[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 11/18 (2006.01)
G06F 11/16 (2006.01)

[12] 发明专利说明书

专利号 ZL 02803967.X

[45] 授权公告日 2006 年 7 月 5 日

[11] 授权公告号 CN 1262924C

[22] 申请日 2002.2.28 [21] 申请号 02803967.X
[30] 优先权
[32] 2001.3.7 [33] EP [31] 01105702.3
[86] 国际申请 PCT/EP2002/002181 2002.2.28
[87] 国际公布 WO2002/071223 德 2002.9.12
[85] 进入国家阶段日期 2003.7.22
[71] 专利权人 西门子公司
地址 德国慕尼黑
[72] 发明人 奥利弗·凯泽
审查员 覃冬梅

[74] 专利代理机构 北京市柳沈律师事务所
代理人 马莹 邵亚丽

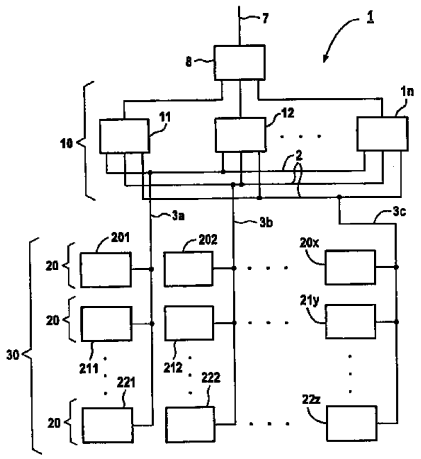
权利要求书 2 页 说明书 11 页 附图 3 页

[54] 发明名称

容错计算机装置以及运行该装置的方法

[57] 摘要

在一种根据本发明的计算机装置中，包括一个中继层(10)和一个处理层(30)，其中，该中继层(10)由至少一个中继计算机(11，12，13)形成，该计算机可以为到达的请求数据(7)设置一个时间标记；该处理层(30)由至少两个处理计算机(201，202，20x，211，212，21y，221，222，22z)形成，该计算机由中继层(10)并行地提供请求数据(7)，以及，然后各处理计算机都可以在下一个处理步骤中处理该请求数据(7)，如果所述时间标记的当前值分别在有效值范围内。通过这种方式，用简单的方法实现了计算机的“隐含同步”。



1. 一种容错计算机装置(1),包括一个中继层(10)和一个处理层(30),其特征在于,

所述中继层(10)由至少一个中继计算机(11,12,1n)形成,该中继计算机为从外界到达该中继层的请求数据(7)设置一个时间标记(46),

所述处理层(30)由至少两个处理计算机(201,202,20x,211,212,21y,221,222,22z)形成,这些处理计算机由中继层(10)并行的提供请求数据(7),以及

如果所述时间标记(46)的当前值分别在有效值范围内,则各处理计算机都在下一个处理步骤(C_0,C_1,C_2)中处理所述请求数据(7),使得所述处理计算机同步地处理该请求数据。

2. 根据权利要求1所述的计算机装置,其特征在于,通过所述中继层(10)形成一个面向外界的界面,该界面提取出从外界到达的请求数据(7),并将对应的计算结果(41)传送到外界。

3. 根据权利要求2所述的计算机装置,其特征在于,所述中继层(10)可以根据预定的有效性规则来估计由所述处理计算机计算的请求数据(7)的结果(54),根据该估计从该结果(54)中选择一个结果(41),并将其传送到外界。

4. 根据权利要求1至3中任何一项所述的计算机装置,其特征在于,所述中继层(10)由至少两个中继计算机(11,12,1n)形成,而每个中继计算机都可以探测至少另一个中继计算机(11,12,1n)的错误功能,并接管该至少另一个中继计算机的功能。

5. 根据权利要求1至3中任何一项所述的计算机装置,其特征在于,所述处理层(30)被划分为至少两个处理子层(20),该处理子层(20)分别由至少两个计算机构成,并用于分别处理预定的请求。

6. 根据权利要求5所述的计算机装置,其特征在于,每个处理子层(20)都对应所述中继计算机(11,12,1n)中的至少一个作为对其发出请求的中继计算机,并且该发出请求的中继计算机用来中继预定种类的请求数据。

7. 一种用于运行容错计算机装置(1)的方法,该计算机装置(1)包括一个中继层(20)和一个处理层(30),其特征在于以下步骤:

a）从外界到达的请求数据由中继层（10）的至少一个中继计算机（11，12，1n）读入，并被设置一个时间标记（46），

b）设置了时间标记（46）的请求数据（7）由中继计算机（11，12，1n）并行地传送到处理层（30）的至少两个处理计算机（201，202，20x，211，212，21y，221，222，22z）中，以及

c）如果该时间标记（46）的当前值在一个有效值范围内，则请求数据（7）分别由处理计算机在下一个处理步骤（C_0，C_1，C_2）中进行处理，使得所述处理计算机同步地处理请求数据。

