明名称

用于在中间件机器环境中支持发现和路由退化胖树的系统和方法

(57)摘要

一种能够支持具有多个交换机的架构中的发现和路由的系统和方法。所述系统允许使用交换机角色标记架构中的一或多个交换机。随后,架构中的子网管理器检测与所述一或多个交换机相关联的交换机角色。此外,根据所检测的与所述一或多个交换机相关联的交换机角色,可在架构上应用路由算法。



1. 一种用于支持架构中的发现和路由的方法,所述架构具有多个交换机、多个节点和子网管理器,所述方法包括:

用从叶交换机和根交换机中选择的交换机角色标记所述架构中的所述多个交换机中的每个交换机,其中,所述标记包括在每个交换机的能够被子网管理器检测的属性中明确地定义交换机角色;

经由子网管理器检测与所述多个交换机中的每个交换机相关联的交换机角色;以及在所述标记和所述检测之后,基于所检测的与所述多个交换机相关联的交换机角色,在所述架构上应用胖树路由算法,

其中,所述架构被配置为具有多个层的胖树拓扑。

2. 根据权利要求1所述的方法,其中

所述属性是能够被子网管理器用子网管理分组查询检测的子网管理分组属性。

3. 根据权利要求1或2所述的方法,进一步包括:

向路由算法提供一或多个根全局唯一标识符GUID。

4. 根据权利要求1或2所述的方法,进一步包括:

放宽针对所述架构的胖树拓扑验证和链路计数不一致性检查的限制,使得所述多个交换机中的在胖树拓扑的所述多个层中的同一层上的任何两个交换机不具有相同数量的上行链路和下行链路的情况下,胖树路由算法不失败。

5. 根据权利要求1或2所述的方法,进一步包括:

作为组件替换之后的正常交换机配置维护的一部分,在所述多个交换机中的每个交换机的属性中保存和恢复每个交换机的交换机角色。

6. 根据权利要求1或2所述的方法,进一步包括:

将所述架构中的一或多个交换机从一个交换机角色翻转为另一交换机角色。

7. 根据权利要求1或2所述的方法,进一步包括:

根据交换机角色检测架构中的连接错误。

8. 根据权利要求1或2所述的方法,进一步包括:

检测所述多个交换机中的特定交换机,其中所述特定交换机用叶交换机的交换机角色标记,并且其中在胖树拓扑中所述特定交换机没有连接的节点。

9. 根据权利要求8所述的方法,进一步包括:

将所述特定交换机的交换机角色从叶交换机翻转为根交换机。

10. 根据权利要求8所述的方法,进一步包括:

确保所述特定交换机被置于所述胖树拓扑中正确的层。

11. 一种用于支持架构中的发现和路由的系统,所述架构具有多个交换机、多个节点和子网管理器,所述系统包括:

一或多个微处理器,

在所述一或多个微处理器上运行的子网管理器;

其中,所述多个交换机中的每个交换机与从叶交换机和根交换机中选择的交换机角色相关联,所述交换机角色由此在每个交换机的能够被子网管理器检测的属性中明确地定义交换机角色;

其中,子网管理器被配置为检测与所述多个交换机中的每个交换机相关联的交换机角

色,基于所检测的与所述多个交换机相关联的交换机角色,在所述架构上应用胖树路由算法,

其中,所述架构被配置为具有多个层的胖树拓扑。

12. 根据权利要求11所述的系统,其中:

所述属性是能够被子网管理器用子网管理分组查询检测的子网管理分组属性。

13. 根据权利要求11或12所述的系统,其中:

一或多个根全局唯一标识符GUID被提供给路由算法。

14. 根据权利要求11或12所述的系统,其中:

子网管理器被配置为放宽针对所述架构的胖树拓扑验证和链路计数不一致性检查的限制,使得所述多个交换机中的在胖树拓扑的所述多个层中的同一层上的任何两个交换机不具有相同数量的上行链路和下行链路的情况下,胖树路由算法不失败。

15. 根据权利要求11或12所述的系统,其中:

作为组件替换之后的正常交换机配置维护的一部分,在每个交换机的属性中保存和恢复交换机角色。

16. 根据权利要求11或12所述的系统,其中:

所述多个交换机中的一或多个交换机被从一个交换机角色翻转为另一交换机角色。

17. 根据权利要求11或12所述的系统,其中:

子网管理器操作以根据与所述多个交换机中的一个或多个交换机相关联的交换机角色检测所述架构中的连接错误。

18. 根据权利要求11或12所述的系统,其中

子网管理器被配置为检测所述多个交换机中的特定交换机,其中所述特定交换机用叶交换机的交换机角色标记,并且其中在胖树拓扑中所述特定交换机没有连接的节点。

19. 根据权利要求18所述的系统,其中:

子网管理器被配置为将所述特定交换机的交换机角色从叶交换机翻转为根交换机。

20. 一种其上保存有指令的非临时性机器可读存储介质,所述指令用于支持架构中的发现和路由,所述架构具有多个交换机、多个节点和子网管理器,所述指令在被执行时使得所述架构执行以下步骤:

