



[12] 发 明 专 利 说 明 书

[21] ZL 专利号 00810124.8

[45] 授权公告日 2004 年 6 月 9 日

[11] 授权公告号 CN 1153162C

[22] 申请日 2000.5.10 [21] 申请号 00810124.8

[30] 优先权

[32] 1999.5.10 [33] NO [31] 19992269

[86] 国际申请 PCT/NO2000/000155 2000.5.10

[87] 国际公布 WO2000/068834 英 2000.11.16

[85] 进入国家阶段日期 2002.1.9

[71] 专利权人 快速检索及传递公司

地址 挪威奥斯陆

[72] 发明人 B·斯文根 K·M·里斯维克

A·哈拉尔斯 T·埃格

审查员 蔡 萍

[74] 专利代理机构 中国专利代理(香港)有限公司

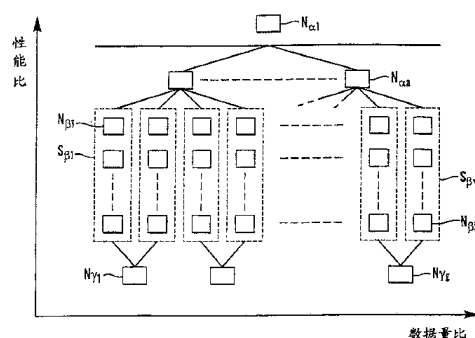
代理人 吴立明 梁 永

权利要求书 3 页 说明书 9 页 附图 6 页

[54] 发明名称 具有二维线性可缩放并行结构的搜索引擎

[57] 摘要

用具有二维线性可缩放并行结构的搜索引擎搜索文档集(D), 搜索引擎包含用于形成连接在网络中的节点(N)组的数据处理单元, 第一组节点包含发送节点(N_a), 第二组节点包含搜索节点(N_b)以及第三组节点包含索引节点(N_y)。搜索节点(N_b)成组为列, 这些列处于发送节点(N_a)和索引节点(N_y)之间, 并行连接在网络中。发送节点(N_a)用来处理搜索请求和搜索结果, 搜索节点(N_b)包含搜索软件, 至少部分搜索节点还包含至少一个搜索处理模块(M), 索引节点(N_y)通常用来给搜索软件生成索引I。还可已有第四组节点即采集节点(N_s), 采集节点用来处理搜索结果, 这就可以分离搜索任务中的发送节点。通过采用各自的结构匹配来实现数据容量缩放和搜索引擎性能缩放, 从而实现了二维缩放。



1. 一种具有二维线性可缩放并行结构的搜索引擎, 用于搜索
文本文档集 D , 其中, 文档可以分成若干文档分区 $d_1, d_2, \dots, d_n$,
5 其中, 文档集 D 在文本过滤系统中经预处理后可以获得预处理文档集
 D_p 以及相应的预处理文档分区 $d_{p1}, d_{p2}, \dots, d_{pn}$, 其中从文档集 D 中
可以生成索引 I , 这就为每个预处理文档分区 $d_{p1}, d_{p2}, \dots, d_{pn}$ 生成对
应的索引 $i_1, i_2, \dots, i_n$, 其中用文档分区相关的数据组 $d_{p,k}$ 对文档集 D
的文档分区 d 进行搜索, 其中 $1 \leq k \leq n$, 且其中该搜索引擎包含形成
10 联接在网络中的节点组(N)的数据处理单元, 其特征在于:

第一组节点, 包含 a 个发送节点($N_{a1} \dots N_{aa}$), 第二组节点,
包含 b 个搜索节点($N_{\beta 1} \dots N_{\beta b}$), 第三组节点, 包含 g 个索引节点
($N_{\gamma 1} \dots N_{\gamma g}$), 以及可选的第四组节点, 包含 e 个采集节点
($N_{\delta 1} \dots N_{\delta e}$),

15 发送节点(N_a), 联接在网络的多级结构中,

搜索节点(N_β), 分组成 v 列(S), 这些列处于发送节点(N_a)
和索引节点(N_γ)之间, 并行连接在网络中, 该发送节点(N_a)用
来处理搜索查询和搜索结果, 搜索查询再发送给所有的搜索节点
(N_β), 如果没有采集节点(N_δ), 搜索结果就返回给发送节点
20 (N_a)并在此综合成最终的搜索结果,

每个搜索节点(N_β)用来存放搜索软件,

索引节点(N_γ)通常用来为搜索软件生成索引 i 以及用来对各
包含一个搜索处理器模块的搜索节点(N_β)可选地生成文档分区相
关的数据组 $d_{p,k}$,

25 如果有采集节点(N_δ), 这些节点与发送节点(N_a)类似,
联接在网络的多级结构中, 用来汇集搜索查询的答案并输出其最终
结果, 这样发送节点不再执行该项任务,

用文档分区 d 的数目 n 对搜索节点列(S)的数目 v 进行比例缩
放, 就可以实现对数据容量的缩放; 用估计或预期的业务量负荷对

每个搜索节点列 (S_β) 中的搜索节点 (N_β) 的数目 b/v 进行缩放, 可以实现缩放性能, 这样, 在任何情况下, 通过让每个搜索节点列 (S) 包含一个文档分区相关的数据组 $d_{p,k}$ 并且让搜索节点列 (S_β) 中的所有搜索节点 (N_β) 每一个含有相同的文档分区相关数据组 $d_{p,k}$, 进行对文档集 D 的搜索。。

5 2. 按照权利要求 1 的搜索引擎, 其特征不在于网络中发送节点 (N_a) 和可选采集节点 (N_s) 的多级结构由分层树结构提供。

 3. 按照权利要求 2 的搜索引擎, 其特征不在于可选采集节点 (N_s) 的多级结构是发送节点 (N_a) 的多级结构的一个镜像。

10 4. 按照权利要求 2 的搜索引擎, 其特征不在于分层树结构是二进制树结构。

 5. 按照权利要求 1 的搜索引擎, 其特征不在于每个搜索节点 (N_β) 包含搜索软件模块 (SW)。

 6. 按照权利要求 5 的搜索引擎, 其特征不在于至少部分搜索节点
15 (N_β) 包含至少一个专用搜索处理器模块 (M), 每个专用搜索处理器模块 (M) 由一个或多个专用搜索处理器芯片 (PMC) 实现, 每个芯片用于并行处理多个 q 搜索查询。

 7. 按照权利要求 6 的搜索引擎, 其特征不在于专用搜索处理器芯片 (PMC) 由具有 y 组处理器组 (G) 的处理器模块 (M) 提供, 每组处理器含有 z 个搜索处理器芯片 (PMC), 每组处理器与分配给处理器
20 组 (G) 的存储器 (RAM) 相连, 并从中接收数据。

