(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115129465 A
(43)申请公布日 2022.09.30

(21)申请号 202210299443.X

(22)申请日 2022.03.25

(30)优先权数据
102021203061.0 2021.03.26 DE

(71)申请人 罗伯特·博世有限公司
地址 德国斯图加特

(72)发明人 A·哈曼 D·N·达萨利 F·雷姆

(74)专利代理机构 中国专利代理(香港)有限公司 72001
专利代理师 韩瑞 司昆明

(51)Int.Cl.
G06F 9/50(2006.01)
G06F 13/28(2006.01)
G06M 1/27(2006.01)

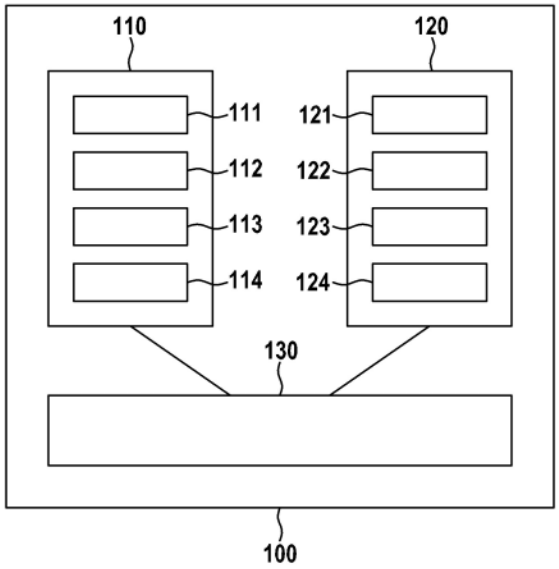
权利要求书2页 说明书10页 附图2页

(54)发明名称

用于运行计算单元的方法

(57)摘要

本发明涉及一种用于运行计算单元(100)的方法,在计算单元中执行多个进程,所述进程按照预定的资源分配共同地访问资源(130),所述方法包括:在执行多个进程中的至少一个进程期间确定(220、230)多个进程中的至少一个进程的当前的实际处理时间,多个进程中的至少一个进程在所述实际处理时间内被处理;将所述实际处理时间与指派给多个进程中的至少一个进程的额定处理时间和/或与所指派的额定处理时间和指派给多个进程中的至少一个进程的处理时间延长的和进行比较(230);和根据这种比较的结果调整(240、250)预定的资源分配。



CN 115129465 A

1.用于运行计算单元(100)的方法,在计算单元中执行多个进程(111、112、113、114、121、122、123、124),所述进程按照预定的资源分配共同地访问资源(130),所述方法包括:

在执行多个进程(111、112、113、114、121、122、123、124)中的至少一个进程期间确定(220、230)多个进程(111、112、113、114、121、122、123、124)中的至少一个进程的当前的实际处理时间,多个进程(111、112、113、114、121、122、123、124)中的至少一个进程在所述实际处理时间内被处理,

将所述实际处理时间与指派给多个进程(111、112、113、114、121、122、123、124)中的至少一个进程的额定处理时间和/或与所指派的额定处理时间和指派给多个进程(111、112、113、114、121、122、123、124)中的至少一个进程的处理时间延长的和进行比较(230),和

根据比较的结果调整(240、250)预定的资源分配。

2.按照权利要求1所述的方法,其中,实行(260):确定所述当前的实际处理时间,将所确定的实际处理时间与相应的所指派的额定处理时间和/或与相应的所指派的额定处理时间和相应的所指派的处理时间延长的和进行比较,并且在预定的监控间隔内周期性地调整资源分配。

3.按照权利要求1或2所述的方法,其中,根据所述多个进程(111、112、113、114、121、122、123、124)中的至少一个进程的实际处理时间是否偏离相应的所指派的额定处理时间偏离超过了相应的所指派的处理时间延长实行对所述资源分配的调整,和/或其中,根据所述多个进程(111、112、113、114、121、122、123、124)中的至少一个进程的实际处理时间偏离相应的额定处理时间到何种程度来实行对所述资源分配的调整。

4.按照前述权利要求中任一项所述的方法,其中,借助至少一个性能计数器(231、232、233)实行对所述多个进程(111、112、113、114、121、122、123、124)中的至少一个进程的当前的实际处理时间的确定。

5.按照前述权利要求中任一项所述的方法,其中,在资源分配的过程中,借助不同的分配参数调节(241、242、243、244)所述多个进程(111、112、113、114、121、122、123、124)对所述资源(130)的访问。

6.按照权利要求5所述的方法,其中,根据所述实际处理时间与所述额定处理时间和/或与所述额定处理时间和所述处理时间延长的和的比较的结果调整(240、250)所述资源分配包括了(241、242、243、244)对所述分配参数中的一个或多个分配参数的调整。

7.按照权利要求5或6所述的方法,还包括:确定在调整各个分配参数和由此产生的多个进程中的至少一个进程的实际处理时间的改变之间的关系,并且还包括:使用所确定的关系来根据所述实际处理时间与所述额定处理时间和/或与所述额定处理时间和所述处理时间延长的和的比较的结果调整所述一个或多个分配参数。

8.按照前述权利要求中任一项所述的方法,其中,以如下方式调整资源分配:

将资源的指派给所述多个进程中的第一进程的那一部分局部或完整地指派(241)给所述多个进程中的第二进程,

减少(242)预定的时间间隔内有待执行的业务的数量,和/或

提高或降低(242)所述多个进程中的特定的进程的执行优先权,和/或

将所述资源或所述资源的至少一部分独占地指派(243)给所述多个进程中的特定的进程,和/或

其中，对所述资源进行分区并且将分区中的其中一个分区独占地指派(244)给所述多个进程中的特定的进程。

9.按照前述权利要求中任一项所述的方法，其中，当所述多个进程(111、112、113、114、121、122、123、124)中的许多进程的实际处理时间分别偏离相应的所指派的额定处理时间偏离超过相应的所指派的处理时间延长时，所述方法包括：

