



(12) 发明专利申请

(10) 申请公布号 CN 115827419 A

(43) 申请公布日 2023. 03. 21

(21) 申请号 202111360721.X

(22) 申请日 2021.11.17

(66) 本国优先权数据

202111085342.4 2021.09.16 CN

(71) 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72) 发明人 周维 刁阳彬 肖杰

(74) 专利代理机构 北京亿腾知识产权代理事务所(普通合伙) 11309

专利代理师 刘辰雷 陈霖

(51) Int.Cl.

G06F 11/36 (2006.01)

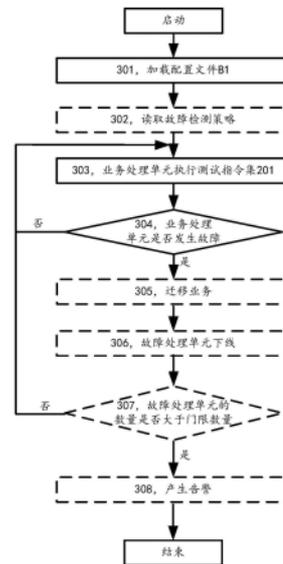
权利要求书3页 说明书18页 附图9页

(54) 发明名称

故障检测方法、处理单元集群及计算设备

(57) 摘要

处理单元集群,包括多个处理器核心,该多个处理器核心包括管理处理器核心和第一业务处理器核心,其中:第一业务处理器核心用于执行第一测试指令集,得到第一实际执行结果;管理处理器核心用于比较第一实际执行结果和第一测试指令集的第一预期执行结果;当第一实际执行结果和第一预期执行结果不一致时,管理处理器核心将第一业务处理器核心标记为故障处理单元。该方案可以提高故障检测效率,降低故障检测开销。



1. 一种处理单元集群,其特征在于,包括多个处理单元,所述多个处理单元包括管理处理单元和第一业务处理单元,其中:

所述第一业务处理单元用于执行第一测试指令集,得到第一实际执行结果;

所述管理处理单元用于比较所述第一实际执行结果和所述第一测试指令集的第一预期执行结果;

当所述第一实际执行结果和所述第一预期执行结果不一致时,所述管理处理单元用于将所述第一业务处理单元标记为故障处理单元。

2. 根据权利要求1所述的集群,其特征在于,所述业务处理单元为以下任一种:

多核处理器中的计算核、CPU、GPU、专用集成电路ASIC、神经网络处理单元NPU。

3. 根据权利要求1或2所述的集群,其特征在于,所述第一业务处理单元属于第一处理器架构,所述第一测试指令集为适配于所述第一处理器架构的传统测试指令集中的部分指令,所述传统测试指令集还包括其他指令集。

4. 根据权利要求1-3任一项所述的集群,其特征在于,

当所述第一实际执行结果和所述第一预期执行结果一致时,所述第一业务处理单元还用于执行第二测试指令集,得到第二实际执行结果,所述第二测试指令集和所述第一测试指令集为适配于第一处理器架构的传统测试指令集中的不同指令,所述第一业务处理器属于所述第一处理器架构;

当所述第二实际执行结果和所述第二测试指令集的第二预期执行结果不一致时,所述管理处理单元用于将所述第一业务处理单元标记为故障处理单元。

5. 根据权利要求4所述的集群,其特征在于,所述第一测试指令集用于测试所述第一业务处理单元中的第一运算单元集合,所述第二测试指令集用于测试所述第一业务处理单元中的第二运算单元集合,其中,所述第一运算单元集合中的运算单元发生故障的可能性高于所述第二运算单元集合中的运算单元发生故障的可能性。

6. 根据权利要求1-3任一项所述的集群,其特征在于,所述多个处理单元还包括第二业务处理单元;当所述第一实际执行结果和所述第一预期执行结果不一致时,所述第一业务处理单元将当前业务迁移到所述第二业务处理单元。