 8. 按照权利要求 1 的搜索引擎, 其特征不在于数据容量缩放中的文档分区 d 的数目的增加伴随相应的发送节点 (N_a) 数目的增加, 根据情况还伴随采集节点 (N_s) 数目的增加, 以及可选地还
25 伴随索引节点 (N_y) 数目的增加。

 9. 按照权利要求 5 的搜索引擎, 其特征不在于: 每个搜索节点 (N_β) 仅包括搜索软件模块 (SW), 该文档分区相关数据组仅包括索引 i_k 。

30 10. 按照权利要求 1 的搜索引擎, 其特征不在于一个或多个搜索节点包括搜索软件模块 (SW) 和一个或多个专用搜索处理器模

块 (M)，其特征在于该文档分区相关数据组 $d_{p,k}$ 包括预处理文档分区 d_p 和相关索引 i_k 。

11. 按照权利要求 1 的搜索引擎，其特征在于每个独立的节点组 $(N_\alpha, N_\beta, N_\gamma, N_\delta)$ 由一个或多个联接在数据通信网络中的工作站来实现。

5

具有二维线性可缩放并行结构的搜索引擎

5 技术领域

本发明涉及具有二维线性可缩放并行结构的搜索引擎。

背景技术

10 现有的绝大多数搜索引擎技术需要处理大量的数据，它借助具有强大计算能力的计算机来实现搜索。而搜索问题是数据可分割处理问题，可利用这个特点把搜索问题分解成大量的细节查询，并把每个不同的查询交给并行联接在网络中的同量级的处理器同时处理。特定的搜索问题可以看作二进制数据分割处理问题。因此可用二进制树形网络建立一个多处理器结构，如已公开的美国专利 No. 4860201 (Stolfo 等人) 和国际专利申请，PCT/N099/00308 专利属于本申请人，在此一并引用作为参考。该专利申请人开发了用于在普通索引文本中搜索的专利技术。这些技术尤其建立在上述国际专利申请 PCT/N099/00308 中论述的用于搜索的搜索系统和搜索方法上的，这些技术属于本申请人，在此一并引用。按照本发明，搜索系统基于有效的核心搜索算法，该算法可用在搜索引擎中。

20 然而，搜索不断增大的文档、处理不断增加的业务量负荷即由搜索系统处理的每秒钟的查询数，变得越来越重要了。这就要求除了在处理器层面上能够同时处理大量的查询之外，还应该使搜索引擎优选地具有二维可线性缩放的结构。即，既考虑数据容量和性能，又考虑每秒钟需处理大量查询的能力。纵观万维网 (WWW) 的发展，由于目前的因特网中文献资料 and 用户的数量以巨大的增长率增加，搜索引擎结构中的可缩放性变得尤为重要。

30 现有的解决因特网搜索问题的搜索引擎技术能够缩放到一定的水平，但是几乎所有的这种缩放总是需要付出高昂的成本来扩容搜索引擎系统，以增加数据容量和数据通信。通常，系统的成本与数据容量或流量的平方成比例，因而数据容量每提高一倍将导致系统成本提高四倍。而且，目前所有主要的因特网搜索引擎都基于非常昂贵的服务器技术，并常常伴有粗暴的计算机强行访问 (brute

computing force-approaches)和其他的如服务器缓慢处理、需要附加特定的具有容错能力的硬件等缺点。比如,系统成本能够用建立一个搜索引擎所需的硬件成本来计算或者以实际的系统总成本来计算。

5 发明内容

因此本发明的主要目标是提供一个具有多级数据和并行处理功能的搜索引擎。它允许大量用户同时、快速搜索大量数据。

特别地,本发明的另一目标是提供用于建立一个具有多级数据和并行处理功能的搜索引擎的并行结构。

10 而本发明还有一个目标是提供一个并行结构,它是二维线性可缩放的,即数据容量和性能,即查询速度。

本发明涉及具有二维线性可缩放并行结构的搜索引擎,该搜索引擎用来搜索文本文档集 D ,其中,文档可以分成多个文档分区 $d_1, d_2, \dots, d_n$,其中,文档集 D 在文本过滤系统中经预处理后可以获得
15 预处理文档集 D_p 以及相应的预处理文档分区 $d_{p1}, d_{p2}, \dots, d_{pn}$,其中可以从文档集 D 生成索引 I ,也就为每个以前的预处理文档分区 $d_{p1}, d_{p2}, \dots, d_{pn}$ 生成相应的索引 $i_1, i_2, \dots, i_n$,其中利用文档分区相关的数据组 $d_{p,k}$ 进行对文档集 D 的文档分区 d 的搜索,其中 $1 \leq k \leq n$,且其中搜索引擎包含数据处理单元,该数据处理单元形成联接在网络中的
20 节点组。

本发明的搜索引擎能够满足上述目标并具有其他特点和优点,其特点在于第一组节点包括 a 个发送节点,第二组节点包括 b 个搜索节点,第三组节点包括 g 个索引节点,以及可选的第四组节点包括了
25 e 个采集节点,发送节点联接在网络中的多级结构上。搜索节点按列分成组,这些列处于发送节点和索引节点之间,并行连接在网络中;发送节点用来处理搜索请求和搜索结果;搜索请求还要分配给所有搜索节点,如果没有采集节点,则搜索结果返回到发送节点并在此综合为最终的搜索结果,每个搜索节点都包含搜索软件,索引节点用于生成搜索软件的索引 i ,并针对包括搜索处理器模块的搜索节点
30 可选地生成文档分区相关的数据组 $d_{p,k}$,如果有采集节点,则这些节点类似于发送节点,连接在多级网络结构上,用于汇集搜索请求

的结果并输出最终的查询结果，这样发送节点就不再执行该项任务，用文档分区 d 的数目 n 对搜索节点列的数目 v 进行比例缩放，就可以实现对数据容量的缩放；用估计或预期的通信量对每个搜索节点列中的搜索节点进行缩放，可以实现缩放性能，这样，在任何情况下，对文档集 D 的搜索发生在包含一个文档分区相关的数据组 $d_{p,k}$ 的单个搜索节点以及含有相同的文档分区相关数据组 $d_{p,k}$ 的搜索节点的列中的所有搜索节点。

按照本发明，分层树结构用来实现发送节点的多级结构和可选采集节点网络。因而，可选采集节点的多级结构就可以作为发送节点的多级结构的镜像，分层树结构用来实现二进制树结构。

按照本发明，其优点是每个搜索节点包含一个搜索软件模块。

还有，按照本发明，至少部分搜索节点包含至少一个专用的搜索处理器模块，每个专用搜索处理器模块由一个或多个专用处理器芯片实现，这些芯片用来并行处理大量的搜索请求。在这方面，专用的搜索处理器芯片位于搜索处理器模块的 y 个处理器组中。每个处理器组有 z 个搜索处理器芯片，每个芯片与处理器组中的存储器相连并从存储器中接收数据。