基于所述进程(111、112、113、114、121、122、123、124)的运行时间优先权调整所述进程(111、112、113、114、121、122、123、124)的相应的所指派的处理时间延长，其中，尤其针对低优先权的进程将相应的所指派的处理时间延长提高了预定的值，和/或

根据所述进程(111、112、113、114、121、122、123、124)的运行时间优先权迭代地为所述进程(111、112、113、114、121、122、123、124)中的每个进程调整资源分配，特别是在最高优先权的进程开始。

10.按照前述权利要求中任一项所述的方法，其中，所述计算单元构造成车辆的控制设备(100)，并且其中，所述各个进程(111、112、113、114、121、122、123、124)设计用于控制车辆功能。

11.计算单元(100)、特别是车辆的控制设备，所述计算单元设置用于实行按照前述权利要求中任一项所述的方法的所有的方法步骤。

12.计算机程序，当在计算单元(100)上执行所述计算机程序时，所述计算机程序促使计算单元(100)实行按照权利要求1至10中任一项所述的方法的所有的方法步骤。

13.能机读的存储介质，带有按照权利要求12所述的储存于其上的计算机程序。

用于运行计算单元的方法

技术领域

[0001] 本发明涉及一种用于运行计算单元的方法以及一种计算单元和用于执行所述方法的计算机程序。

背景技术

[0002] 在例如涉及到自主驾驶或自动驾驶的车辆中的应用或进程可能极为计算密集和数据密集。相应地存在在网络物理系统中使用高性能计算机平台的倾向(例如通过到车辆外部的计算系统(所谓的“云”)中的交换)。这种平台可以用于执行在质量或服务质量(英文为“Quality of Service”,QoS)方面有不同的质量要求的不同的进程,例如伴随关于潜伏时间、流量或所谓的“Best Effort”(尽力而为)的个性化的要求。“服务质量”是衡量服务的质量与要求相符到何种程度的尺度。

[0003] 在这种平台上典型地存在诸如CPU集群、GPU或其它用户定义的硬件加速器(也称为存储器主机(Speichermaster)或主机(Master))的不同的单元,它们全部可以通过连接访问共享的资源,例如访问共享的存储器(Slave,即从机)。在此,在特定的核(尤其指的是CPU或CPU集群的多个处理器核中的一个处理器核)上执行的应用首先与在其它核上的应用竞争存储器。此外,诸如GPU之类的其它主机可以通过共享连接与CPU竞争共享的存储器。

[0004] 已经表明,由于对共享的资源、如主存储器的访问,当一个进程与其它进程一起在共同的多核平台上执行时,这个进程可能遭受执行时间的大幅延长。与其它主机的这种竞争再次使问题尖锐化。因此重要的是,在多核平台上共同执行不同的进程时能实现无干扰和时间解耦,特别是针对有实时要求的进程。

发明内容

[0005] 按照本发明,建议了带有独立权利要求特征的用于运行计算单元的方法以及计算单元和用于执行所述方法的计算机程序。有利的设计方案是从属权利要求的以及接下来的说明书的主题。

[0006] 在计算单元中执行多个进程或应用,它们共同按照预定的资源分配访问资源。本发明利用所述措施分别将有待遵守的运行时间或处理时间以及分别将允许的运行时间延长作为公差指派给各个进程。若无法遵守这些规定,例如因为在访问共享的资源时出现了冲突,那么就根据各个进程调整资源的分配。

[0007] 计算单元尤其具有带有多个处理器核的处理器单元,处理器核执行多个进程并且处理器核共同连接到相应的资源上,例如通过诸如总线之类的数据连接。这种处理器单元可以例如构造成CPU(“中央处理单元”)、GPU(“图形处理单元”)或例如也构造成DMA或DMA控制器(“直接存储器存取”)或构造成专用加速器。

[0008] 特别适宜的是,这些共享的资源可以是存储器单元,例如共同的工作存储器,这就是说特别是诸如DRAM(动态随机存取存储器)之类的易失的存储器。这种存储器单元具有带宽或存储器带宽,即每单位时间能读取和/或写入的数据的速率。这种带宽不一定是恒定不

变的。DRAM例如具有自己的内部的优化机制(参看例如缓存),因而视对存储器地址的访问顺序和数据的地点而定,这个带宽可以较高或较低。与此无关的是,处理器核并且因此所执行的进程必须分享可用的带宽。

[0009] 此外,共享的资源例如也可以是计算单元或处理器单元的计算容量,例如当各个进程不是固定地指派给特定的处理器核,而是可以由多个核执行时。共享的资源例如也可以是在计算单元中的模块或其它单元,例如外围设备。

[0010] 根据预定的资源分配,适宜地调节各个处理器核并且因此各个处理器对共享的资源的访问。特别适宜的是,通过这种资源分配在共享的存储器单元中将可用的存储器带宽分配给各个处理器核和进程。通过当前的方法提供了一种可能性,以便在计算单元运行期间动态地调整到共享的资源上的资源分配,以便能满足各个进程的要求。

[0011] 为了这个目的,在本发明的方法的范畴内分别为单独的或特别是所有的进程指派或预定额定处理时间或额定执行时间和允许的处理时间延长。在此应当在相应的所指派的额定处理时间内特别是完全处理或执行相应的进程。相应的进程的实际处理时间应当偏离相应的所指派的额定处理时间最多偏离了所指派的相应的处理时间延长。特别是当多个进程并行地访问共同的资源时,就出现了实际处理时间与额定处理时间的偏差。当实际处理时间超出额定处理时间在此超出得比相应的(允许的)处理时间延长还多时,可能妨碍相应的进程的或甚至整个计算单元的功能性。