用从叶交换机和根交换机中选择的交换机角色标记所述架构中的所述多个交换机中的每个交换机,其中,所述标记包括在每个交换机的能够被子网管理器检测的属性中明确地定义交换机角色;

经由子网管理器,检测与所述多个交换机中的每个交换机相关联的交换机角色;以及

在所述标记和所述检测之后,基于所检测的与所述多个交换机相关联的交换机角色,在所述架构上应用胖树路由算法,

其中,所述架构被配置为具有多个层的胖树拓扑。

21. 一种用于支持架构中的发现和路由的系统(500),所述架构具有多个交换机(520)、多个节点和子网管理器,所述系统包括:

标记模块(510),被配置为用从叶交换机和根交换机中选择的交换机角色标记所述架构中的所述多个交换机中的每个交换机,其中,所述标记包括在每个交换机的能够被子网管理器检测的属性中明确地定义交换机角色;

检测模块(530),被配置为检测与所述多个交换机(520)中的每个交换机相关联的交换机角色;以及

算法模块(540),被配置为在所述标记和所述检测之后,基于所检测的与所述多个交换机(520)相关联的交换机角色,在所述架构上应用胖树路由算法,

其中,所述架构被配置为具有多个层的胖树拓扑。

22.一种用于支持架构中的发现和路由的系统,所述架构具有多个交换机、多个节点和子网管理器,所述系统包括:

用于用从叶交换机和根交换机中选择的交换机角色标记所述架构中的所述多个交换机的装置中的每个交换机,其中,所述标记包括在每个交换机的能够被子网管理器检测的属性中明确地定义交换机角色;

用于经由子网管理器检测与所述多个交换机中的每个交换机相关联的交换机角色的装置;以及

用于在所述标记和所述检测之后,基于所检测的与所述多个交换机相关联的交换机角色在所述架构上应用胖树路由算法的装置,

其中,所述架构被配置为具有多个层的胖树拓扑。

23.根据权利要求22所述的系统,其中

所述属性是能够被子网管理器用子网管理分组查询检测的子网管理分组属性。

24.根据权利要求22或23所述的系统,进一步包括:用于向所述路由算法提供一或多个根全局唯一标识符的装置。

25.根据权利要求22或23所述的系统,进一步包括:用于放宽针对所述架构的胖树拓扑验证和链路计数不一致性检查的限制的装置,使得所述多个交换机中的在胖树拓扑的所述多个层中的同一层上的任何两个交换机不具有相同数量的上行链路和下行链路的情况下,胖树路由算法不失败。

26.根据权利要求22或23所述的系统,进一步包括:用于作为组件替换之后的正常交换机配置维护的一部分,在所述多个交换机中的每个交换机的属性中保存和恢复每个交换机的交换机角色的装置。

27.根据权利要求22或23所述的系统,进一步包括:用于将所述架构中的一或多个交换机从一个交换机角色翻转为另一交换机角色的装置。

28.根据权利要求22或23所述的系统,进一步包括:用于根据交换机角色检测架构中的连接错误的装置。

29.根据权利要求22或23所述的系统,进一步包括:用于检测所述多个交换机中的特定交换机的装置,其中所述特定交换机用叶交换机的交换机角色标记,并且其中在胖树拓扑中所述特定交换机没有连接的节点。

30.根据权利要求29所述的系统,进一步包括:用于将所述特定交换机的交换机角色从叶交换机翻转为根交换机的装置。

31.根据权利要求30所述的系统,进一步包括:

用于确保所述特定交换机被置于所述胖树拓扑中正确的层的装置。

用于在中间件机器环境中支持发现和路由退化胖树的系统和 方法

[0001] 版权声明

[0002] 本专利文献中的一部分公开内容包含受版权保护的素材。当专利文献或专利公开出现在可公开获得的美国专利商标局文件或档案中时,版权所有人不反对任何人影印再现专利文献或专利公开,不过保留其它方面的所有版权。

技术领域

[0003] 本发明一般地涉及计算机系统,具体涉及中间件机器环境。

背景技术

[0004] 胖树拓扑被用于高性能计算(HPC)集群,并且被用于基于无限带宽(IB)技术的集群。和大部分其它拓扑一样,对胖树来说,路由算法有益于网络资源的高效利用。然而,当涉及交换机到交换机通信时,现有的路由算法存在局限性。现有的路由算法都不支持有益于高效系统管理的无死锁和全连接的交换机到交换机通信。这些是本发明的实施例意图针对的一般领域。

发明内容

[0005] 这里描述可以支持具有多个交换机的架构中的发现和路由的系统和方法。所述系统允许架构中的一或多个交换机被标记以交换机角色。随后,架构中的子网管理器可以检测与所述一或多个交换机相关联的交换机角色。此外,可基于所检测的与所述一或多个交换机相关联的交换机角色在架构上应用路由算法。