7. 根据权利要求1-3任一项所述的集群,其特征在于,当所述第一实际执行结果和所述第一预期执行结果不一致时,所述第一业务处理单元下线。

8. 根据权利要求1-3任一项所述的集群,其特征在于,所述管理处理单元还用于:当所述多个处理单元中故障处理单元的数量大于第一阈值时,产生告警;当所述数量不大于所述第一阈值时,不产生告警。

9. 一种计算设备,其特征在于,包括:

用于存储第一测试指令集的存储器;

权利要求1-8任一项所述的处理单元集群。

10. 一种故障检测方法,其特征在于,处理单元集群包括多个处理单元,所述多个处理单元包括管理处理单元和第一业务处理单元,所述方法包括:

所述第一业务处理单元执行第一测试指令集,得到第一实际执行结果;

所述管理处理单元比较所述第一实际执行结果和所述第一测试指令集的第一预期执行结果;

当所述第一实际执行结果和所述第一预期执行结果不一致时,所述管理处理单元将所述第一业务处理单元标记为故障处理单元。

11. 根据权利要求10所述的方法,其特征在于,所述多个处理单元还包括第二业务处理单元,所述方法还包括:

当所述第一实际执行结果和所述第一预期执行结果不一致时,所述第一业务处理单元将当前业务迁移到所述第二业务处理单元。

12. 根据权利要求10所述的方法,其特征在于,所述方法还包括:

当所述第一实际执行结果和所述第一预期执行结果不一致时,所述第一业务处理单元下线。

13. 根据权利要求10所述的方法,其特征在于,所述方法还包括:所述管理处理单元在所述多个处理单元中故障处理单元的数量大于第一阈值时,产生告警;在所述数量不大于所述第一阈值时,不产生告警。

14. 一种处理单元集群,其特征在于,包括多个处理单元,所述多个处理单元包括管理处理单元和多个业务处理单元;

所述多个业务处理单元分别用于执行第一测试指令集,得到多个实际执行结果,所述多个实际执行结果和所述多个业务处理单元一一对应;

所述管理处理单元用于比较所述多个实际执行结果;

当所述多个实际执行结果不完全一致时,所述管理处理单元用于指示所述多个业务处理单元中存在故障处理单元。

15. 根据权利要求14所述的集群,其特征在于,所述业务处理单元包括以下任一种:

多核处理器中的计算核、CPU、GPU、专用集成电路ASIC、神经网络处理单元NPU。

16. 根据权利要求14或15所述的集群,其特征在于,所述业务处理单元属于第一处理器架构,所述第一测试指令集为适配于所述第一处理器架构的传统测试指令集中的部分指令,所述传统测试指令集还包括其他指令集。

17. 根据权利要求14-16任一项所述的集群,其特征在于,所述多个业务处理单元包括第一业务处理单元,所述第一业务处理单元对应所述多个实际执行结果中的第一实际执行结果;

所述管理处理单元用于:在所述多个实际执行结果中,当与所述第一实际执行结果相同的实际执行结果占少数时,将所述第一业务处理单元标记为故障处理单元。

18. 根据权利要求14-16任一项所述的集群,其特征在于,

当所述多个实际执行结果完全一致时,所述多个业务处理单元还用于分别执行第二测试指令集,得到多个第二实际执行结果,所述第二测试指令集和所述第一测试指令集为适配于第一处理器架构的传统测试指令集中的不同指令,所述多个业务处理器属于所述第一处理器架构,所述多个第二实际执行结果和所述多个业务处理单元一一对应;

当所述多个第二实际执行结果不完全一致时,所述管理处理单元用于指示所述多个业务处理单元中存在故障处理单元。

19. 根据权利要求18所述的集群,其特征在于,所述第一测试指令集用于测试所述业务处理单元中的第一运算单元集合,所述第二测试指令集用于测试所述业务处理单元中的第二运算单元集合,其中,所述第一运算单元集合中的运算单元发生故障的可能性高于所述

第二运算单元集合中的运算单元发生故障的可能性。

20. 根据权利要求14-19任一项所述的集群,其特征在於,所述多个业务处理单元包括第一业务处理单元和第二业务处理单元;当所述第一业务处理单元为故障处理单元,且所述第二业务处理单元不是故障处理单元时,所述第一业务处理单元将当前业务迁移到所述第二业务处理单元。