8. 根据权利要求 7 所述的方法，其特征在于，由所述中继计算机（11，12，1n）根据预定的有效性规则估计在步骤 c）中由所述处理计算机计算的请求数据（7）的结果（54），并根据该估计从该结果（54）组中选择一个结果（41）。

9. 根据权利要求 7 或 8 所述的方法，其特征在于，到达的请求数据（7）由至少两个中继计算机（11，12，1n）并行地读入。

容错计算机装置以及运行该装置的方法

技术领域

本发明涉及一种计算机装置，包括多个计算机，这些计算机用硬件和/或软件这样逻辑连接，使得该计算机装置对外的功能不会由于一个或多个计算机的故障而受到影响，或仅受到不重要的影响（容错计算机装置）。

本发明还涉及运行这类装置的方法。

背景技术

在现代社会里，目前已经利用数字计算机实现了大量业务线路、通信连接、监控任务等。例如，通过因特网订购货物逐渐多地代替了不久前还很普遍的邮政信件订购。

在这类订购过程中，顾客利用其与因特网连接的计算机在提供服务公司的服务器中进行选择，以使用在此现有的订购软件进行其订购委托。

顾客在订购过程中没有注意到，同时或先后有多少不同的计算机对其订购过程进行处理；只要在此订购过程中不出现错误，顾客的订购状况就好像他只同一个作为“对话伙伴”的计算机在通信。

但是，如果在订购过程的某个步骤中出现错误，则顾客经常会通过以下方式注意到该错误，即，他必须再次输入已输入的信息，因为由于错误的出现而使得订购系统的某个计算机丢失了信息。

这种可通过因特网使用的订购系统已经公知，且每天都有上百万用户在使用。

这种系统的缺点在于，尽管该系统一般由多个计算机组成，但其中一个计算机的故障会导致整个计算机系统的故障，或至少导致部分功能的丢失，并由此导致信息和处理时间的损失。该缺点基于下述理由，即使用这种计算机装置（群集）一般要解决的任务是，将对该计算机系统提出的要求分配到多个计算机上（负荷分配），以便提高可同时处理过程的速度和数量。由此，在这种装置中，由于所期望的负荷分配而没有将待处理的要求同时引导到多个计算机上，且该装置的计算机不同步，该装置中一个计算

机的故障将不可避免地至少导致部分功能的丢失和/或信息丢失。

具有多个服务器的计算机装置例如在 EP 0942363 A2 中给出。其中，将输入的请求数据分配到服务器类中，然后由一定数量的服务器进行处理。如果此时一个确定的服务不能被处理，因为该时刻可供利用的计算机容量资源不足，则从其它服务器类中会分离出有可供利用的计算资源的服务器，并用于所要求的服务。

在所述欧洲公开文献中还描述了一种计算机集群，其中，在服务器上进行请求数据的负荷分配，从而在一个服务达到资源极限时，用另一个服务的、仍然具有可用计算容量的服务器来代替。

其中的缺陷是，对错误情况没有解决方法，使得尽管一个服务的故障不会导致整个所涉及服务的丢失，但不能保证，在发生错误的情况下还会获得传递到计算集群的请求数据，以及尽可能无间断的继续处理该数据。

因此，这种计算机装置不适用于不允许出现数据丢失和/或处理延迟的重要应用，以便使人和环境都不受到损害。因此，禁止采用这种装置作为例如核能站的监控系统、在危险处理（例如电或化学处理）中的保护系统，或具有与时间密切相关的过程的服务系统。

在 DE 19814096 A1 中，描述了一种用于转换冗余连接的同类组件的方法。

该同类组件中的一个组件作为主组件用于自动化处理。一个第二同类组件处于所谓的从属模式（保留），以便能在主组件出现故障时接管其功能。

两个同类组件都由一个置于其上的装置同步地提供相同的请求数据。

在主组件出现错误时，处于从属模式的组件直接在置于其上的装置的处理下被激活，以便接管主组件的功能。这样就保证了在出现错误的情况下从有缺陷的组件迅速转换到能工作的组件。

但是还不能识别，如何在出现错误的情况下可以保证不丢失请求数据，并且保证在错误情况下接管功能的组件提供了正确的任务数据。

此外，在这种根据现有技术的方法中，存在的缺点是组件必须是相同种类的。由此，采用互不相同的组件就不能具有相同的用于解决问题的功能，这在实现这类冗余装置时会导致很高的费用。可以将例如主计算机（主机）实施为功能极其强大的计算机，而将保留计算机（从机）实施为功能较弱的计算机。正常情况下，功能强大的计算机接管计算机装置的功能，

而只有在出现错误的情况下计算功能才会出现轻微的损失（如果保留计算机接管功能）；但是，这种与引用的现有技术相比花费更少的计算机装置用所描述的方法无容错的运行。

在 WO 98/44416 中描述了一种容错计算机系统。

该系统包括例如 4 个或多个 CPU，它们钟同步地工作。

到达的数据由所有 CPU 同时时钟同步地处理。

CPU 将其计算结果传送到处理单元，该单元确定出该结果的有效性，并输出有效的结果。

在该系统中，容错性实际只能在硬件中实现。这意味着相互完全同类的单元（CPU）绝对同时（时钟同步）地处理相同的输入数据，并提供所属的结果。因此，一个单元的故障不会导致整个系统的失效。