附图说明

[0006] 图1表示根据本发明的实施例的在网络环境中支持交换机角色机制的示例。

[0007] 图2图解根据本发明的实施例的用于在网络环境中支持交换机角色机制的示例流程图。

[0008] 图3表示根据本发明的实施例的网络环境中的退化胖树的例示。

[0009] 图4表示根据本发明的实施例的修复网络环境中的退化胖树的例示。

[0010] 图5表示根据本发明的实施例的系统的示意框图。

具体实施方式

[0011] 附图中举例而非限制地图解了本发明,附图中相同的附图标记指示相似的元件。应注意本公开中对“一个”或“一些”实施例的引用不一定指的是相同实施例,并且这样的引用意味至少一个。

[0012] 本发明的如下说明利用无限带宽(IB)网络作为高性能网络的例子。对本领域的技术人员来说,显然可以无限制地使用其它种类的高性能网络。另外,如下的本发明的说明利

用胖树拓扑作为架构拓扑的例子。对本领域的技术人员来说,显然可以不受限制地使用其它种类的架构拓扑。

[0013] 这里描述的是可以支持具有多个交换机的架构中的发现和路由的系统和方法。

[0014] 无限带宽 (IB) 体系结构

[0015] 根据本发明的实施例,IB网络可被称为子网,其中子网由利用交换机和点对点链路互连的一组主机组成。IB架构可构成一或多个子网,每个子网可利用路由器互连。子网内的主机和交换机是利用本地标识符 (LID) 寻址的,一个子网局限于49151个LID。

[0016] IB子网可具有至少一个子网管理器 (SM),子网管理器 (SM) 负责初始化和培养 (bring up) 网络,包括存在于子网中的交换机、路由器和主机通道适配器 (HCA) 上的所有IB端口的配置。在初始化时,SM始于发现状态,在发现状态下,SM进行网络的扫描,以便发现所有交换机和主机。在发现状态期间,SM可以发现其它SM,并且可以协商哪个SM应是主SM。当发现状态完成时,该SM进入主状态。在主状态中,SM继续进行LID分配、交换机配置、路由表计算和部署和端口配置。一旦主状态结束,该子网就正常运行并随时可用,在子网被配置之后,SM负责监视网络的变化。

[0017] 另外,SM可负责计算路由表,所述路由表维持全连接、无死锁,和所有源和目的地对之间的适当负载均衡。可在网络初始化时计算路由表,并且每当拓扑变化时可以重复该处理,以便更新路由表并且确保最佳性能。

[0018] 在正常操作期间,SM执行网络的定期光扫描 (light sweep),以检查拓扑变化事件,比如当链路故障时,当增加设备时,或者当链路被移除时。如果在光扫描期间发现变化事件,或者如果SM收到以信令通知网络变化的消息 (陷阱) 时,SM可按照发现的变化重新配置网络。这种重新配置还可包括在初始化期间使用的步骤。此外,存在于各个设备上的子网管理代理 (SMA) 可生成响应以控制分组 (子网管理分组 (SMP)),并且可以配置用于子网管理的本地组件。

[0019] IB是无损连网技术,其中可以每个虚拟通道 (VL) 地进行流量控制。VL是相同物理链路上的具有独立的缓存、流量控制和拥塞管理资源的逻辑信道。VL的概念使得可以在物理拓扑之上建立虚拟网络。这些虚拟网络或层可用于诸如高效路由、死锁避免、容错和服务差异化之类的各种用途。

[0020] 胖树路由

[0021] 根据本发明的实施例,胖树拓扑是分层的网络拓扑,例如,平衡的胖树可在每一层具有相同的链路容量。此外,通过建立具有多个根的树,例如,可利用XGFT符号描述的m端口n树定义或k叉n树定义,可以实现胖树拓扑。

[0022] 为了构成更大的拓扑,可基于将几个胖树连接在一起,而不是建立单一的大胖树来建立系统。从几个单一胖树建成的这种胖树可被称为多核胖树。使用水平链路通过叶交换机可以互联多核胖树,或者通过在胖树底部使用附加层交换机可以互连多核胖树,在胖树底部,每个这种附加层的交换机被连接到构成多核胖树的所有胖树。

[0023] 胖树路由算法可采用可用的网络资源。胖树路由算法可包括两个阶段:其中从源转发分组的上行阶段,和当分组被转发给目的地时的下行阶段。在最低的共同先辈处,发生这两个阶段之间的转变,所述共同先辈是通过其向下端口可到达源和目的地的交换机。这种路由实现确保无死锁,并且所述实现还确保朝向相同目的地的每条路径会聚于相同的根

(顶端)节点,以致朝向所述目的地的所有分组沿向下方向的单一专用路径而行。通过针对每个目的地都具有专用的下行路径,下行阶段中的竞争被有效消除(转移到上行阶段),以致去往不同目的地的分组只争用其路径上的一半交换机中的输出端口。另外,过度定制的胖树中的下行路径不是专用的,可被几个目的地共享。

[0024] 此外,优化的胖树路由算法的架构发现复杂度可被表示成 $O(m+n)$,其中 m 是边(链路)的数目, n 是顶点(节点)的数目。路由复杂度为 $O(k \cdot n)$,其中 k 是端节点的数目, n 是交换机的数目。