按照本发明，可以通过增加搜索节点组或节点列来较好地增加数据容量缩放中的文档分区数目。在这方面，文档分区数目的增加相应伴随发送节点数的增加，以及如果有采集节点则伴随采集节点数的增加和伴随索引节点数的增加。

按照本发明，文档分区数的增加能够通过相应增加发送节点和，若有采集节点则增加采集节点，以及通过可选地增加索引节点的数目来很好的实现。

最后，按照本发明的优点，每个独立节点组由一个或多个连接在数据通讯网络中的工作站来实现。

附图说明

根据本发明，搜索引擎可以由非限制性典型实施方案和参考附图来加以说明，其中

图 1 示出了按照本发明的搜索引擎的第 1 个实施方案中的结构的概要，

图 2 示出了按照本发明的搜索引擎的第 2 个实施方案中的结构

的概要,

图 3 示出了一个包含搜索软件和专用搜索芯片硬件的搜索节点的示意图,

图 4 示出了搜索芯片的单模块设计方案,

5 图 5 示出了在专用搜索芯片中处理查询信息的示意图,

图 6 示出了性能缩放原理的示意图,

图 7 示出了基于二进制树结构的多级发送节点结构的示意图

图 8 示出了基于本发明的搜索引擎的第 1 个实施方案中的结构的概要及其二维缩放原理, 以及

10 图 9 示出了按照本发明的搜索引擎的第 2 个实施方案中的结构的概要及其二维缩放原理。

具体实施方式

搜索大型的独立文档集是一项高度并行的任务。按照本发明, 搜索引擎采取不同等级的并行处理, 在下文中展开详细论述。

15 按照本发明的搜索引擎搜索文档 D 的文档集。文档可以分解成 n 个文档分区 $d_1, d_2, \dots, d_n$, 每个文档集 D 或文档分区集的文档分区 d 能用硬件文本过滤系统进行预处理, 比如, 可由专用的硬件如所谓的模式匹配芯片 (Pattern Matching Chip, PMC) 来处理, 该模式匹配芯片已经在国际专利申请 No. PCT/N099/00344 中公开, 在此一并引用。经预处理的文档集用 D_p 表示, 而相应的经预处理的文档分区分别用 $d_{p1}, d_{p2}, \dots, d_{pn}$ 表示。

20 基于软件的搜索系统要求从文档集中生成索引, 用 I 表示, 相应的文档集的文档分区 $d_1, d_2, \dots, d_n$ 用 $i_1, i_2, \dots, i_n$ 表示。

25 搜索文档集 D 的分割文文档 d 所需的数据组叫做文档分区所需数据组或者文档分区相关数据组。在只含软件的系统 (SW system) 中数据组就是索引 i_k , 而在具有硬件的系统 (SW/HW system) 中数据组还包括经预处理的相应索引为 i_k 的文档分区 d_{pk} , 其中 $1 \leq k \leq n$

本质上, 由软件实现的文档分割和预处理操作, 其步骤如下所示

30 $*(D) \longrightarrow *(d_1 \dots d_n) \longrightarrow *(d_{p1} \dots d_{pn}) \longrightarrow d_{pk}$, 其中 $*(D)$ 表示对输入文档集 D 的进行分割操作, $*(d_1 \dots d_n)$ 代表过滤操作, 比如给 $d_1 \dots d_n$ 添加

索引, 而 d_{pk} 无疑就是文档分区相关的数据组。该数据组在只有软件的系统 (SW) 中就是索引 i_k , $1 \leq k \leq n$ 。

5 搜索引擎用一组由高性能互联总线连接起来的工作站来实现。图中未显示的工作站组成了搜索服务器系统。工作站作为搜索服务器的节点。不通节点执行不同的搜索任务, 按照本发明, 在下文中进行陈述。

10 节点可以认为是分布在各个工作站上的虚拟节点, 但是为实现基于硬件的搜索节点, 具有硬件系统的搜索引擎, 其专用的搜索处理硬件必须是放置在某些工作站上的物理实体。尽管部分具有硬件系统的搜索引擎中地搜索节点可能只含有软件, 并任意分布在多个工作站上, 但搜索节点软件仍然可以是分布式的。

• 发送节点 $N_{a1}, \dots, N_{a,a}$ 处理输入的搜索请求并把搜索请求发送给所有的搜索节点。发送节点还可以设置成采集节点以便汇集搜索答案和搜索结果并返回给查询者。一旦接收到搜索答案, 处于采集模式的发送节点把搜索答案综合为最终的搜索结果。

• 搜索节点 $N_{\beta 1}, \dots, N_{\beta, b}$ 拥有整个数据组 d_p 的一部分。搜索节点包含专用的搜索软件和大量上文提到的用于硬件搜索的 PMC 模块。

20 • 索引节点 $N_{\gamma 1}, \dots, N_{\gamma, g}$ 负责为大量搜索节点上的专用搜索软件生成索引。对于 PMC 模块, 索引节点 N_{γ} 还对原始数据进行过滤和预处理。

• 采集节点 $N_{\delta 1}, \dots, N_{\delta, e}$ 可以任选地用于汇集结果并把搜索结果综合为最终结果, 这种情况无疑可以分离该任务中的发送节点。

25 按照本发明的第一个实施方案图示在图 1 中, 其中搜索节点 N_{β} 形成节点组或列 S, 搜索节点组或列 S 并行连接在发送节点 N_{α} 和索引节点 N_{γ} 之间。图中还图示了发送节点的排列原理图。而实际上, 这些节点形成了多级分层结构。

30 按照本发明的第二个实施方案图示在图 2 中, 其中采集节点 N_{δ} 的排列与发送节点的排列类似, 这样就分离了该任务中的发送节点。

可以理解, 单个工作站可能只作为某一特定类型的节点, 或者

多种其他类型的节点。换句话说不同类型的节点可能分布在工作站组中的各个工作站中。因此，图 1 和 2 所示的结构由整个组中的工作站来组成，而这些图中既不显示工作站也不显示互联总线。

5 接下来需要对节点进行更为详细的说明。根据本发明，处于起始位置的搜索节点是搜索引擎的关键节点。

搜索节点 N_β 包含上文提到的整个数据组 d_p 中的一部分。搜索节点具有软件搜索引擎 (SW) 和大量可选的 PMC 模块 M，如图 3 所示。随后，在索引节点 N_γ 中生成搜索节点 N_β 的数据组。

10 为能快速搜索，搜索节点可以配备 x 个 PMC 模块 M，如图 3 所示。每个 PMC 模块 M 具有 y 组含有 z 个 PMC 模块的 G，如图 4 所示，其中每组 G 从单个存储芯片 RAM 中接收数据。这些 A 模块 M 通常具有独立的电路板。每个 PMC 芯片能够同时处理 q 个查询，如图 5 所示。