[0012] 适宜的是,额定处理时间是相应的进程的运行时间的标准值或参考值,并且当相应的进程与竞争资源的另外的进程孤立或独立地执行并且因此可以独占地访问相应的资源时可以例如对应预期的运行时间。由额定处理时间和处理时间延长构成的和可以特别适宜地预定为实际处理时间的允许的阈值或允许的极限值。处理时间延长尤其可以对应允许的百分比或绝对时间,当由于对资源的共同访问出现了相应的延迟效应时,额定处理时间最多应当延长了所述允许的百分比或绝对时间。

[0013] 可以整体上针对整个进程或在间隔层级上观察和说明额定执行时间或处理时间。额定处理时间以及处理时间延长可以分别例如被说明为绝对的时间值或者适宜地也说明为相对值,由所述相对值可以推导出绝对的时间值。额定处理时间和处理时间延长可以例如分别预定为性能计数器、硬件性能计数器或硬件计数器(英文为“performance monitoring counter”,PMC,或者“performance monitoring unit”,PMU)的计数器值,其对计算单元中的特定的过程或活动进行测量或计数。工作时间改变还可以例如预定为与这些计数器值的按百分比计算的偏差。基于这种计数器值,通过目标硬件平台适宜地自行得出了相应的运行时间或相应的额定处理时间和处理时间延长。

[0014] 在计算单元中执行各个进程时,确定或求出各个进程的当前的实际处理时间,各个进程在当前实际上在所述实际处理时间内加以处理。将各个进程的这些实际处理时间与相应的所指派的额定处理时间相比较和/或与相应的所指派的额定处理时间和相应的所指派的处理时间延长的和相比较。根据这个比较的结果来调整资源分配。因此尤其监控,各个进程的实际上的实际处理时间在计算单元正常运行中是否偏离预定的额定处理时间,并且如果是的话,偏离到何种程度。若各个实际时间偏离相应的额定处理时间,这尤其表明,在访问共享的资源时发生了各个进程的冲突,由此使进程的处理时间延迟。若各个实际处理时间偏离所指派的额定处理时间甚至偏离了超过相应的允许的处理时间延长,那么所述系

统在允许范围外活动并且可能例如由于关于共同的资源的访问冲突而不利地损害相应的进程以及可能出现系统的不稳定。

[0015] 适宜的是，根据实际处理时间和额定处理时间是否偏离彼此并且偏离到何种程度，这样来调整资源分配，使得减小这些偏差并且相应的进程的处理时间接近它们的相应的额定值。在此尤其这样来调整资源分配，使得保证了超过了它们的相应的额定处理时间的关键的进程更多地访问资源，因而这些进程在不会在执行时受到负面的妨碍。它们的实际处理时间和额定处理时间一致或至少基本上一致的非关键的进程，则可以在过程中保证它们更少地访问所述资源。例如可以为此也将运行时间优先权值指派给每个进程，所述运行时间优先权值标明了进程的运行时间满足的重要性。

[0016] 本发明因此使得能在计算单元正在运行期间动态地和个性化地调节，哪个进程应当以哪种程度访问共享的资源，因而不会在执行进程时出现访问冲突、延迟和甚至出现错误或不稳定。资源到各个进程上的分配因此由计算单元独立地和自动地调节，个性化地与当前所执行的应用和进程相适应。

[0017] 适宜的是，对处理时间的监控和对资源分配的调整在软件中通过相应的软件单元实行或者通过相应的软件模块或软件机制实行。这种软件机制例如可以设置成执行管理器或资源管理器。这种软件机制或执行管理器可以尤其由计算单元的处理器单元执行。此外也可以想到的是，将执行这种软件机制的自身的硬件单元实现到计算单元中。

[0018] 所述方法此外还使得能将对尽可能最好的资源分配的确定在硬件侧和软件侧分离。尤其可以由程序员或软件开发人员为相应的进程预定额定处理时间和处理时间延长，例如在相应的进程的建立或编程阶段期间。尤其可以独立于特殊的计算单元或硬件配置地为相应的进程预定额定处理时间和处理时间延长，之后在所述特殊的计算单元或硬件配置上执行所述进程。因此可以在开发时给予各个软件进程的开发者高度的自由和灵活度。软件开发者的如何能尽可能好地执行他们的相应的进程的这些规定由计算单元在其运行期间自动地考虑到。额定处理时间为此目的例如可以由例如形式为计数器值的相对值等说明并且处理时间延长例如分别说明为这些计数器值的提高。

[0019] 在计算单元中执行的各个进程或应用尤其可以具有在质量或服务质量(英文为“Quality of Service”，QoS)方面的不同的质量要求。“QoS”或“Quality of Service”是衡量服务的质量与要求相符到何种程度的尺度。视相应的应用的类型和优先权而定，各个进程可以对它们的处理的延迟作出不同的反应。每个进程在其与其它应用同时执行时，尤其可以根据其时间上的关键程度一定程度上容忍由于争用共享的硬件资源引起的形式为执行时间延长的性能变差。延迟因此可以或多或少关键地影响各个进程和整个系统。对时间不关键的、不那么重要的进程，例如建立记录或日志文件，在处理延迟时例如不会产生或几乎不会产生负面的影响。时间关键的或要实时执行的进程，例如在控制电机、内燃机、逆变器等期间，则在它们的处理时间方面具有特别是仅小的余地并且仅能容忍它们的处理的有限的延迟。基于每个进程的关键性和敏感性，特别是基于执行时间的最大的增加(相应的进程可以容忍所述增加，以便正确地运行)适宜地为每个进程个性化地预定额定处理时间和允许的处理时间延长。基于这些规定在线监控和动态控制系统中的资源分配，从而将每个进程的特别是形式为执行时间的延长的访问冲突的影响限制在相应的预定的允许的值上。因此通过当前的方法可以特别适宜地确保了针对不同的进程和应用的服务质量(Quality