[0025] 根据本发明的实施例,不同的路由算法,比如分层最短路径(LASH)、无死锁单源最短路径(DFSSSP)和最小跳数路由可用于支持胖树拓扑中的路由。

[0026] LASH路由是一种用于不规则网络的确定性最短路径路由算法。可以利用最小路径路由所有分组,通过找出并断开经由虚拟通道(VL)的循环,所述算法实现无死锁。此外,LASH不可以按任何方式均衡通信量,这在胖树架构中是显然的。该算法目的在于利用最小数目的VL,于是在同一层上,即,利用相同的链路,路由所有可能的无死锁对。LASH的计算复杂度为 $O(n^3)$,其中 n 是节点的数目。

[0027] DFSSSP路由是一种用于任意拓扑的高效健忘路由。它利用虚拟通道来保证无死锁,并且与LASH相比,其目的在于在路由处理期间不限制可能路径的数目。与LASH相比,它还利用改进的启发法来减少使用的虚拟通道的数目。DFSSSP的问题在于针对交换机到交换机流量,它假设无死锁,并且不会中断针对交换机到节点对和交换机到交换机对来说可能发生的任何循环。离线DFSSSP的计算复杂度为 $O(n^2 \cdot \log(n))$,其中 n 为节点的数目。

[0028] 最小跳数路由是用于OpenSM的默认回退路由算法。它找出所有端点之间的最小路径,并试图平衡本地交换机处的每条链路的路由数目。然而,利用最小跳数路由通常导致信用回路,这会使架构死锁。最小跳数的复杂度为 $O(n^2)$,其中 n 是节点的数目。

[0029] 交换机角色机制

[0030] 根据本发明的实施例,交换机角色机制可用于使架构发现的复杂问题和路由问题分开。

[0031] 图1表示根据本发明的实施例,在网络环境中支持交换机角色机制的例子。如图1中所示,网络环境100可包括子网管理器(SM)101,SM 101管理具有多个交换机(例如,交换机111-114、121-124和131-134)和多个主机节点(例如,节点141-144)的架构110,例如,IB架构。

[0032] 系统可实现用于明确地定义架构110中的稍后可由SM 101检测的交换机角色的交换机角色机制。例如,厂商SMP属性103可用于定义架构110中的交换机角色104。经由厂商特有的SMP,可以查询厂商SMP属性103。

[0033] 此外,架构110中的每个交换机可被分配主机名、IP地址和节点描述。通过利用厂商属性,系统可以使SM 101获得交换机角色信息,而不依赖于SM 101配置输入或者用于动态地提供配置信息的任何其它带外接口。另外,交换机角色可被保存和恢复,作为组件替换之后的正常交换机配置维护的一部分,因为它不像硬件全局唯一标识符(GUID)似的被绑定到实际硬件实例。

[0034] 作为替代地,系统可把诸如RootGUID的地址信息提供给路由算法,以便产生类似的效果。这种情况下,系统可能需要在(多个)组件替换操作之后维护正确的列表。

[0035] 根据本发明的实施例,交换机角色机制可向架构110中的每个交换机提供它应遵循的简单角色。例如,系统可用架构中的每个交换机的相应角色物理标记每个交换机。如图1中所示,根交换机111-114(所述根交换机被置于架构110的没有任何上行链路的顶部)可具有角色“根”,叶交换机131-134(所述叶交换机连接到架构110底部的主机节点)可具有角色“叶”。

[0036] 利用交换机角色机制,系统可以缩短架构发现时间(例如,可能不需要一致性检查)。此外,系统可以减少或者甚至消除从路由算法102发现架构110的需要,这意味在路由表生成期间犯错的可能性将低得多。

[0037] 图2图解根据本发明的实施例,用于在网络环境中支持交换机角色机制的示例流程图。如图2所示,在步骤201处,系统允许架构中的一或多个交换机被标记以交换机角色。随后,在步骤202处,子网管理器可检测与所述一或多个交换机相关联的交换机角色。此外,在步骤203处,子网管理器可根据检测的与所述一或多个交换机相关联的交换机角色,在架构上应用路由算法。

[0038] 退化的胖树

[0039] 根据本发明的实施例,可在用于无限带宽(IB)网络的现有胖树路由算法中识别缺陷,并且系统可利用减轻当发现和路由退化架构时遇到的问题的不同扩展。首先,可以放宽对拓扑验证的限制以使胖树路由更通用。其次,通过可由厂商特有SMP查询的厂商子网管理分组(SMP)属性,可以实现交换机标记,并且该交换机标记可用于使用特定的架构角色配置交换机,这使拓扑发现和实际路由分离。另外,通过利用SMP属性可以解决翻转(flipping)交换机问题。

[0040] 在IB网络中,可存在其中胖树路由会失败的不同情形。例如,如果在相同层上的任何两个交换机上的上行链路和下行链路的数目不相等,那么拓扑验证会失败。另外,甚至当整个架构中的一个链路出故障时,胖树路由会回退到最小跳数路由。