15 模式匹配芯片 PMC 一秒钟能够处理 t_c 字节的数据容量。若存储模块每秒钟能够向典型匹配芯片 PMCs 传送 t_y 字节的数据，则 PMC 能够在给定的时间内搜索 T_c 字节的数据容量， $T_c = \min\{t_c, t_y\} t$ 。

如图 4 所示，典型匹配芯片组 PMC 置于模块 M 中，模块 M 含有 y 组 G，每组 G 又含有 z 个 PMC 芯片。其中，每组 G 从单个存储芯片 RAM 中接收数据。存储芯片的容量是 T_c 。模块能够搜索的总的数据量为 T_y ， $T_y = T_c$ 乘以 $z q$ 个不同查询。

20 若每个搜索节点 N_β 含有 x 个模块 M，由于没有任何 PMC 模块搜索相同的数据；同时查询数目仍是 $z q$ ，则这些 PMC 模块 M 能够搜索相当于 T_r 的数据量， $T_r = T_y * x = \min\{t_c, t_y\} txy$ 。

PMC 模块在搜索节点上的总的搜索速度可表大为：

$$r_{HW} = \frac{\frac{zq}{T_r}}{\min\{t_c, t_y\}txy} \quad (1)$$

25 其中 T_r 代表一个节点总的数据容量。接下来就可以计算搜索节点的性能。

给定的 PMC 模块 M (或者任何硬件设备) 具有 r_{HW} 的查询速度，也就是说搜索节点 N_β 上的搜索软件具有 r_{HW} 的查询速度，则搜索节点

N_s 的总的查询速度可表示为

$$r_{\Sigma} = r_{t.HW}(1 - \varphi_{SW}) + r_{t.SW}\varphi_{SW} \quad (2)$$

其中 φ_{SW} 代表在软件中执行的查询数 q 的百分比。 φ_{SW} 的实际值在运行中动态变化。

- 5 发送节点 N_a 接收所有的查询，并把查询发送给所有的搜索节点 N_{β} 。从不同搜索节点 N_{β} 得到的结果进行综合，并且，如果发送节点具备采集节点的功能，就可返回一个完整的结果。

- 索引节点 N_{γ} 汇集文档并为不同搜索节点 N_{β} 上的搜索软件建立预索引。因此索引节点 N_{γ} 能够与含有合适的索引软件的搜索节点 N_{β} 合并。
10 硬件是基于对全部原始数据的扫描，但是部分原始数据的预处理和过虑可由索引节点 N_{γ} 完成。

关于节点互联和数据通信，可以在下文中得到大体的了解。

不同类型的互联可以用于节点的连接。比如，在低端系统中，通常用 100M 字节带宽的快速以太网来处理通信。

- 15 互联节点中的通信可以分成两类：

- 查询通信 - 发送节点 N_a 和搜索节点 N_{β} 之间的通信，查询通信在整个搜索期内都在进行，查询通信的特点在于小到中等的

- 数据通信 - 索引节点 N_{γ} 和搜索节点 N_{β} 之间的通信，数据通信的特点是较高的数据容量和较低的通信频率（一般每天一批）。
- 20

- 通常，查询从发送节点 N_a 向搜索节点 N_{β} 传送一个查询字符串，然后搜索节点 N_{β} 作出响应并用一系列文档与查询字符串相匹配。可选节点 N_a 也可以查询搜索节点，查询文档的 URL 字符串，但这不是本发明的关键内容。
- 25

按照本发明，搜索引擎的结构能够根据上述条件，很容易对数据容量和性能两个维独自进行二维缩放。

- 增加文档分区 d 的数据组来实现数据容量的缩放，换句话说，可以向搜索节点 N_{β} 增加更多的节点组或列 S 。为处理更多的文档分区 d 的数据组，还可以增加索引节点 N_{γ} 和发送节点 N_a 。
- 30

性能缩放可以在搜索引擎结构中通过复制文档分区的数据组并相应增加搜索节点 N_β 的数目来实现, 即如图 6 所示增加 N_β 数目。当采用复制文档分区的数据组来缩放性能时, 每个搜索节点 N_β 都是搜索节点组 S 的一部分。这样, 就把搜索节点 $N_{\beta 1} \dots N_{\beta s}$ 连接到组 $S_{\beta 1} \dots S_{\beta s}$ 中, 其中 $v = \frac{s}{h_s}$, h_s 代表缩放因子。节点组 $S_{\beta, j}$ 包含搜索节点 $N_{\beta, j}$, $N_{\beta, j+1}$, $N_{\beta, j+2}$ 和 $N_{\beta, j+3}$, 如图 8 所示。

数据容量的缩放可能引起搜索节点 N_β 接收到的来自发送节点 N_a 的查询数的大量增加。本结构通过使用发送节点 N_a 的层数 λ 如图 7 所示来解决该问题。这个结构使发送节点 N_a 形成二进制数据分布树中的一部分节点。二进制数据分布树允许较容易地对其线性缩放。类似的二进制数据分布树已经在上述提到的国际专利申请 PCT/N099/00344 中公开, 该专利公开了一个典型匹配芯片 PMC 的实际实现方案的结构。规则二进制树中发送节点 N_a 的数目是第 λ 层为 $2^{\lambda-1}$ 个, $\lambda = 1, 2, 3 \dots$, 发送节点中的根节点在树结构中的第一层, 从顶层到 λ 层, 树中全部的发送节点数是 $2^\lambda - 1$ 。如果发送节点还作为采集节点使用, 以及用于汇集从搜索节点返回的结果, 则在发送节点上综合成搜索结果, 并由发送节点中的根节点向查询用户输出最后的查询结果。然而, 这并没有违背按照本发明的搜索引擎: 它建立的是独立的与搜索节点相连接的数据采集树, 包含采集节点 N_s , 这些采集节点从数据汇集树以及数据采集节点树的根节点上汇集和输出查询的最终结果。因而采集节点树可以作为发送节点的一个镜像。

图 8 中给出了按照本发明的一个可缩放搜索引擎结构的设计原理图, 该引擎结构具有二维缩放原理的插图。可以看出, 发送节点 N_a 组成了搜索引擎的前端结束部分, 向搜索节点 N_β 发送查询并从搜索节点接收查询结果, 在搜索节点中对索引数据进行了实际的搜索。如果使用了专用采集节点 N_s , 如图 9 所示, 而其它部分跟图 8 相似, 则搜索结果无疑将返回给这些专用采集节点。随着发送节点 N_a 加入到树形结构, 如图 9 所示, 作为搜索引擎的后端结束部分的采集节点网络形成一个发送节点网络的镜像。索引(蛛网状)节点 N_y 组成