其中的缺点在于，这种时钟同步工作的解决方法花费极大，因为时钟同步运行对所应用的硬件提出了很高的要求，此外这些硬件必须相互是完全相同的类型；其中实际上不允许容错。此外，同步所应用的单元花费极大，因为并联的单元在处理请求数据时甚至不允许相差一个时钟。此外，也不能采用种类互不相同的硬件，以实现根据该现有技术的冗余。

根据现有技术的这类只能在硬件中实现冗余的冗余系统的其它例子是所谓的西门子 SIMATIC 自动化系列中的 H 系统（高度可控系统）（例如 s5-155h，s7-400h）。其中，各使用两个相互完全相同的、特殊的中央单元，其各自时钟同步地、并行地处理相同的请求数据。在中央单元出现故障时的同步和监控花费极大，此外购置费用也极高。

发明内容

本发明要解决的技术问题是，提供一种容错计算机装置，其克服上述缺陷，也可以灵活地由不同的组件组合，并可以低廉的制造。

根据本发明，该技术问题是通过一种具有中继层和处理层的容错计算机装置来解决的，其中

- 该中继层由至少一个中继计算机形成，该计算机可以为到达的请求数据设置一个时间标记，

- 该处理层由至少两个处理计算机形成，该计算机由中继层并行的提供请求数据，

- 如果时间标记的当前值分别在有效值范围内，则然后各处理计算机都可以在下一个处理步骤中处理该请求数据。

在根据本发明的装置中，由该装置处理为一个结果的请求数据被传送到中继层（中介）的一个或多个计算机。

在此，中继层为到达的请求数据设置时间标记，该标记可以例如是时钟元件的当前时间信号或一个序列号，该序列号包含请求数据抵达中继层的时刻。

请求数据在必要时由中继层进行预处理，并与所述的时间标记一起并行传送到处理层的处理计算机。如果处理层由多个处理分层（处理子层）组成，它们各由至少两个计算机构成，并特定为处理各自确定的要求类型，则根据其类型由中继层将请求数据传送到所述处理子层的所涉及的计算机中。

因此，中继层的基本任务在于，对将由根据本发明的计算机装置待处理为一个结果的请求数据设置输入时间辍，并传送到处理层的计算机中，该计算机然后将请求数据处理为一个结果。

根据本发明，与一个处理计算机的故障相对的装置的容错是这样实现的，即，不仅象现有技术的很多解决方法（所谓的集群解决方法）那样将请求数据继续传送到处理层的一个计算机中，而且还传送到处理层的所有计算机中。

这样就保证了，如果处理层中的一个计算机出现故障，该层中的请求数据不会丢失。

然后，各处理计算机借助请求数据所具有的时间标记的当前值确定，是否由各处理计算机在下一个处理步骤中处理该请求数据。

由此防止了，到处理层的非同步并行信息传送导致处理计算机确定出不同的回答作为请求数据的结果。

处理计算机这样处理时间标记的当前值，即，其确定时间标记的当前值是否在一个各自的有效值范围内。

处理层的计算机典型的在多个处理步骤中处理请求数据，这些步骤可循环的相互跟随。这意味着例如每 100ms 完成一个新的处理步骤。根据本发明，由处理计算机在一个确定的处理周期内仅处理这样的数据，即其时间标记的当前值在一个各自的有效值范围内；随后的处理周期可以例如由

这些时刻组成，即这些时刻早于下一个扣除了最大延迟时间的处理周期的起始时刻，为了将数据从中继层传送到处理层的计算机中，该延迟时间非常必要。

通过这种方法，保证了处理层的计算机至少在开始其处理步骤的时刻同步处理请求数据。此外，还防止了处理计算机由于从中继层传送时的时间延迟而互相独立，即异步地处理请求数据，并由此计算出关于该请求数据的不同结果。

根据本发明的所述同步被称为“隐含同步”。在此，不需要各计算机相互绝对时钟同步地工作。在此更有决定意义的是，处理计算机仅这样被同步，即，其分别只处理这样的请求数据，该数据肯定到下一个处理步骤的起始时刻就传送时间可以被从中继层传送到处理层的所有计算机中。

如果例如在检查中通过处理计算机证明，请求数据关于下一个处理步骤的起始时刻而太晚进入中继层（这可以通过处理时间标记确定），也就是说，请求数据可以不是可靠地到下一个处理步骤开始时被传送到处理层的所有计算机中，则请求数据在下下个处理步骤中才由处理计算机进行处理。通过这种方式，根据本发明的装置的冗余这样被保证，处理层的所有计算机都处理请求数据，且其中一个计算机的故障不会导致数据或结果的丢失。