[0041] 通过放宽对拓扑验证的限制并禁止链路计数不一致性检查,在任何不完整的胖树上,胖树路由默认不会失败。此外,系统可利用交换机角色机制使胖树路由(例如,OpenSM中的当前实现)能够路由不能通过严格的拓扑验证的任何非纯正胖树架构。

[0042] 图3表示根据本发明的实施例的网络环境中的退化胖树的例示。如图3中所示,IB架构300可包括多个交换机(例如,交换机311-314、321-324和331-334),和多个主机节点(例如,节点341-343)。

[0043] 当存在例如交换机334的未连接节点的叶交换机时,IB架构300中可能出现问题。从而,IB架构300变成退化的胖树。

[0044] 图4表示根据本发明的实施例的修复网络环境中的退化胖树的例示。如图4中所示,IB架构400可包括多个交换机(例如,交换机311-314、321-324和331-333),和多个主机节点(例如,节点341-343)。

[0045] 另外,最初被分类成叶交换机(如图3中示为交换机334)的交换机434可被重新分类成不是叶交换机。如图4中所示,为了修复拓扑问题,交换机434可被重新定位在具有两个子交换机323-324的新层次,leaf level+2。

[0046] 如图4中所示,涉及翻转交换机的胖树路由可以是反直觉的。此外,由于需要向交换机分派不正确的排行(rank),因此翻转交换机使胖树路由中的故障解决变得困难。另外,

提供修复可能是有问题的,因为只有当架构400中出现排行冲突时,SM才会反应性地起作用,即,SM必须首先检测冲突,随后重新排行架构。这种要求可能麻烦,并且由于架构的高复杂度,冲突可能不可检测。

[0047] 根据本发明的实施例,系统可利用交换机角色机制修复翻转的交换机。例如,可以利用下述算法1来确保架构中的交换机总是被置于正确的排行。

算法 1 *osm_ftree_rank_fabric(p_tree)function*

需要: 固件厂商特有的交换机角色

[0048] **1: if switch has no CNs then**
 2: if smpquery(switch, role) == leaf then
 3: switch.rank = tree_rank
 4: end if

[0049] **5: end if**

[0050] 随后,通过利用在前述各节中描述的不同的胖树路由算法,系统可以在架构400上运行胖树路由。

[0051] 图5表示根据本发明的实施例的系统500的示意框图。系统500包括标记模块510、交换机模块520、检测模块530和算法模块540。标记模块510用交换机角色标记具有多个交换机520的架构中的一或多个交换机。检测模块530检测与所述一或多个交换机520相关联的交换机角色。算法模块540根据检测的与所述一或多个交换机520相关联的交换机角色,在架构上应用路由算法。

[0052] 一种支持具有多个交换机的架构中的发现和路由的系统包括:用于使用交换机角色标记架构中的一或多个交换机的装置;用于经由子网管理器检测与所述一或多个交换机相关联的交换机角色的装置;和用于根据所检测的与所述一或多个交换机相关联的交换机角色,在架构上应用路由算法的装置。

[0053] 系统还包括用于使用厂商属性明确地定义稍后可被子网管理器检测的交换机角色的装置。

[0054] 系统还包括用于向路由算法提供一或多个根全局唯一标识符 (GUID) 的装置。

[0055] 系统还包括用于放宽对关于架构的拓扑验证和一致性检查的限制的装置。

[0056] 系统还包括用于作为在组件替换之后的正常交换机配置维护的一部分而保存和恢复交换机角色的装置。

[0057] 系统还包括用于使用另一个交换机角色标记架构中的另外的一或多个交换机的装置。

[0058] 系统还包括用于根据交换机角色检测架构中的连接错误的装置。

[0059] 系统还包括用于将架构中的多个交换机配置成胖树的装置。

[0060] 系统还包括用于检测胖树中未连接节点的叶交换机的装置。

[0061] 系统还包括用于确保叶交换机被置于正确排行的装置;和用于允许路由算法在胖树上运行的装置。

[0062] 通过使用包括一或多个处理器、存储器和/或根据本公开的教导编程的计算机可读存储介质的一或多个常规的通用或专用数字计算机、计算装置、机器或微处理器,可以便利地实现本发明。对软件领域的技术人员来说显然根据本公开的教导有经验的程序员能够容易地准备适当的软件编码。

[0063] 在一些实施例中,本发明包括计算机程序产品,所述计算机程序产品是其上/其中保存有可用于对计算机编程以执行本发明的任意处理的指令的存储介质或者计算机可读介质。所述存储介质可包括但不限于任意种类的盘(包括软盘、光盘、DVD、CD-ROM、微驱动和磁光盘)、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、闪存装置、磁卡或光卡、纳米系统(包括分子存储器IC)、或者适于保存指令和/或数据的任何种类的介质或装置。

[0064] 出于图解和说明的目的提供了本发明的上述说明。上述说明并不是详尽的,也不意图把本发明局限于公开的具体形式。对本领域技术人员来说,许多修改和变化是明显的。所述修改和变化包括公开的特征的有关组合。选择和说明实施例是为了最好地说明本发明的原理及其实际应用,从而使本领域其他技术人员能够理解本发明的各个实施例和适于预期的特定使用的各种修改。本发明的范围旨在由以下权利要求及其等同物限定。