了搜索引擎的后端结束部分，并从，如，因特网中汇集数据以及检索数据以生成一个可搜索目录。通过水平增加搜索节点 N_β 或者搜索节点组 S ，可以线性缩放搜索引擎的数据容量。每个增加的搜索节点或搜索节点组容纳不同的数据。通常的搜索引擎容量参数可以从下文的一个非限制性例子中给出。通常一个搜索节点 N_β 每天能够在 5,000,000 文档目录中处理 8,000,000 页示图。可缩放搜索引擎中，每个搜索节点 N_β 通常能够容纳 5,000,000 不同的索引文档，一个具有 40 个搜索节点的节点行足够用于维护 200,000,000 个文档目录。性能缩放以及通信能力的增加需要增加更多行具有相同数据容量的搜索节点 N_β ，它使每个搜索节点组或列 S 中的搜索节点能容纳相同数量的数据。因此一个具有 10 个 N_β 搜索节点的节点组或列 S 每天能够处理 80,000,000 页示图，则 40 个这样的节点列 S 每天能处理 3,200,000,000 页示图。

按照本发明的具有可缩放结构的搜索引擎的另一个重要优点是查询响应时间跟目录的大小完全无关，因为每个查询是在所有搜索节点 N_β 中并行处理的，而且该结构具有固有的容错能力，它确保了个别节点中的错误不会导致系统的崩溃，而只是暂时的在错误纠正之前减低了性能。

而且，按照本发明的数据容量和通行容量的无限线性缩放原理，可以应用于搜索引擎中。与现有的搜索引擎技术相比，后者的搜索成本通常随数据或通信容量的增加成指数增加，并且现有的搜索引擎技术的最大容量通常只能达到较低到中等水平。而按照本发明的搜索引擎，其成本随容量的增加最多成线性增加，事实上它只决定于容量的增加是否仅仅通过增加 SW 搜索节点还是通过另增 SW/HW 搜索节点来实现。最后，按照本发明的搜索引擎还具有以下优点，即实际中的每个节点可以由标准的低成本的商用 PCs 来实现，当然也可以由更昂贵的基于 $UNIX$ 的服务器比如现有的 Sun 或 $Alpha$ 计算机来实现。