of Service)。

[0020] 共享的资源经常以传统的方式被静态地提供或分配并且至少在考虑到所谓的最坏情况竞争场景的情况下进行配置。这种静态的资源分配可能导致系统资源的无效率的使用。例如由Marco Caccamo、Rodolfo Pellizzoni、Lui Sha、Gang Yao和Heechul Yun在2013年的《MemGuard:用于多核平台中的高效性能隔离的存储器带宽预留系统》(2013年IEEE19届实时和嵌入式技术与应用研讨会(RTAS)(RTAS'13);IEEE计算机学会,美国华盛顿特区,55-64)中已知一种机制或一种方法,在所述机制或方法中,保障了用于CPU集群内不同的CPU核的最小的存储器带宽。这以如下方式运作,即,向核指派预算,即特定的值或使用整个可供使用的存储器带宽的份额,以及监控在调节间隔内最后的层级的缓存错误并且当超过了调节间隔内的预算时就使核停止运行。此外,这种方法还提供了这样一种可能性,即,基于之前的使用变化曲线动态地指派存储器带宽。但预测始终着眼于之前的间隔并且不是由事件控制的。假设将预算静态地指派给每个核,并且然后重新分配未使用的预算。

[0021] 存储器意识到的规划的另一种传统的可能性例如是,应当在特定的时间点上可以在不同的核上共同执行的任务,事先例如由规划者确定,因而不会扩展存储器极限。应用也可以安排在存储和计算阶段中并且可以不同的核上计划应用的不同的存储阶段。但值得期望的是这样一种方案,其独立于规划逻辑并且没有改变应用特性。

[0022] 嵌入式高性能平台通常在硬件侧例如越来越多地为在主机(例如CPU、GPU、DMA)和主存储器之间的连接提供QoS模块。这些QoS模块在基于诸如未结交易之类的不同的参数调节数据通信量时、调节速率时或基于潜伏调节时有用。例如可以这样配置在连接上的寄存器,即,确定针对未结业务、例如针对GPU的限制,因此所述连接不会被存储器业务淹没。一旦超过所述限制,连接就会对来自GPU的正在进入的数据通信量节流,因而其它主机可以访问存储器。这些模块典型地能通过寄存器编程并且控制连接到所述连接上的主部件。

[0023] 但这些QoS模块通常将核的总集群大多处理成唯一一个主机,因为总集群在唯一一个端口或唯一一个接口处与所述连接连接。因此QoS模块必要时仅提供了在集群的层级上的调节并且可能无法区分不同的核,这并没有解决跨核竞争的问题。此外,调节参数的静态配置可能不足以有效使用共同的存储器。

[0024] 此外,在CPU集权内的不同的核的QoS调节通常没有得到下方的硬件的支持。这意味着,在互连层级上的硬件-QoS支持仅能保护核不受其它干扰源的影响,但无法保护核不受在同一集群内其它核的影响。因此顶层软件机制通常结合互连层级上的QoS-支持提供必要的干扰调节。

[0025] 但在这种基于软件的调节机制中经常仅将静态的资源预算指派给每个核。资源预算在此经常设定到特定的调节间隔内最大数量的资源访问,这可能导致非实时核极大地变慢。此外,可能被证实困难的是,配置这种机制。这需要有关底层硬件和特殊的工作负载的专业知识。例如在由虚拟机监视程序(Hypervisor)、操作系统、中间件和额外的虚拟部构成的大型软件堆栈中,如Docker Container,这可以被证明极为昂贵。错误配置的系统可能导致巨大的性能损失或者无法减少干扰效应。

[0026] 与这些办法相反的是,通过当前的方法建议了一种软件机制,其自动考虑到了进程的变化的资源使用模式,如变化的存储器使用模式,并且可以在QoS模块中重新配置这些调节参数。所述方法提出了一种可能性,即,还在连接层级上动态地调节诸如用于不同的主

机的存储器带宽之类的资源(在此既可以涉及到单独的处理器单元也可以涉及到处理器单元(或核)的组群)并且额外在处理器单元组群内,即在主机内调节在核之间的资源。

[0027] 有利地实行:确定当前的实际处理时间;将所确定的实际处理时间与相应的所指派的额定处理时间或与相应的所指派的额定处理时间和相应的所指派的处理时间延长的和相比较;和在预定的监控间隔内周期性地调整资源分配。因此可以在当前的监控事件下确定当前的实际处理时间与相应的额定值的偏差并且可以基于所述偏差调整资源分配。在接下来的监控事件中可以观察这种调整对处理时间的影响。

[0028] 优选根据至少一个行程的实际处理时间是否偏离相应的所指派的额定处理时间偏离了多于相应的所指派的处理时间延长来调整资源分配。适宜的是,这样来动态地调整资源分配,使得尽可能不超过允许的处理时间延长并且因此不会发生系统的不稳定。

[0029] 备选或附加的是,优选根据各个进程的实际处理时间偏离相应的额定处理时间到何种程度来调整资源分配。在这种情况下尤其动态地调节资源分配,从而可以找到资源的最优的分配并且针对各个进程分别达到尽量接近预定的额定值的尽可能最好的处理时间。

[0030] 各个进程的当前的实际处理时间有利地借助至少一个性能计数器确定。性能计数器或硬件性能计数器或硬件计数器(英文为“performance monitoring counter”,PMC,或者“performance monitoring unit”,PMU)是处理器单元或微处理器的例如形式为特殊的寄存器的部件,它们对计算单元内的特定的过程或活动进行测量或计数。因此适宜的是,收集对应的性能计数器的数据,以便监控进程的执行时间的增加(进程在其执行期间遭受所述执行时间的增加)。在此尤其考虑到了由多个性能计数器构成的组合的数据。尤其周期性地在预定的监控间隔内执行通过性能计数器的监控。适宜的是,软件机制或执行管理器可以从这些性能计数器读取信息并且将其与所储存的预定的额定处理时间和处理时间延长相比较。