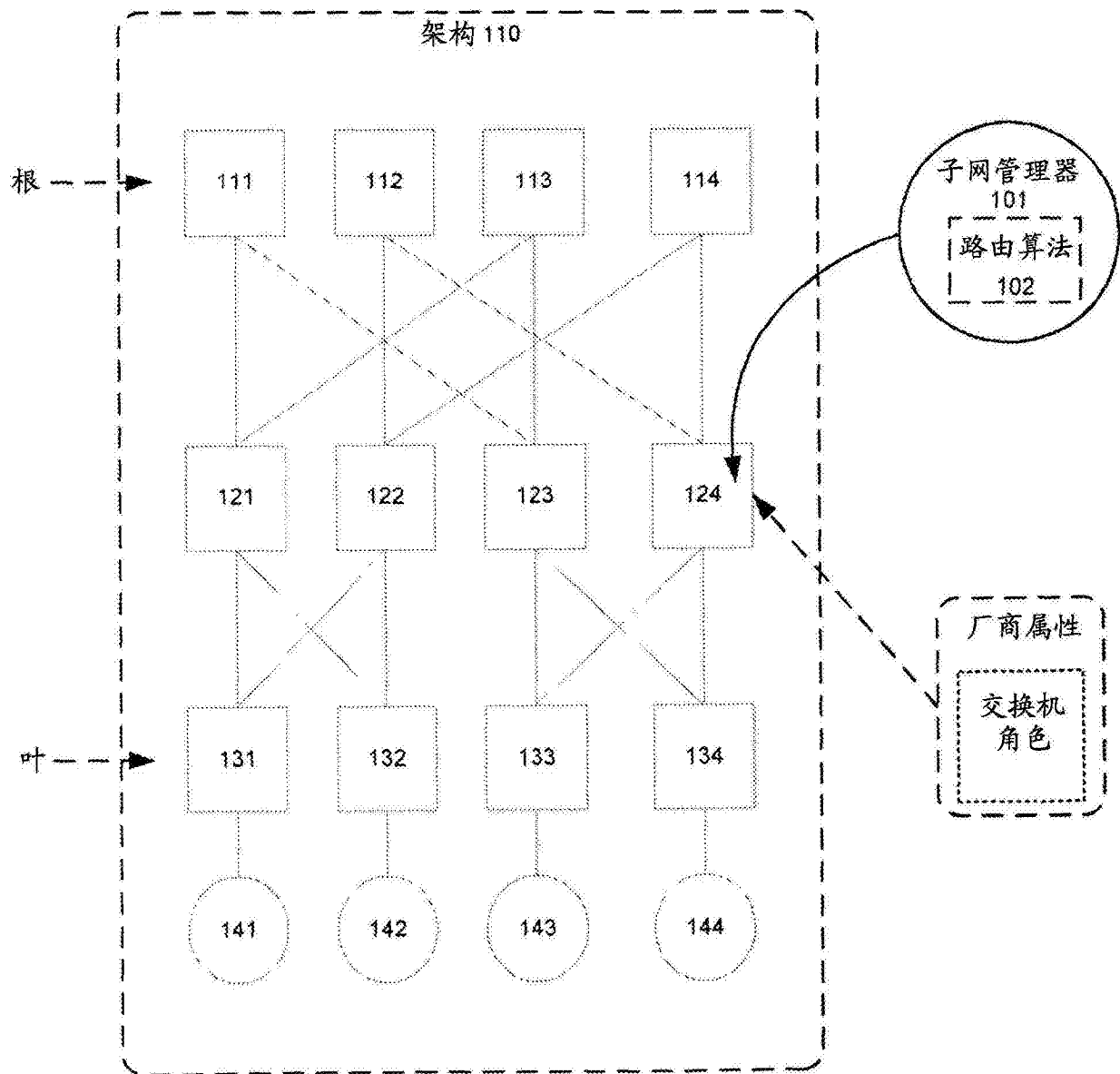
100

图1

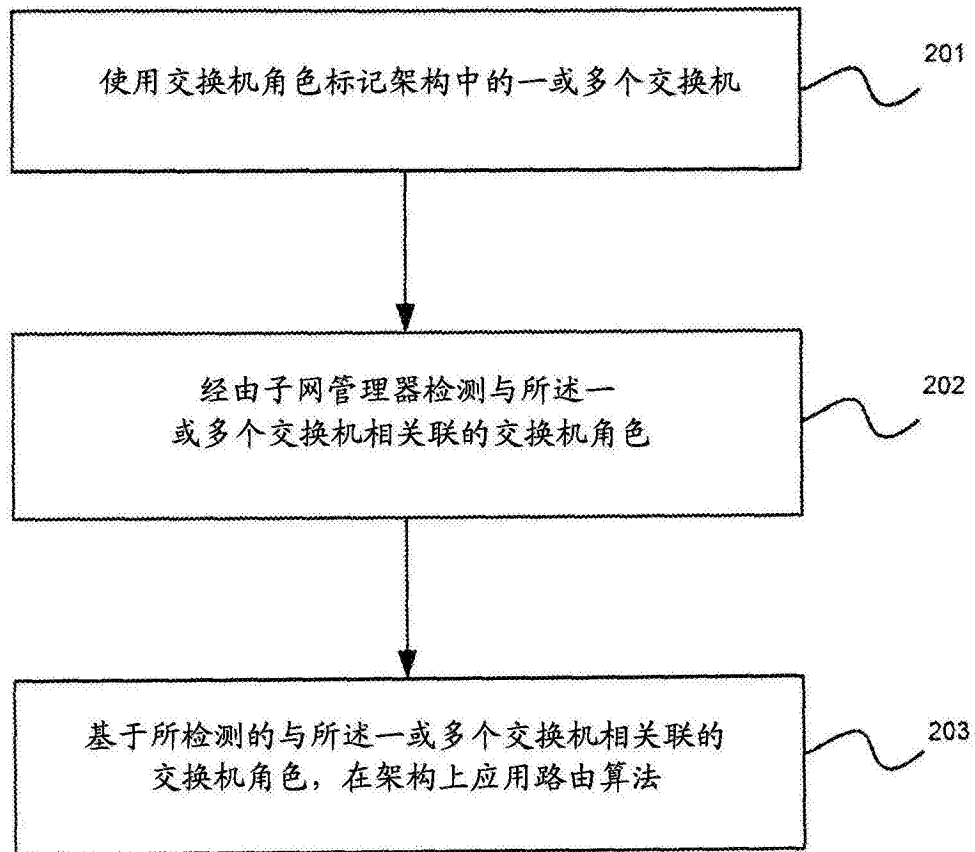