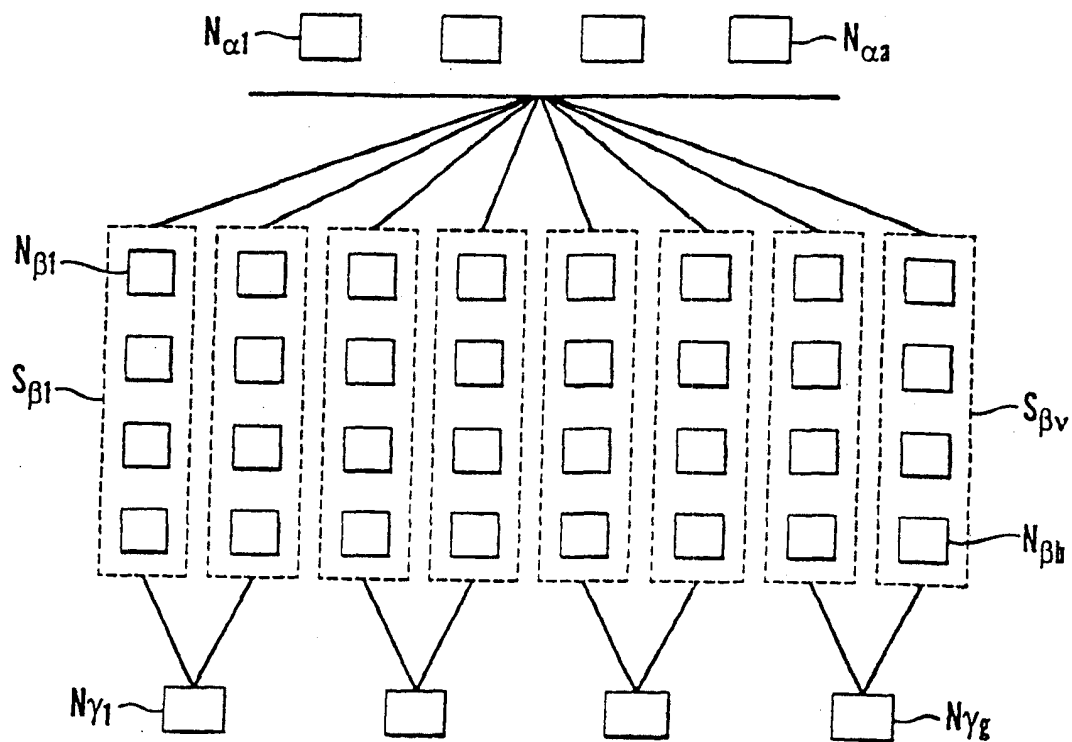


图 1

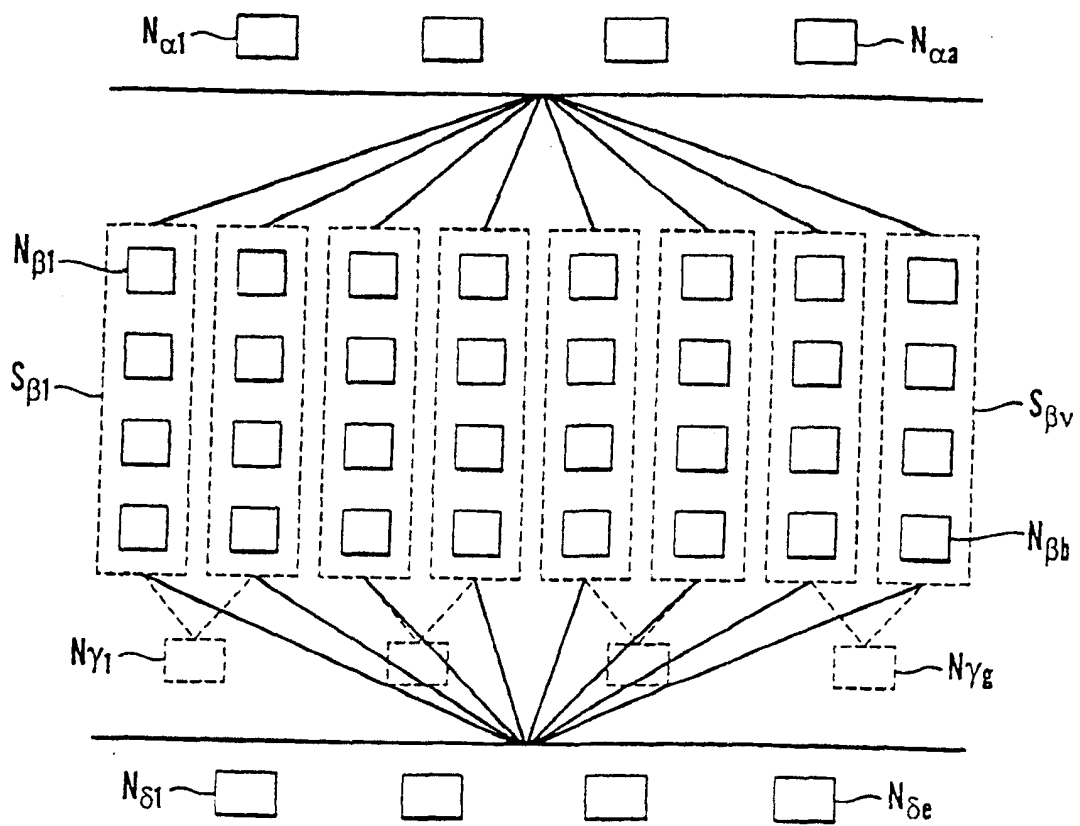


图 2

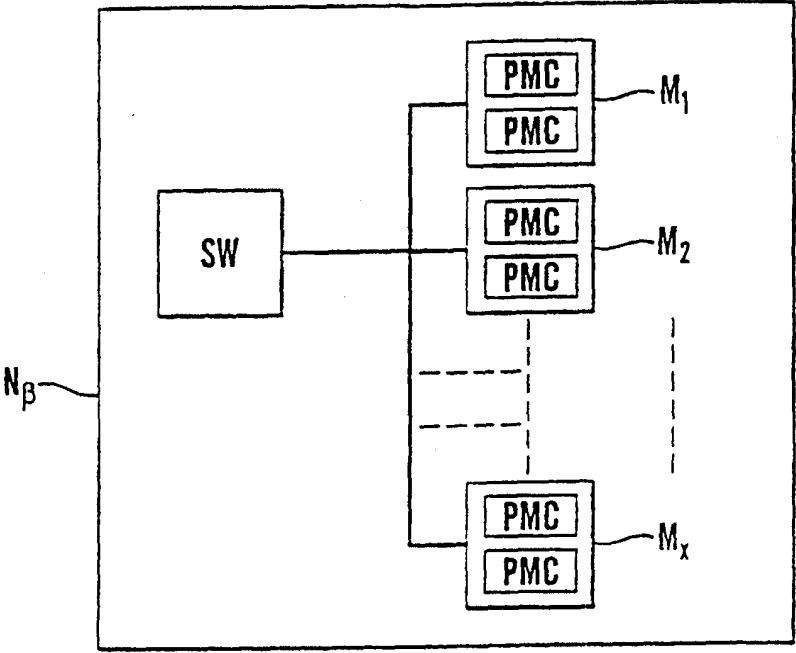


图 3