21. 根据权利要求14-19任一项所述的集群,其特征在於,所述故障处理单元下线。

22. 根据权利要求14-19任一项所述的集群,其特征在於,所述管理处理单元还用于:当所述多个业务处理单元中故障处理单元的数量大于第一阈值时,产生告警;当所述数量不大于所述第一阈值时,不产生告警。

23. 一种计算设备,其特征在於,包括:

用于存储第一测试指令集的存储器;

权利要求14-22任一项所述的处理单元集群。

24. 一种故障检测方法,其特征在於,处理单元集群包括多个处理单元,所述多个处理单元包括管理处理单元和多个业务处理单元,所述方法包括:

所述多个业务处理单元分别执行第一测试指令集,得到多个实际执行结果,所述多个实际执行结果和所述多个业务处理单元一一对应;

所述管理处理单元比较所述多个实际执行结果;

当所述多个实际执行结果不完全一致时,所述管理处理单元指示所述多个业务处理单元中存在故障处理单元。

25. 根据权利要求24所述的方法,其特征在於,所述多个业务处理单元包括第一业务处理单元和第二业务处理单元,所述方法还包括:

当所述第一业务处理单元为故障处理单元,且所述第二业务处理单元不是故障处理单元时,所述第一业务处理单元将当前业务迁移到所述第二业务处理单元。

26. 根据权利要求24所述的方法,其特征在於,所述方法还包括:所述故障处理单元下线。

27. 根据权利要求24所述的方法,其特征在於,所述方法还包括:所述管理处理单元在所述多个业务处理单元中故障处理单元的数量大于第一阈值时,产生告警;在所述数量不大于所述第一阈值时,不产生告警。

## 故障检测方法、处理单元集群及计算设备

### 技术领域

[0001] 本申请涉及处理器故障检测技术,具体涉及一种故障检测方法、处理单元集群及计算设备。

### 背景技术

[0002] 随着半导体技术的发展,片上系统(system on chip,SOC)制程工艺越来越先进,一枚处理器集成的计算核(core)越来越多,处理器的集成度也越来越高。相应的,SOC的特征尺寸也越来越小。对于作为SOC主流工艺的互补型金属氧化物半导体(complementary metal-oxide semiconductor,CMOS)而言,越来越小的特征尺寸越来越接近CMOS的缩放极限,导致计算核越来越容易发生故障。

[0003] 目前,计算核的故障已成为处理器最常见故障,因此,研究计算核故障的快速检测方案越发迫切。

### 发明内容

[0004] 提供了一种故障检测方法、处理单元集群及计算设备,可以提高故障检测效率,降低故障检测开销。

[0005] 第一方面,提供了一种处理单元集群,包括多个处理单元,该多个处理单元包括管理处理单元和第一业务处理单元,其中:第一业务处理单元用于执行第一测试指令集,得到第一实际执行结果;管理处理单元用于比较第一实际执行结果和第一测试指令集的第一预期执行结果;当第一实际执行结果和第一预期执行结果不一致时,管理处理单元用于将第一业务处理单元标记为故障处理单元。

[0006] 提供的处理单元集群在进行故障检测时,可以将测试指令集的预期执行结果和处理单元执行测试指令的实际执行结果进行比较,以处理单元故障检测,从而可以发现故障处理单元。与处理单元重复执行同一测试指令集,对先后获得的测试结果进行比较的方案相比,本方案提高了故障检测效率,且降低了故障检测开销。

[0007] 在一种可能的实现方式中,业务处理单元为多核处理器中的计算核、CPU、GPU、专用集成电路ASIC、神经网络处理单元NPU中的任一种。

[0008] 也就是说,在该实现方式中,提供的故障检测方案可以实现对多种类型的处理单元的故障检测,具有广泛的应用性。

[0009] 在一种可能的实现方式中,第一业务处理单元属于第一处理器架构,第一测试指令集为适配于第一处理器架构的传统测试指令集中的部分指令,所述传统测试指令集还包括其他指令集。