图2

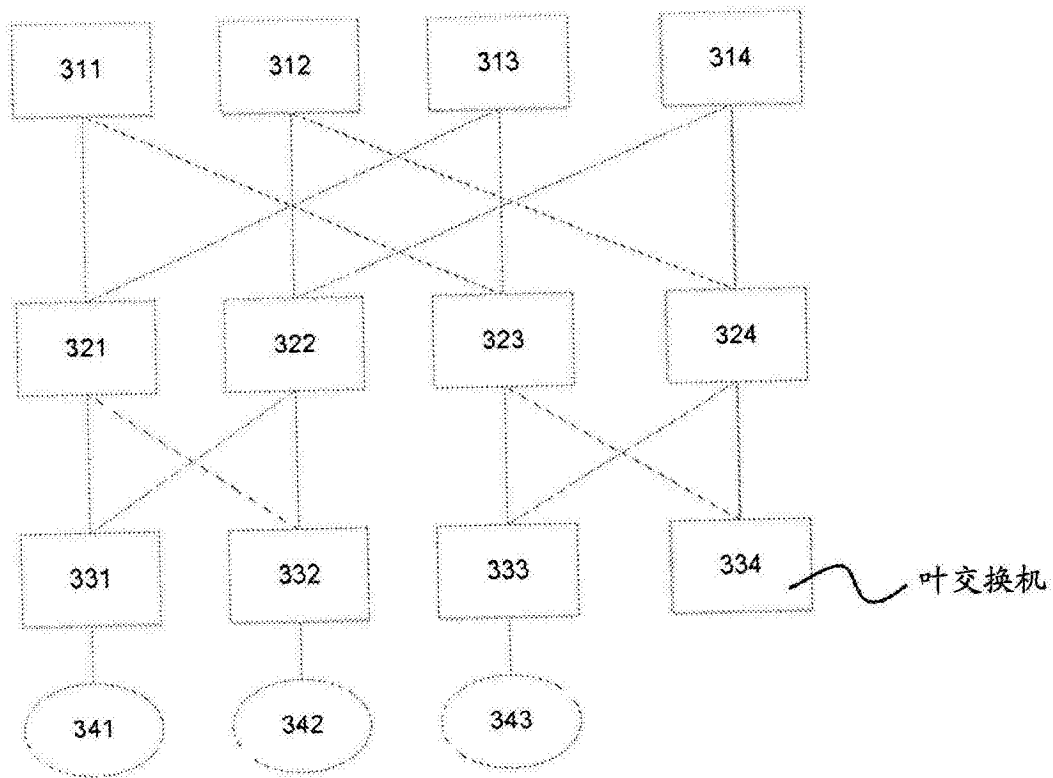
300

图3