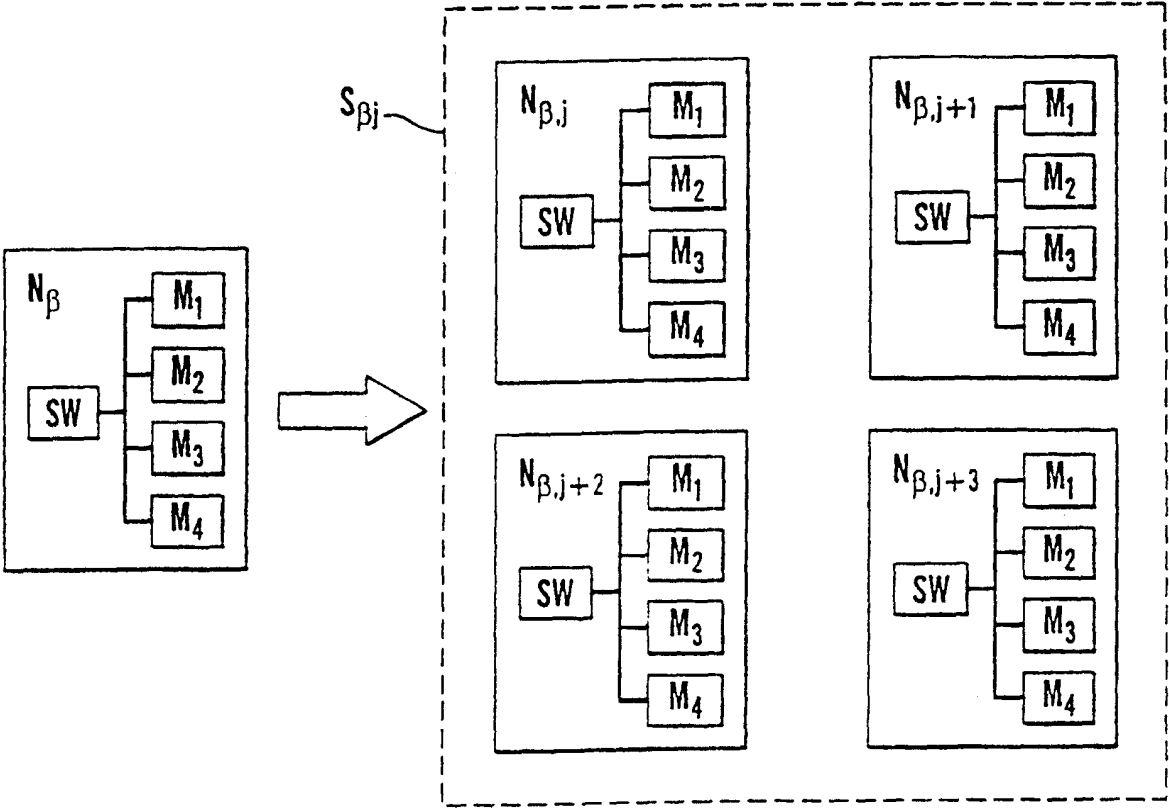


图 6

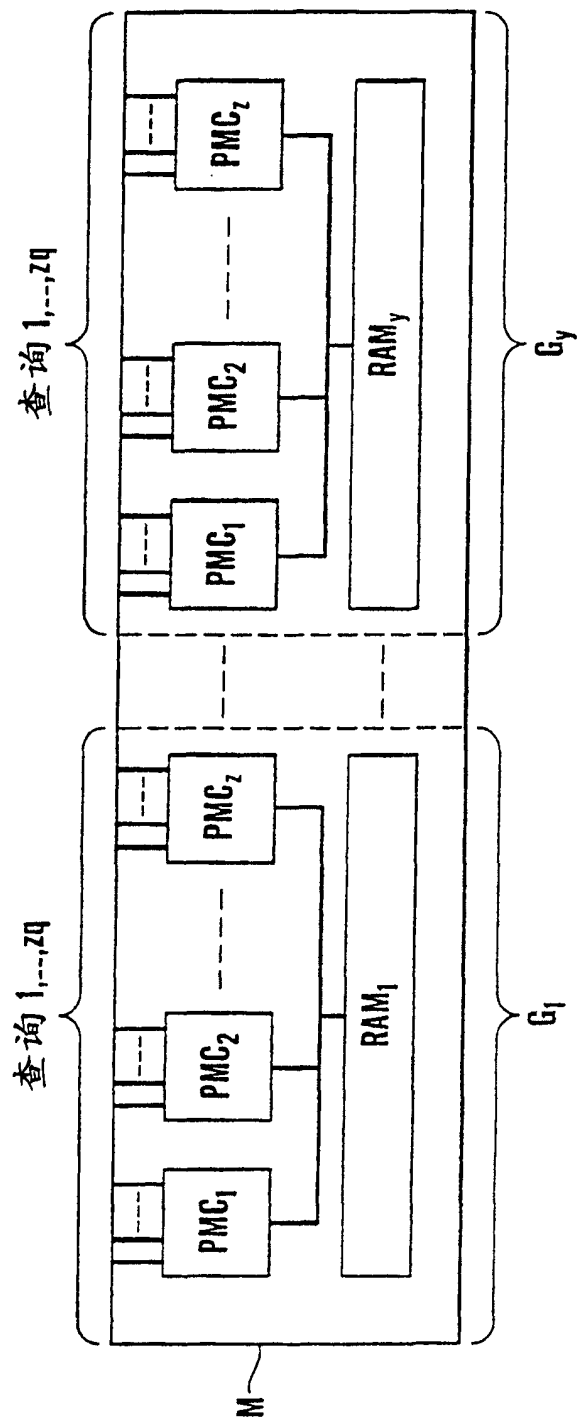


图 4

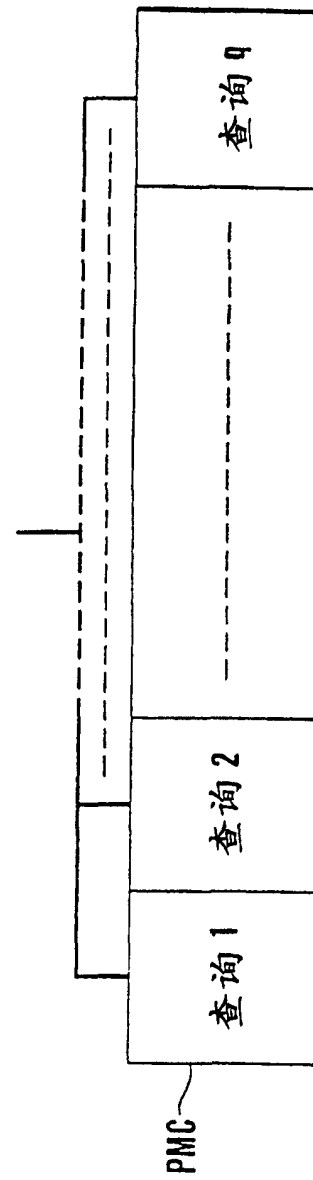


图 5