[0010] 也就是说,在对业务处理单元进行故障检测时,业务处理单元可以执行其所属处理器架构对应的测试指令集中的部分指令,从而可以快速检测出故障处理单元,进一步提高了故障检测效率。

[0011] 在一种可能的实现方式中,当第一实际执行结果和第一预期执行结果一致时,第

一业务处理单元还用于执行第二测试指令集,得到第二实际执行结果,第二测试指令集和第一测试指令集为适配于第一处理器架构的传统测试指令集中的不同指令,第一业务处理器属于第一处理器架构;当第二实际执行结果和第二测试指令集的第二预期执行结果不一致时,管理处理单元用于将第一业务处理单元标记为故障处理单元。

[0012] 也就是说,在该实现方式中,业务处理单元可以分批执行测试指令集,每当执行一批测试指令集时,可以判断业务处理单元是否为故障处理单元,从而可以及早检测出故障处理单元,提高了故障检测效率。

[0013] 在一种可能的实现方式中,第一测试指令集用于测试第一业务处理单元中的第一运算单元集合,第二测试指令集用于测试第一业务处理单元中的第二运算单元集合,其中,第一运算单元集合中的运算单元发生故障的可能性高于第二运算单元集合中的运算单元发生故障的可能性。

[0014] 也就是说,把传统的测试指令集分为多个子测试指令集,不同的子测试指令集所测试的对象不同或者说所主要针对的运算单元不同。业务处理单元可以先执行其中一个子测试指令集,并且,用于执行的子测试指令集所针对的测试对象,是容易发生故障的运算单元。因此可以在执行少量测试指令的情况下,便有较大几率检测到故障运算单元,进而可以及早将故障运算单元所在的处理单元标记为故障处理单元。当第一个子测试指令集测试完成后没有检测到运算单元故障,那么可以针对次容易发生故障的运算单元,进行下一个子测试指令集的测试。以此类推,直至检测到故障,或者所有子测试指令集测试完成。

[0015] 在一种可能的实现方式中,该多个处理单元还包括第二业务处理单元;当第一实际执行结果和第一预期执行结果不一致时,第一业务处理单元将当前业务迁移到第二业务处理单元。

[0016] 也就是说,在该实现方式中,故障处理单元可以将其正在执行的业务迁移到其他处理单元,从而提高了处理单元集群的容错性和可用性,改善了用户体验。

[0017] 在一种可能的实现方式中,当第一实际执行结果和第一预期执行结果不一致时,第一业务处理单元下线。

[0018] 也就是说,在该实现方式中,故障处理单元可以下线,以避免执行业务,从而提高了处理单元集群的容错性和可用性,改善了用户体验。

[0019] 在一种可能的实现方式中,管理处理单元用于:当该多个处理单元中故障处理单元的数量大于第一阈值时,产生告警;当该数量不大于第一阈值时,不产生告警。

[0020] 也就是说,在该实现方式中,当故障处理单元的数量较多时,可以及时提醒用户,以便用户采取相应措施,从而改善了用户体验。

[0021] 在一种可能的实现方式中,管理处理单元用于当第三实际执行结果和预期执行结果一致时,比较第一实际执行结果和第一测试指令集的第一预期执行结果,第三实际执行结果为管理处理单元执行第一测试指令集而得到的结果。

[0022] 也就是说,在该实现方式中,管理处理单元也可以执行第一测试指令集,并根据执行结果判断管理处理单元自身是否发生了故障,并在管理处理单元没有发生故障的情况下,比较第一实际执行结果和第一测试指令集的第一预期执行结果,以提高故障检测的效果。

[0023] 在一种可能的实现方式中,管理处理单元用于比较经过第一数据压缩后的第一实际执行结果和经过第一数据压缩后的预期执行结果;当经过第一数据压缩后的第一实际执行结果和经过第一数据压缩后的预期执行结果不一致时,管理处理单元用于将第一业务处理单元识别标记为故障处理单元。