[0031] 按照一种优选的实施方式,在资源分配的过程中,借助不同的分配参数来调节或组织或安排多个进程对共享的资源的访问。尤其可以设置不同的分配机制或控制或组织机制,其按照这些分配参数分别以不用的方式将部分资源指派给各个处理器核或进程。特别适宜的是,为此目的设置多个QoS模块或QoS机制,其借助相应的分配参数来调节相应的分配机制。

[0032] 优选根据实际处理时间与额定处理时间或者与相应的额定处理时间和处理时间延长的和的比较来调整分配参数中的一个或多个分配参数。不同的分配参数提供了灵活的可能性来调整资源的分配。尤其可以根据对各个进程的预定,即根据额定处理时间和处理时间延长来确定分配参数的额定值。因此适宜的是,将额定处理时间和处理时间延长转化成特定于QoS的额定值或用于不同的硬件和软件QoS机制240的调谐参数。软件机制或执行管理器尤其可以为了确定和比较实际处理时间而基于当前的比较结果指示各个QoS模块或QoS机制调整相应的分配参数。适宜的是,软件机制将特定于QoS的额定值传达给各个QoS模块或QoS机制,其与相应的分配参数相应地协调。

[0033] 按照一种优选的实施方式,确定或学习了在调整各个分配参数和由此得出的各个进程的实际处理时间改变之间的关系或相互关系或关联。这些所确定的和所学习的关系用于,根据实际处理时间与额定处理时间或由额定处理时间和处理时间延长构成的和的比较的结果来调整所述分配参数中的一个或多个分配参数。软件机制尤其可以在机械学习的过

程中或借助用于机器学习的算法来学习在分配参数和处理时间之间的相应的关联。这些关系尤其可以在第一时间点上确定，例如在学习阶段期间，并且在之后的第二时间点上使用，特别是在计算单元正常运行期间。

[0034] 有利地以如下方式来调整资源分配，即，将资源的指派给第一进程的那一部分局部或完整地指派给第二进程。因此资源可能局部再分配，例如从实际处理时间和额定处理时间彼此接近或例如不是时间关键的进程的第一进程再分配到其处理不应被延迟的第二进程。

[0035] 备选或附加地可以以如下方式调整资源分配，即，优选在预定的时间间隔内、特别是在下一个监控间隔内减少有待执行的业务或操作的数量。例如可以推迟非时间关键的业务，以便保证时间关键的进程对资源的访问。

[0036] 此外，可以备选或附加地以如下方式调整资源分配，即，优选提高特定的进程的、特别是其实际处理时间偏离相应的所指派的额定处理时间偏离超过相应的所指派的处理时间延长的进程的执行优先权。适宜的是，保证这种更高优先权的进程比有较小执行优先权的进程能更快地访问资源。

[0037] 备选或附加的是，可以优选以如下方式调整资源分配，即，将资源或资源的至少一部分独占地指派给特定的进程。因此例如可以保证由于访问冲突而被延迟的时间关键的进程能独占地访问资源或所述资源的一部分。

[0038] 备选或附加地以如下方式有利地调整资源分配，即，对资源进行分区并且将这些分区中的其中一个分区独占地指派给特定的进程。例如可以在进程的过程中将共享的存储器划分成多个单独的存储区域，它们能灵活地指派给各个进程。

[0039] 当多个进程的实际处理时间分别偏离相应的所指派的额定的处理时间偏离超过相应的所指派的处理时间延长时，存在的风险是，由于访问冲突而在所有这些进程中都可能出现问题或不稳定。在这种情况下特别重要的是，对调整资源分配时以何种顺序考虑这些进程区分优先次序。在这种情况下，优选基于这些进程的运行时间优先权来调整这些进程的相应的所指派的处理时间延长。特别适宜的是，在此针对低优先权的进程将相应的所指派的处理时间延长提高了一个预定的值。因此可以尤其达到的是，在资源分配时优先处理较高优先权的时间关键的进程。备选或附加的是，优选根据这些进程的运行时间优先权迭代地或连续地或前后相继地为这些进程中的每个进程分别调整资源分配，特别是在最高优先权的进程开始。因此适宜的是，先为最高优先权的进程调整资源分配，并且紧接着迭代地相应地处理其余的进程。

[0040] 本发明适用于许多不同的应用领域，本发明特别有利地适用于(机动)车辆领域，其中，计算单元有利地构造成(机动)车辆的控制设备，并且其中，各个进程设计用于控制车辆功能。由计算单元执行的进程可以例如包括安全关键的功能，其设计用于例如在马达控制过程中或驾驶辅助功能等过程中安全地运行车辆和控制车辆。通过所述方法可以特别适宜地使这种安全关键的和特别是时间关键的、有待实时地执行的进程能满足它们的相应的实时条件并且不会由于到控制设备的共享的资源、特别是到共享的存储器的访问冲突而被延迟。控制设备的安全性或故障安全性和完善性可以通过当前的方法提高。尤其可以通过所述方法满足(机动)车辆领域中的安全要求，如在ISO标准26262中或特别是由所谓的“汽车安全完整性等级”(ASIL)预定的针对机动车中的安全重要的系统的由ISO 26262详述的