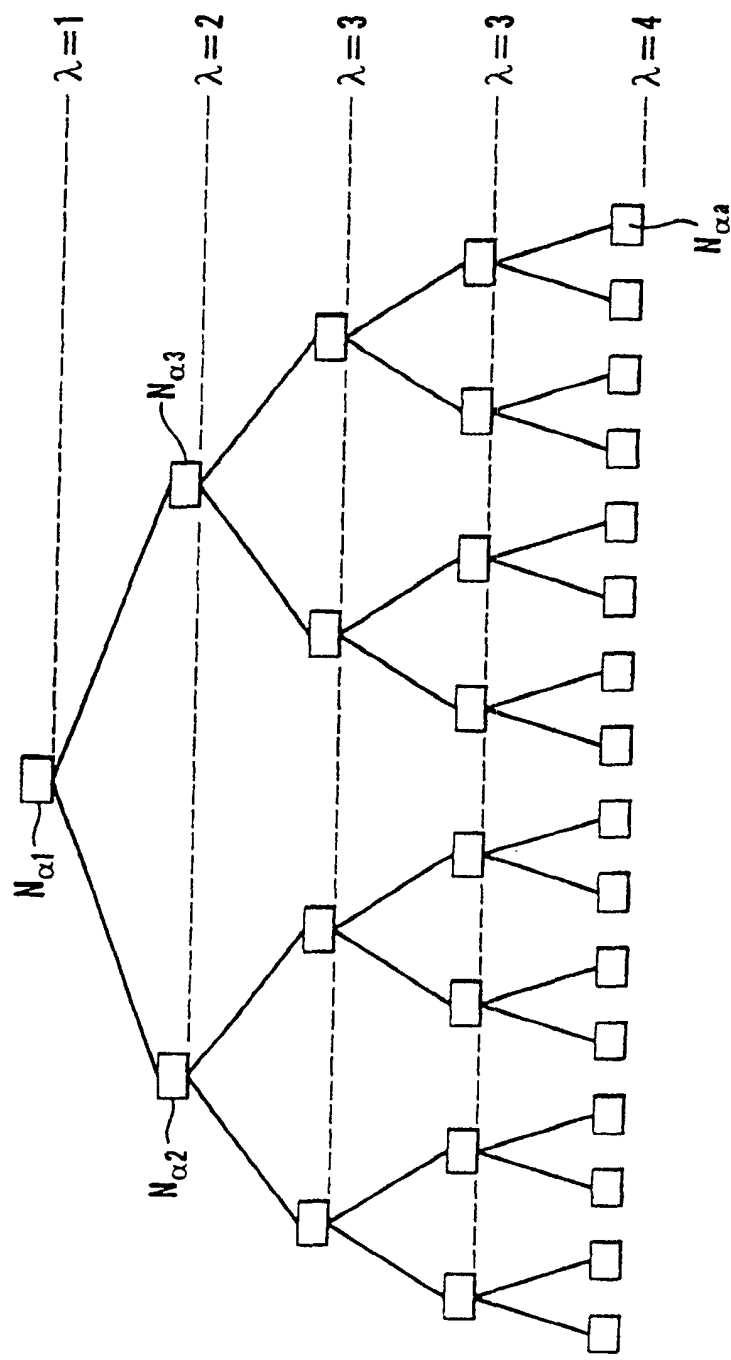


图 7

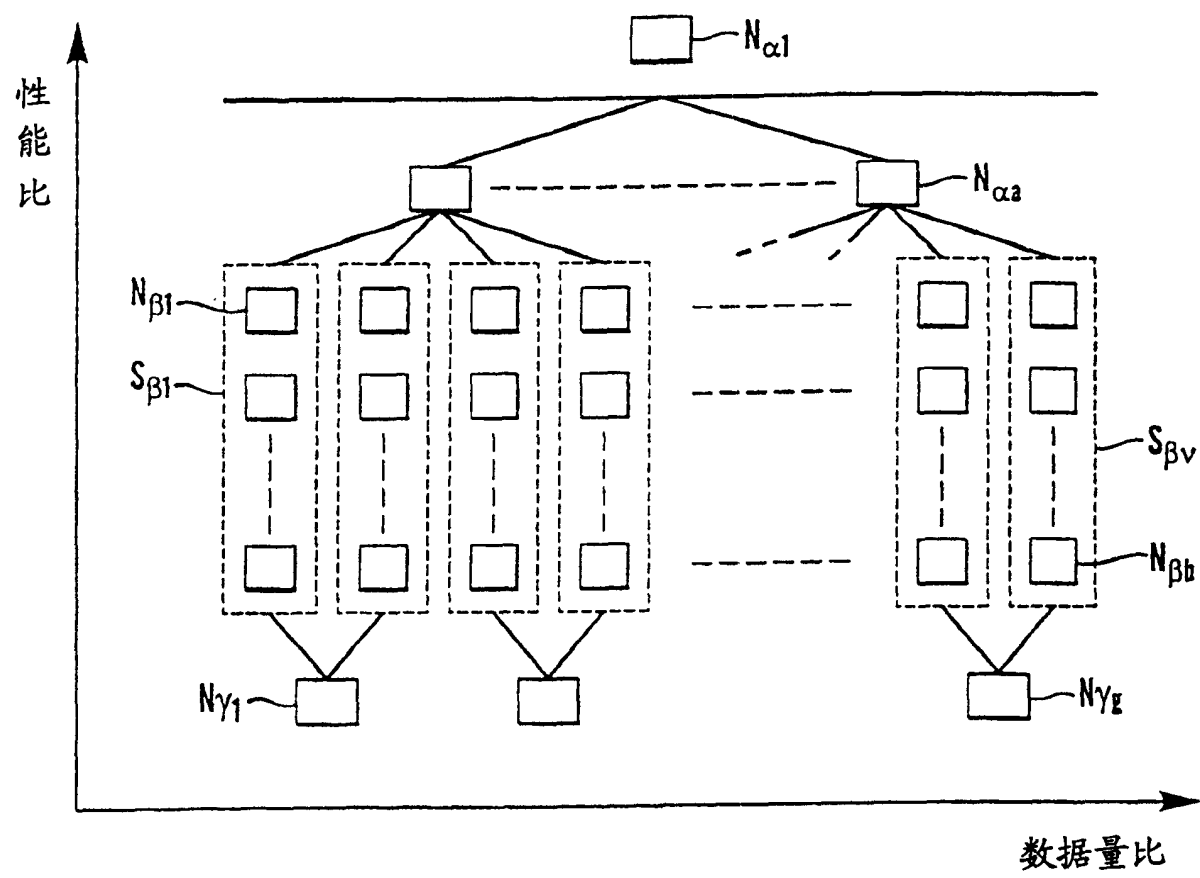


图 8

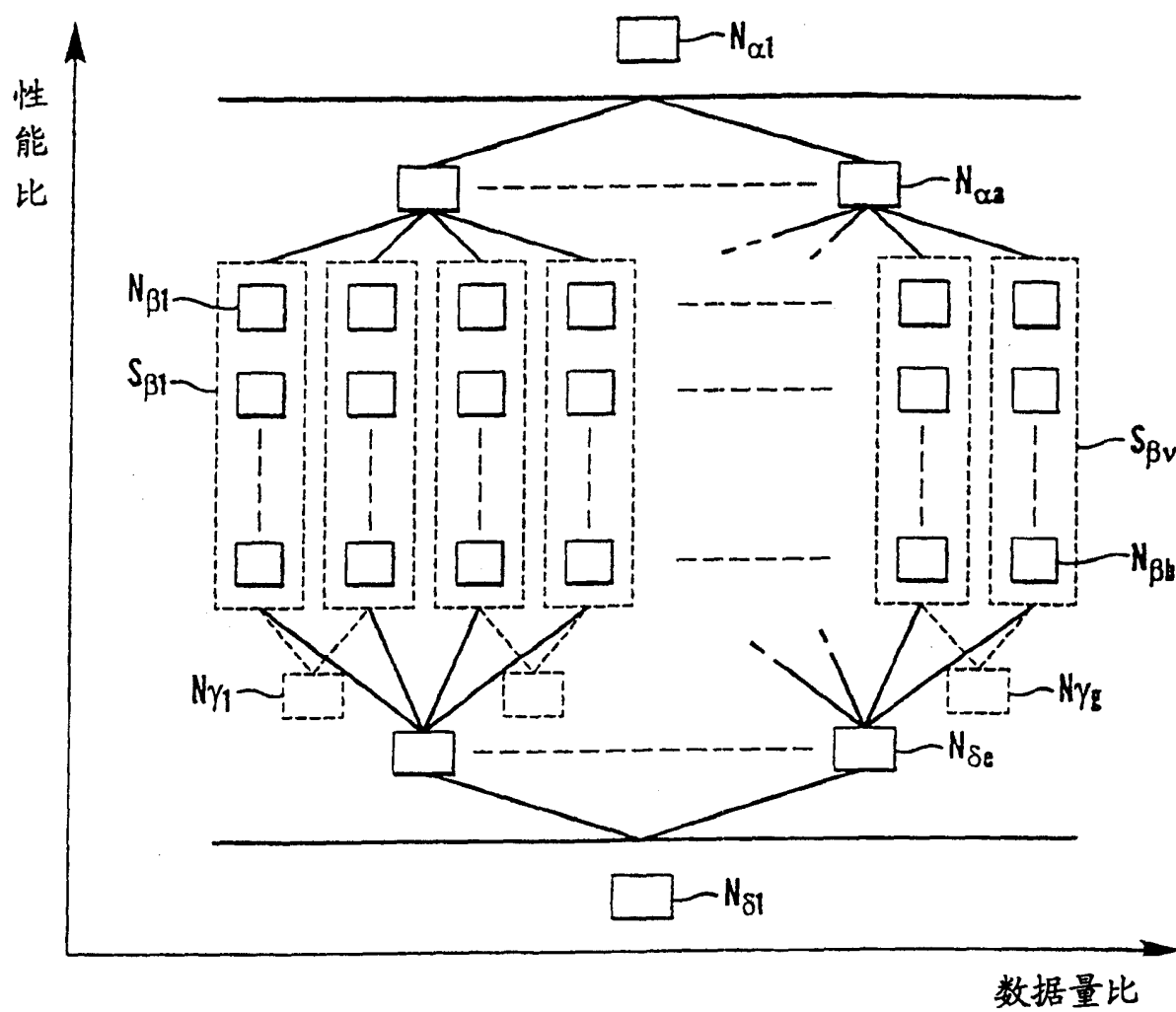


图 9