在本发明的优选结构中，通过中继层形成根据本发明计算机装置的面向外界的界面，该界面提取出从外界到达的请求数据，并将所属的计算结果传送到外界。

通过这种方式，对于从外界利用一个要求进入该计算机装置、并打算获得一个结果的用户，该计算机装置就象是一个惟一的计算机。输入数据和输出数据都仅由一个界面传输。

在本发明的另一个优选结构中，中继层可以根据预定的有效性规则来估计由处理计算机计算的请求数据的结果，根据该估计从结果中选择一个结果，并将其传送到外界。

在根据本发明的计算机装置中，由于冗余的原因而由多个处理计算机处理请求数据。作为请求数据的结果，应当只有其中一个结果被传送到外界；用户应当获得清楚的结果，而不必从不同的结果中选择一个。

因此，借助预定的有效性规则，中继层判断所述结果中的哪个结果将被传送到外界。

有效性规则可以例如在于，中继层对由处理计算机计算出的请求数据的结果进行互相比较，并确定如何使多个结果一致。如果一致结果的数量大于不一致结果的数量，则从一致结果的组中选择其中一个结果，并作为有效结果传送到外界。

该有效性规则还可以这样加强，即，例如只有当处理计算机的所有结果都一致时才将一个有效结果传送到外界。这样就最大限度地保证了结果是正确的。

优选的，中继层由至少两个中继计算机形成，每个中继计算机都可以探测至少另一个中继计算机的错误功能，并接管其功能。

通过这种方式，根据本发明的计算机装置的冗余也可以关于中继层的功能实现。这在中继层的计算机出现故障时至少另一个中继层的计算机对其识别。接着，该计算机接管出现错误的中继计算机的功能，并处理到达的请求数据。

所述在中继层的错误识别可以例如这样实现，中继计算机彼此交换周期性信号（心跳，看门狗），这些信号在连续的现有信号上被检查。如果对于中继层的一个计算机没有出现一个这样的信号至少一个时钟周期，则所涉及的计算机被识别为出现错误，其功能将由中继层的另一个计算机接管。优选地，通过通信总线连接中继计算机，并在通信总线上传送请求数据。通过这种方式，中继层的每个计算机都有权使用请求数据，从而在出现错误的情况下可以由其中一个计算机代替另一个计算机。

在本发明的另一个优选结构中，处理层被划分为至少两个处理子层，该处理子层分别由至少两个计算机构成，并用于分别处理确定的要求。在该优选结构中，每个处理子层都特定为分别处理一个确定种类的请求数据。由于每个处理子层都由至少两个计算机构成，因此，其中一个计算机的故障不会导致所涉及功能的丢失。

处理子层的构成保证了请求数据在处理计算机上的负荷分配，从而可以很好的利用可供使用的计算机负荷。

优选地，为每个处理子层都设置了至少一个中继计算机作为其要求中继计算机。

在本发明的该优选结构中，进一步改进装置内的负荷分配，因为也根据任务使用了中继计算机。如果对于每个任务（请求数据类型）都设置至

少两个计算机作为要求中继计算机，则也可在中继层中为每个任务实现冗余。

本发明进一步导致一种用于运行具有一个中继层和一个处理层的容错计算机装置的方法，其步骤如下：

1. 到达的请求数据由中继层的至少一个中继计算机读入，并被设置一个时间标记。

2. 具有时间标记的请求数据由中继计算机并行地传送到处理层的至少两个处理计算机中，以及

3. 如果该时间标记的当前值分别在一个有效值范围内，则请求数据由处理计算机分别在下一个处理步骤中进行处理。

在本发明的优选结构中，由中继计算机根据预定的有效性规则估计步骤 3 中由处理计算机计算的请求数据的结果，并根据该估计选择其中一个结果。

在本发明的另一个优选结构中，请求数据由至少两个中继计算机并行地读入。

附图说明

下面详细示出了本发明的 3 个实施方式。其中：

图 1 为根据本发明的计算机装置，具有多个中继计算机以及形成为多个处理子层的处理层，

图 2 为根据本发明的计算机装置，具有一个中继计算机和两个处理计算机，其中详细示出了隐含同步，

图 3 为用于显示请求数据的时间变化的时间图，该请求数据由中继层传送到处理层。

具体实施方式

图 1 示出了一个计算机装置 1，包括一个中继层 10 以及一个处理层 30。

中继层 10 包括中继计算机 11、12、1n，请求数据 7 通过一个中继模块 8 传送到这些中继计算机上。在此，中继层的所有计算机都并行地获得相同的请求数据。

中继计算机 11、12、1n 通过通信连接 2 彼此相连。在每个通信连接 2

上都借助通信连接 3a、3b、3c 至少分别连接了每个处理子层 20 的一个处理计算机。

由多个处理计算机 201、202、20x、211、212、21y、221、222、22z 形成的每个处理子层 20 用于分别处理一个确定种类的请求数据 7；每个处理子层 20 都特定为处理一个确定任务。