安全要求等级。

[0041] 按本发明的计算单元，例如机动车的控制设备，尤其在程序技术上设置用于实行按本发明的方法。

[0042] 以用于实行所有的方法步骤的计算机程序或带有程序代码的计算机程序产品的形式实现按本发明的方法也是有利的，因为这引起了特别低的成本，特别是当正在执行的控制设备还能用于另外的任务并且因此总是存在时。用于提供计算机程序的合适的数据载体尤其是磁性的、光学的和电气的存储器，如硬盘、闪存、EEPROM、DVD等。也能通过计算机网络(因特网、局域网等)下载程序。

[0043] 本发明的进一步的优点和设计方案有说明书和附图得出。

[0044] 本发明借助实施例在附图中示意性示出并且下文中参考附图加以说明。

附图说明

[0045] 图1示意性示出了按本发明的计算单元的一种优选的设计方案，所述计算单元设置用于实行按本发明的方法的一种优选的实施方式；

图2作为方块图示意性示出了按本发明的方法的一种优选的实施方式。

具体实施方式

[0046] 在图1中示意性地示出了在此例如车辆的一种计算单元，如控制设备，并且用附图标记100标注。

[0047] 控制设备100具有带有两个处理器核110和120的处理器单元。这些处理器核110、120中的各一个处理器核执行多个进程111、112、113、114和121、122、123、124。当然，处理器单元也可以具有更多数量的处理器核并且也还可以在各个处理器核上分别执行更多数量的进程。

[0048] 控制设备100可以例如是用于控制车辆的电机的马达控制设备。各个进程111、112、113、114、121、122、123、124可以包括用于安全地运行车辆和用于控制机械的安全关键的功能。

[0049] 进程111、112、113、114、121、122、123、124可以按照预定的资源分配共同地访问资源130，例如访问共享的存储器单元，例如构造成DRAM的工作存储器。为了这个目的，处理器核110、120通过数据连接、例如数据总线共同地连接到存储器130上。

[0050] 通过共同访问存储器可能发生执行各个进程中的延迟。为了对抗这种延迟，可以在控制设备运行期间动态地调整资源分配。为了这个目的，控制设备特别是在程序技术上设置用于实行如在图2中作为方块图示意性示出并且接下来应当阐释那样的按本发明的方法的一种优选的实施方式。

[0051] 在(准备性的)步骤210中，分别将额定处理时间和处理时间延长指派给进程111、112、113、114、121、122、123、124。在这些额定处理时间内应当分别处理各个进程111、112、113、114、121、122、123、124。各个进程111、112、113、114、121、122、123、124的实际处理时间应当偏离相应的所指派的额定处理时间最多偏离了相应的所指派的处理时间延长。

[0052] 这些额定处理时间和处理时间延长尤其可以由程序员或软件开发人员例如在相应的进程111、112、113、114、121、122、123、124的建立或编程阶段期间确定。当伴随对存储

器的独占访问在计算单元中孤立地执行相应的进程时，额定处理时间可以例如对应预期的运行时间。额定处理时间和处理时间延长的和可以预定为允许的阈值或极限值，处理时间最多应当延长了所述允许的阈值或极限值，因此不会由于对存储器的共同访问而发生不稳定。额定处理时间和处理时间延长尤其基于关键性或优先性预定，此外还尤其基于相应的进程的实时要求预定。

[0053] 额定处理时间以及处理时间延长可以分别例如说明为绝对的时间值或也说明为相对值。额定处理时间例如可以分别预定为性能计数器的、硬件性能计数器的或硬件计数器的计数器值并且处理时间延长可以分别预定为这些计数器值的按百分比计算的偏差。由这些计数器值然后通过目标硬件平台得出了相应的运行时间或针对额定处理时间和针对处理时间延长的具体的时间值。

[0054] 额定处理时间和处理时间延长储存在控制设备100中，特别是执行管理器220中，所述执行管理器设计成控制设备100中的软件机制或软件单元或软件模块或者例如也可以设置成单独的硬件模块。在控制设备100中执行各个进程111、112、113、114、121、122、123、124期间，执行管理器220确定了当前的实际处理时间，在所述当前的实际处理时间内处理各个进程111、112、113、114、121、122、123、124。

[0055] 这些当前的实际处理时间例如借助至少一个性能计数器确定。控制设备100尤其具有带有多个性能计数器、硬件性能计数器或硬件计数器(英文为“performance monitoring counter”，PMC，或者“performance monitoring unit”，PMU)231、232、233的监控单元230，分别例如以特殊的寄存器的形式，它们对控制设备100内的特定的过程或活动进行测量或计数。执行管理器220从性能计数器231、232、233读取信息并且基于此确定了各个进程111、112、113、114、121、122、123、124的当前的实际处理时间。

[0056] 例如可以借助性能计数器231监控进程的执行周期。例如可以在ARM处理器架构中由性能计数器232测量事件0xE7/0xE8，以便基于写入或读取操作来计算存储器停顿。此外，可以借助性能计数器233例如监控Level-2缓存丢失(Level-2-Cache-Misses)和L2缓存回写(L2-Cache-Write-Backs)，以便确定由于存储器访问的延迟。

[0057] 执行管理器220将各个进程111、112、113、114、121、122、123、124的特定的实际处理时间与相应的所指派的额定处理时间相比较并且此外还与相应的额定处理时间和相应的处理时间延长的和相比较。根据这个比较的结果调整资源分配。实际时间与所指派的额定时间的偏差尤其意味着在存储器访问时的冲突和由此引起的处理时间延迟。根据这些偏差来动态地调整资源分配。

[0058] 在资源分配的进程中，各个进程111、112、113、114、121、122、123、124对共同的存储器130的访问尤其借助不同的分配参数加以调节。为了这个目的，尤其设置多个分配、控制或组织机制240，它们根据这些分配参数分别以不同的方式将资源130的一些部分指派给各个进程。特别适宜的是，为了这个目的，设置多个QoS模块、QoS机制或QoS控制器241、242、243、244，它们借助相应的分配参数调节相应的分配机制。