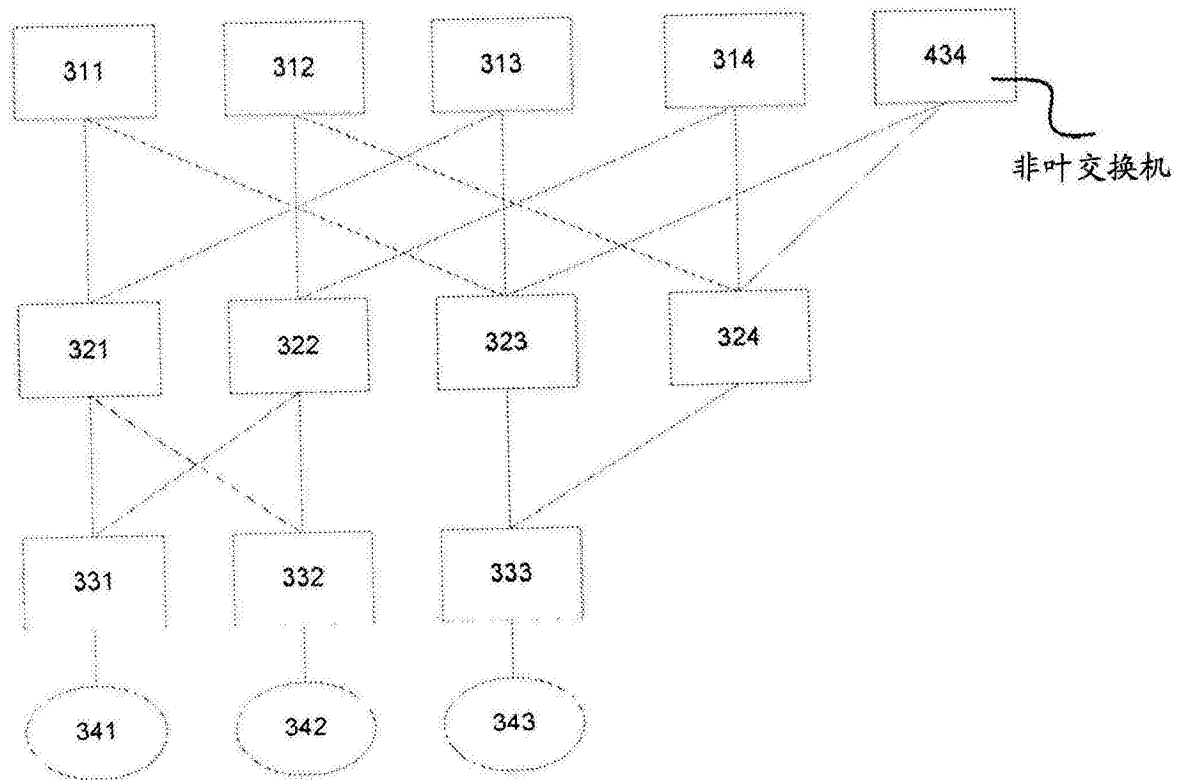
400

图4

500

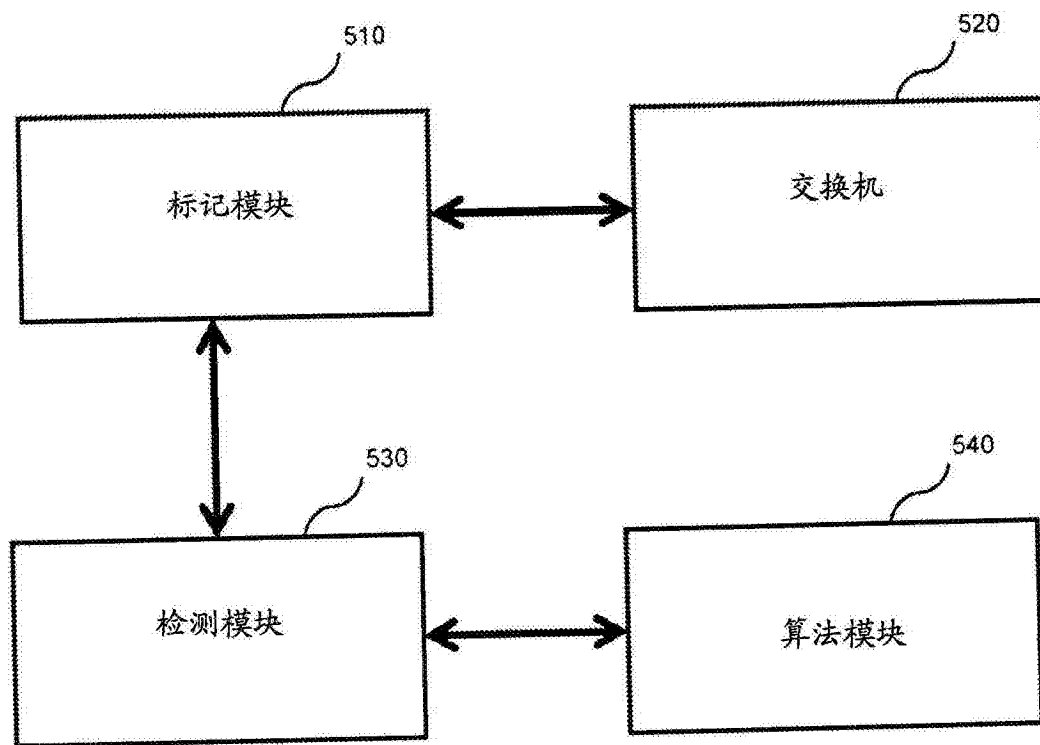


图5