通过通信连接 2、3a、3b、3c 的实施方式，防止了其中任何一个通信连接 2、3 的故障导致数据丢失和/或由处理子层 20 提供的任何一个功能的丢失。如果例如通信连接 3a 出现故障，则处理子层 20 可以借助通信连接 3b 来获得请求数据。

此外，任何一个中继层或处理层的计算机出现故障都不会导致信息和/或功能的损失，因为这些信息和/或功能都是如下多重地被实施的：

每个将在其中一个处理子层 20 中处理的任务都将在那里由多个处理计算机处理，从而其中一个计算机的故障不会导致各处理子层 20 的功能损失。

此外，其中一个中继计算机 11、12、1n 的故障也不会导致中继层 10 的功能损失，因为请求数据 7 借助中继模块 8 被传送到中继层的所有计算机中，且每个中继计算机 11、12、1n 由于通信连接 2、3a、3b、3c 的特殊结构而有权获得处理子层 30 的所有处理计算机的数据。因此，中继层 10 的一个中继计算机的功能损失通过另一个中继计算机接管其功能来平衡。

因此，在中继层 10 以及中继层 30 的每个处理子层 20 中，每个计算机都分别工作到各自出现故障和/或错误为止，而根据本发明的计算机装置 1 的总功能不会损失。

然后在下一个处理步骤中，如果该时间标记的值在一个有效值范围内，则每个处理子层 20 的计算机都对具有时间标记的请求数据进行处理。如果情况并非如此，则计算机首先将处理移置到下下个处理步骤。在此，处理步骤可以周期性连续（处理周期作为工作步骤的特殊情况）。

优选地，处理计算机的处理步骤分别在相同时刻开始，从而尽管处理计算机没有时钟同步，但至少由于其处理步骤的一起开始而同步（这包括在概念“隐含同步”中）。

类似于处理计算机的所述隐含同步，中继层 10 的计算机 11、12、...、1n 同样可以互相被隐含同步。

图 2 示出了根据本发明的计算机装置，包括一个由一个中继计算机 40

形成的中继层 200 和一个包括两个处理计算机 51、52 的处理层 300。

请求数据 42 被读入到输入模块 43a 中。时间信号给定器 44 将一个时间标记 46 传送到该输入模块 43a。

在输入模块 43a 中，为请求数据 42 设置时间标记 46，并作为时间标记了的请求数据 48 被并行地发送到两个处理计算机 51、52 中。在处理计算机 51、52 中，处理元件 511 将时间标记 46 与请求数据 42 分开，并将该请求数据 42 传送到应用程序模块 61。

被分开的时间标记 46 由处理元件 511 传送到处理模块 53 中。在该处理模块 53 中，时间标记 46 紧接着受到检查，以确定其值是否在一个有效值范围内。这例如只有这样才成立，即，如果在处理计算机的下一个处理步骤开始之前，请求数据 42 就这样早地堆积在中继层 200 中，即请求数据 42 在由中继层 200 传送到处理计算机 51、52 之后（为此最多需要一个最大中继延迟时间）就到达所有处理计算机。优选地，每个处理计算机中各自的下个处理步骤的开始时刻对于所有处理计算机都相同。

通过这种方式，实现了“隐含同步”，并防止了处理计算机 51、52“分开运行”，也就是在处理步骤的起始时刻处理不同的数据，并由此提供不同的结果。

如果时间标记 46 的当前值在一个有效值范围内（例如如上所述），则处理模块 53 为应用程序模块 61 给定一个控制信号 55，从而该模块处理请求数据 42，并将产生的结果数据 54 发送到中继计算机 40 的输出模块 43b 中。如果处理模块 53 在处理时间标记 46 时确定其当前值没有在一个有效值范围内，也就是例如请求数据不是肯定直到处理计算机的下一个处理步骤开始为止就被传送到所有处理计算机中，则处理模块 53 对下下个处理周期才产生一个控制信号 55，从而应用程序模块 61 在下下个处理周期中才对请求数据 42 进行处理。这适用于所有参与的处理计算机 51、52，从而这些处理计算机在这方面同步。

处理计算机 51、52 将由其计算的结果数据 54 传送到中继计算机 40 的输出模块 43b。在输出模块 43b 中，结果数据 54 被估值，并由此输出一个结果 41。

输出模块 43b 中的估值可以例如与处理计算机 51、52 提供的结果数据 54 相比较。如果两个结果相同，则输出模块 43b 输出任何一个结果作为最

终结果 41。

如果现在两个结果不一致，则结果 41 例如可能出现了错误。如果处理计算机 51、52 中的一个在出现错误的情况下根本不能提供结果数据 54，则输出模块 43b 选择可以运行的处理计算机的结果数据 54 作为最终结果 41。

图 3 在时间线 I 中示出了请求数据 70、80、90 到达的时间变化，这些数据由中继层根据数据到达的时间分别设置了一个时间标记，并被传送到处理层的所有处理计算机中。请求数据 70、80、90 以确定的时间间隔先后到达中继层。