[0059] 根据比较的结果调整分配参数中的一个或多个分配参数。执行管理器220尤其为了这个目的将额定处理时间和处理时间延长转化成特定于QoS的额定值或用于不同的硬件和软件QoS机制240的调谐参数。若各个实际处理时间偏离相应的额定处理时间，那么执行管理器220指示相应的QoS控制器240调整相应的分配参数。

[0060] 在步骤250中，相应地调整资源分配并且所述进程根据这种经调整的资源分配访问共同的存储器130。于是通过附图标记260表明，当前的实际处理时间重新由性能计数器230确定并且由执行管理器220将其与额定时间或允许的延长相比较。因此尤其可以实现调节回路并且可以适宜地动态地调节或调控资源分配。执行管理器220尤其可以在其过程中周期性地在预定的监控间隔内确定实际处理时间，与预定的值相比较并且调整资源分配。此外，执行管理器220能以这种方式例如在机器学习的过程中学习在调整各个分配参数和由此产生的各个进程的实际处理时间的改变之间的关系、相互关系或关联。这种学习到的关系在接下来的监控间隔中能够用于调整资源分配。因此始终能改进资源分配的调整。

[0061] 每个QoS控制器主管特定的QoS机制并且相应地协调QoS参数。QoS控制器241例如可以设置用于调节存储器带宽。当例如由执行管理器220确认，时间关键的进程111的实际处理时间偏离所指派的额定处理时间偏离超过预定的处理时间延长时，执行管理器220指示QoS控制器241在下一个监控间隔的持续时间内减少指派给其它进程112、113、114、121、122、123、124的存储器带宽并且取而代之地将其指派给所述的时间关键的进程111。

[0062] 此外，QoS管理器242可以例如设置用于控制互连。当执行管理器220例如确认，在一个核上，例如在核110上的进程超出了允许的处理时间延长时，执行管理器220能够例如指示QoS控制器242减少在下一个监控间隔的持续时间内其它核120的未结业务的数量或者提高核110上的业务的优先权，因此更为快速地处理这些业务。

[0063] QoS控制设器243例如可以主管缓存层次结构(Cache-Hierarchie)。当执行管理器220例如确认，时间关键的进程111的处理时间超过了允许的处理时间延长时，执行管理器220能够指示QoS控制器243例如借助诸如缓存着色之类的软件机制或由ARM DynamiQ-CPU架构提供的硬件机制对缓存分区，因而缓存的一些部分被独占地为进程111预留，这减少了干扰效应。

[0064] 因为可能无法同时满足所有状况中的所有应用的要求(例如由于瞬间提高的竞争场景)，所以执行管理器220在需要时决定哪个进程优先。这可以例如通过例如基于关键性和安全要求通过规定预定义的递降顺序发生，以便确保较高优先权的进程的性能。当执行管理器220确认，多个进程的实际处理时间分别偏离所指派的额定处理时间偏离超过所指派的处理时间延长时，执行管理器220可以适宜地提高有较低优先权的进程的所指派的处理时间延长并且然后执行调节回路，以便满足这些经改变的值。备选或附加的是，执行管理器220可以适宜地根据这些进程的运行时间优先权迭代地针对这些进程中的每个进程分别在调节回路的过程中调整资源分配，在最为优先的进程处开始。

[0065] 因此在当前的方法的范畴内可以在软件/硬件中动态地调节不同的QoS参数，其通过由一个或多个性能计数器(PMC)获得的信息加以控制，以便满足应用的由用户确定的QoS要求。例如可以使用有关L2缓存丢失(L2-Cache-Misses)和回写(Write-Backs)的信息或有关由于资源访问的闭锁周期的信息，以便用相应的软件机制或其它QoS机制在互连上为不同的进程或处理器核动态地配置共享的资源的预算。此外，例如也可以使用性能计数器的、例如DDR性能计数器或互连上的性能计数器的测量，以便动态地配置不同的软件/硬件-QoS机制。可以例如附加地使用操作系统内核(OS-Kernel)事件来控制QoS机制的配置。因此可以通过当前的方法特别是提高平均的系统性能并且可以确保对应用的性能保障。

[0066] 借助当前的方法可以特别适宜地控制主存储器干涉(例如用最后的监控周期的缓

存丢失测得)并且此外还控制缓存-干涉,例如在使用性能计数器的附加的信息的情况下。

[0067] 执行管理器按照当前的方法特别适宜地自我调节,特别是基于诸如关键性、期限和安全边界之类的指定的信息。用户尤其仅指定QoS要求,而不知道硬件/软件堆栈的细节。所述机制因此对用户而言是透明的并且从使用者角度来看使配置和使用变得不那么复杂。