时间线 II、III、IV、V 分别对应处理层的计算机，其中时间线 II、III、IV 示出了作为具有时间标记的请求数据 75、85、95 的请求数据 70、80、90 到达的时间。

时间线 V 示出了处理时刻 t_{70}、t_{80}、t_{90}，在该时刻原始请求数据 70、80、90 由处理计算机进行处理。时刻 t_0、t_1、t_2、t_3 是处理计算机的处理步骤 C_0、C_1、C_2、C_3 的起始时刻。处理步骤可以周期性的相互尾随。例如，100ms 是一个处理周期持续时间的典型大小；但是也可以是其它，尤其是更短的周期持续时间。

最大传送延迟时间 t_s 是这样一种时间间隔，即为了将请求数据 70、80、90 可靠地发送到处理层的所有计算机所需要最大时间，，即使处理计算机和中继层之间的通信连接和/或处理计算机彼此不是同一种类，尤其是相差很大。优选地，最大传送延迟时间 t_s 包括一个时间保留，从而在不利的传送情况下，从中继层到处理层的数据传送也不会比最大传送延迟时间 t_s 持续得更长。

请求数据 70 由中继层设置了一个时间标记，并作为请求数据 75 被传送到中继层。如图 3 所示，请求数据 75 在不同时刻到达处理层的 3 个计算机。时间线 III、IV 示出，请求数据 75 在下一个处理步骤 C_0 开始时刻 t_0 没有及时到达三个处理计算机中的两个；请求数据 75 的时间标记没有在一个有效值范围内。由于这个原因，处理层的计算机在时刻 t_{70} 才对请求数据 75 进行处理，这与下下个处理步骤 C_1 的开始对应。通过这种方式保证了，请求数据 75 冗余地由多个、尤其是所有处理计算机进行处理。

请求数据 80 及时地在处理步骤 C_2 开始之前就被发送到处理层的所有计算机中，从而这些计算机在数据 80 到达之后的下一个处理步骤 C_2 的开始

时刻就已经对时间标记的请求数据 85 进行处理。因此，在上述情况下，请求数据 85 的时间标记在一个有效值范围内，从而由处理计算机在下一个、在请求数据到达时刻之后的处理步骤 C_2 进行处理。

请求数据 90 同样由中继层设置时间标记，并被传送到处理层的计算机中，在此请求数据及时的在下一个处理步骤开始之前到达处理计算机中的两个。但是，在处理层的第三个计算机中出现了一个错误，从而请求数据 95 不能由该计算机进行处理。但是，请求数据 95 的时间标记落在一个有效值范围内，因为请求数据及时的在下一个处理步骤开始前就到达可以运行的计算机，从而这些计算机在下一个处理步骤 C_3 的开始时刻 t_{90} 对请求数据 95 进行处理。尽管处理层的一个或多个计算机出现了错误，请求数据 95 还是由处理层中可以运行的计算机进行处理。也就是说，错误不会导致数据或计算结果的丢失。

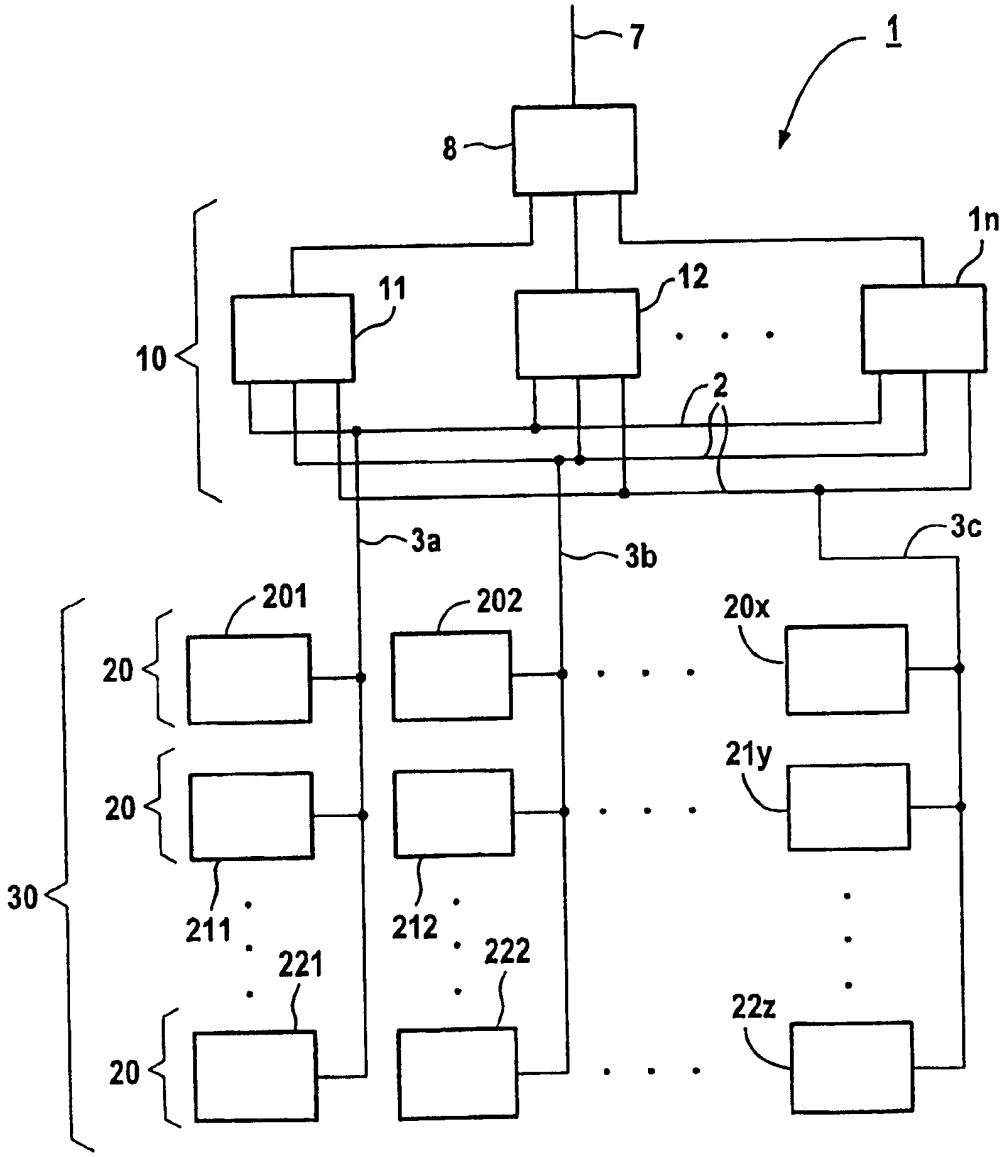


图 1

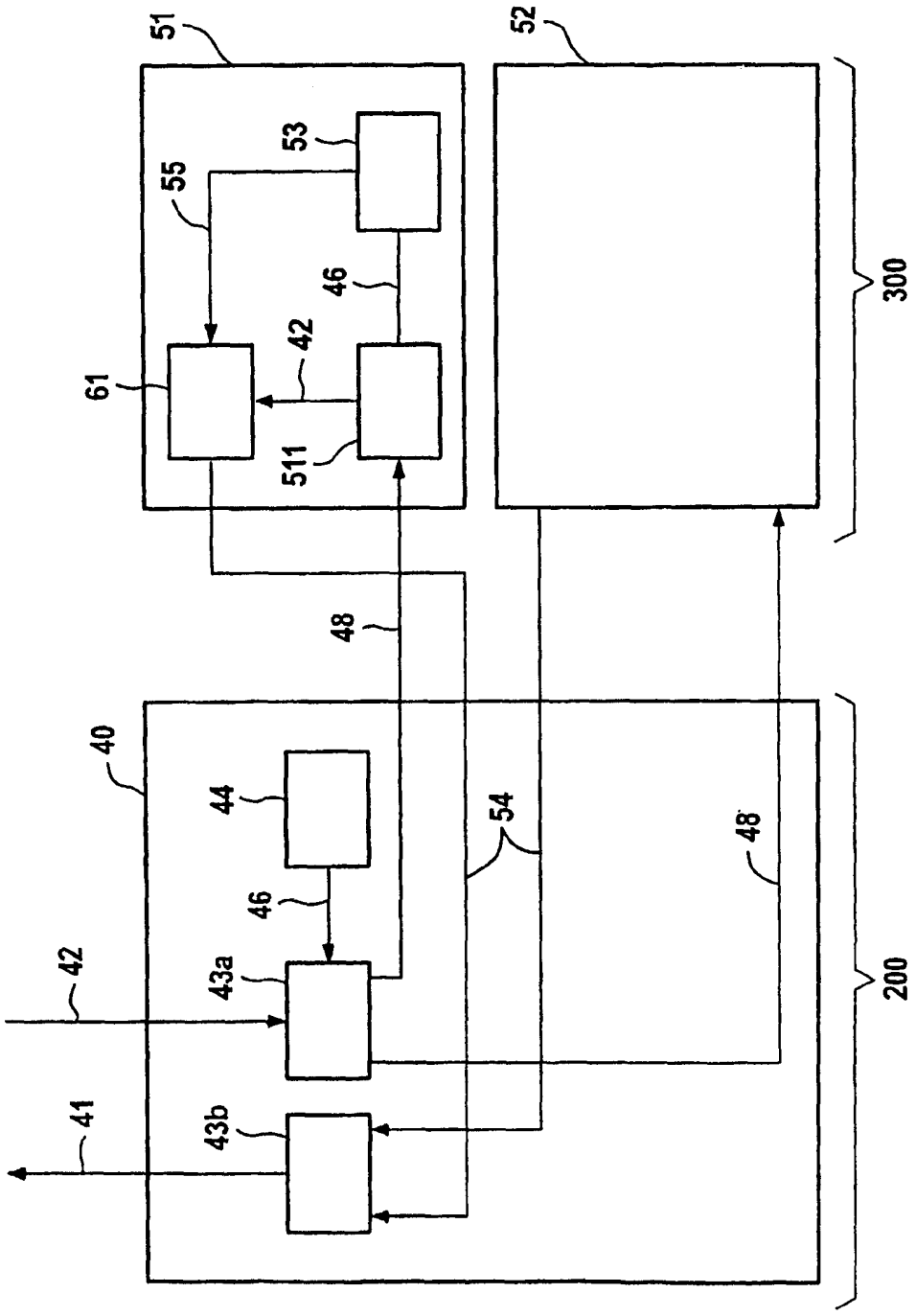


图 2

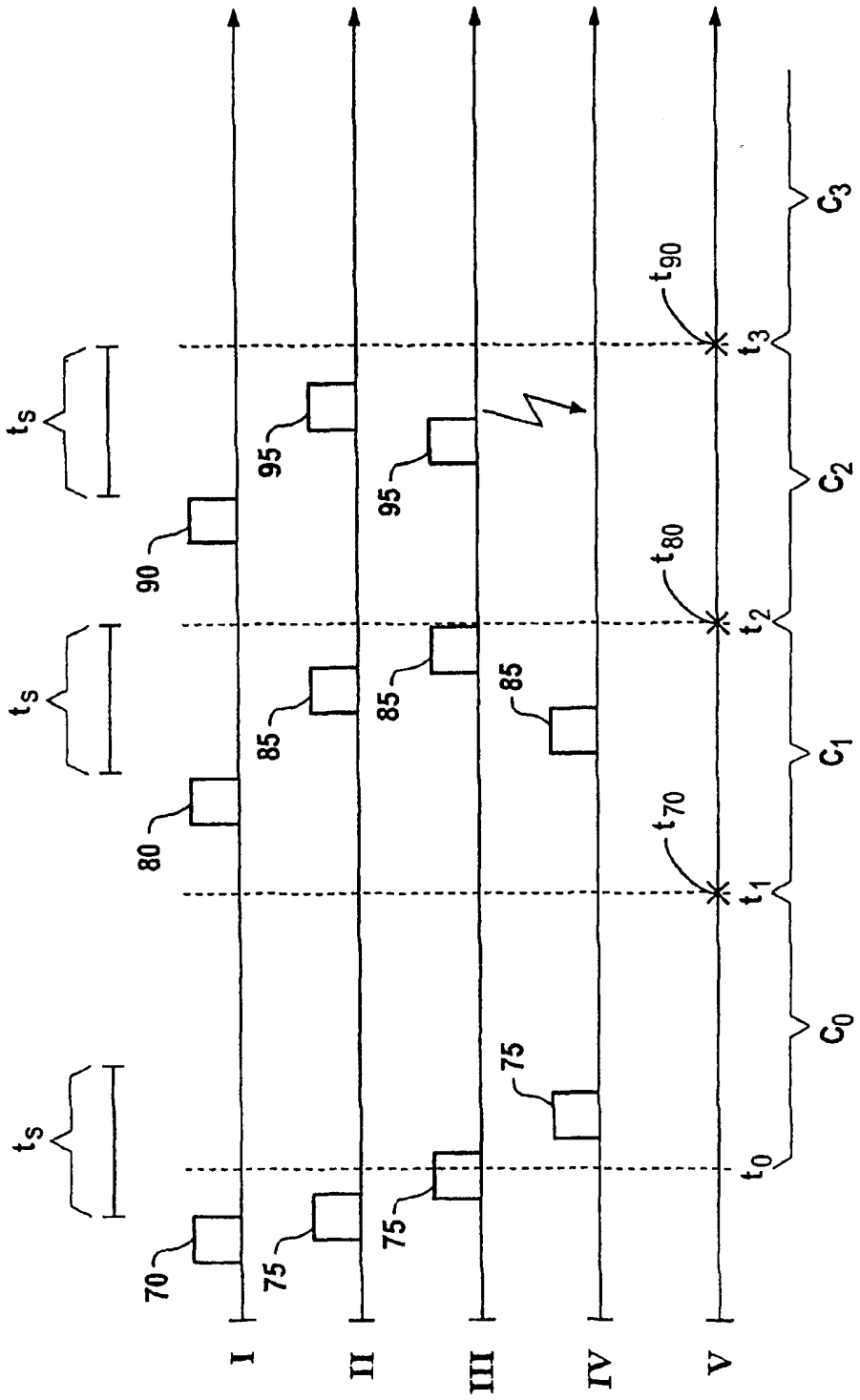

I
II
III
IV
V
70
75
75
80
85
85
90
95
95
75
85
t_S
t_S
t_S
t_0
t_1
t_2
t_3
t_{70}
t_{80}
t_{90}
C_0
C_1
C_2
C_3

图 3