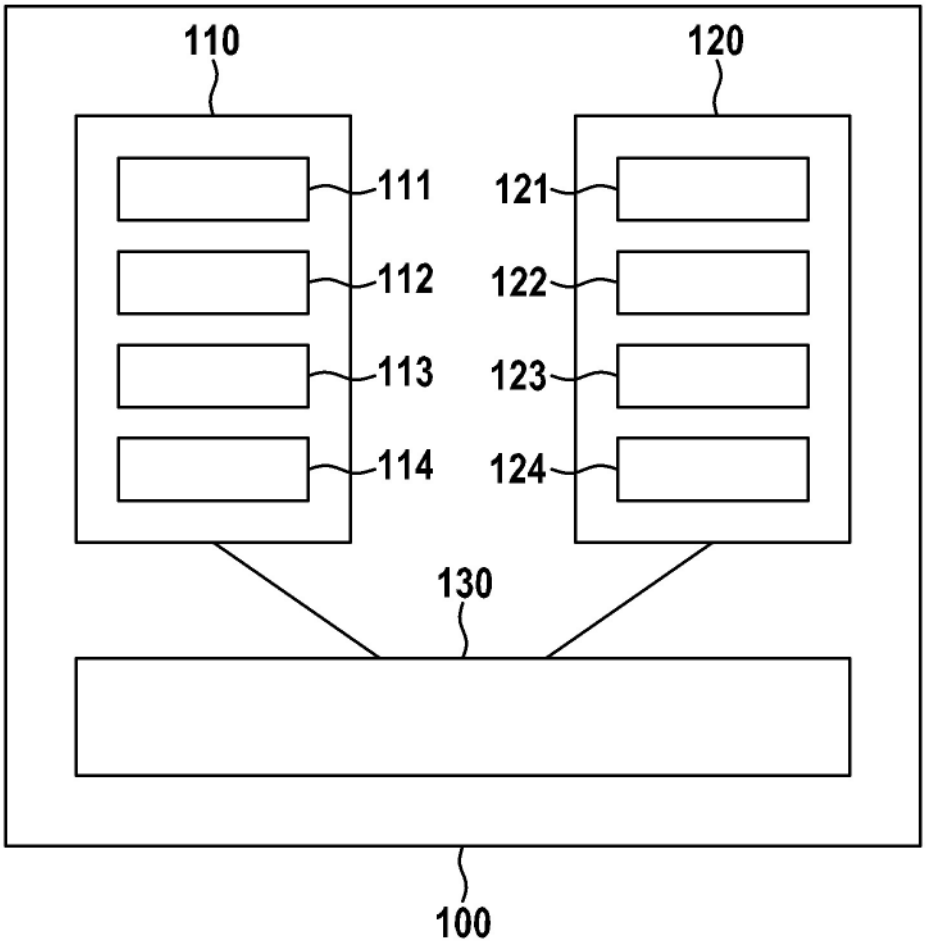


图 1

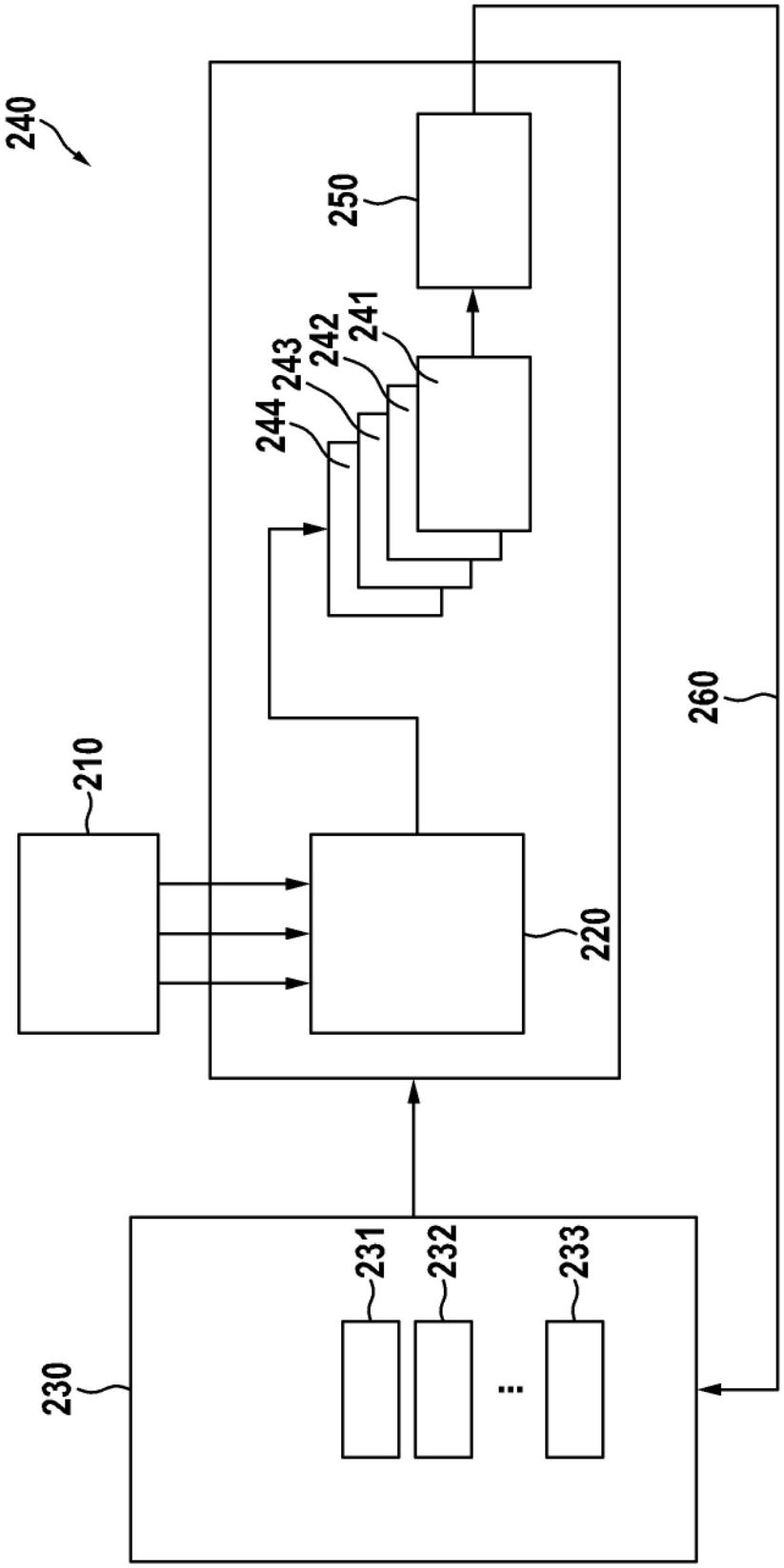


图 2