[0024] 也就是说,在该实现方式中,可以对实际执行结果和预期执行结果进行数据压缩,从而可以节省存储资源;并且,通过比较经过数据压缩实际执行结果和预期执行结果,进行故障检测,可以节省计算资源。

[0025] 在一种可能的实现方式中,数据压缩包括哈希运算和/或MD5散列加密运算。

[0026] 在一种可能的实现方式中,管理处理单元用于采用异或运算,比较第一实际执行结果和预期执行结果。

[0027] 在该实现方式中,采用异或运算进行数据比较,可以缩短比较时间,进一步提高故障检测效率。

[0028] 第二方面,提供了一种计算设备,包括用于存储第一测试指令集的存储器和第一方面所提供的处理单元集群。

[0029] 第三方面,提供了一种故障检测方法,处理单元集群包括多个处理单元,多个处理单元包括管理处理单元和第一业务处理单元,该方法包括:第一业务处理单元执行第一测试指令集,得到第一实际执行结果;管理处理单元比较第一实际执行结果和第一测试指令集的第一预期执行结果;当第一实际执行结果和第一预期执行结果不一致时,管理处理单元将第一业务处理单元标记为故障处理单元。

[0030] 在一种可能的实现方式中,多个处理单元还包括第二业务处理单元,该方法还包括:当第一实际执行结果和第一预期执行结果不一致时,第一业务处理单元将当前业务迁移到第二业务处理单元。

[0031] 在一种可能的实现方式中,该方法还包括:当第一实际执行结果和第一预期执行结果不一致时,第一业务处理单元下线。

[0032] 在一种可能的实现方式中,该方法还包括:管理处理单元在多个处理单元中故障处理单元的数量大于第一阈值时,产生告警;在数量不大于第一阈值时,不产生告警。

[0033] 第四方面,提供了一种处理单元集群,包括多个处理单元,多个处理单元包括管理处理单元和多个业务处理单元;多个业务处理单元分别用于执行第一测试指令集,得到多个实际执行结果,多个实际执行结果和多个业务处理单元一一对应;管理处理单元用于比较多个实际执行结果;当多个实际执行结果不完全一致时,管理处理单元用于指示多个业务处理单元中存在故障处理单元。

[0034] 提供的处理单元集群在进行故障检测时,可以将多个业务处理单元的实际执行结果进行相互比较,以进行故障检测。与处理单元重复执行同一测试指令集,对先后获得的测试结果进行比较的方案相比,本方案可以快速检测出该多个业务处理单元中是否存在故障处理单元,降低了故障检测的开销,并提高了故障处理单元检测效率。

[0035] 在一种可能的实现方式中,业务处理单元包括多核处理器中的计算核、CPU、GPU、专用集成电路ASIC、神经网络处理单元NPU中的任一种。

[0036] 也就是说,在该实现方式中,本申请提供的故障检测方案可以实现对多种类型的处理单元的故障检测,具有广泛的应用性。

[0037] 在一种可能的实现方式中,业务处理单元属于第一处理器架构,第一测试指令集为适配于第一处理器架构的传统测试指令集中的部分指令,该传统测试指令集还包括其他指令集。

[0038] 也就是说,在对业务处理单元进行故障检测时,业务处理单元可以执行其所属处理器架构的测试指令集中的部分指令,从而可以快速检测出故障处理单元,提高了故障检测效率。

[0039] 在一种可能的实现方式中,多个业务处理单元包括第一业务处理单元,第一业务处理单元对应多个实际执行结果中的第一实际执行结果;管理处理单元用于:在多个实际执行结果中,当与第一实际执行结果相同的实际执行结果占少数时,将第一业务处理单元标记为故障处理单元。

[0040] 也就是说,在该实现方式中,可以通过多个业务处理单元的实际执行结果进行相互比较,可以检测出故障处理单元,降低了故障检测的开销,并提高了故障处理单元检测效率。

[0041] 在一种可能的实现方式中,当多个实际执行结果完全一致时,多个业务处理单元还用于分别执行第二测试指令集,得到多个第二实际执行结果,第二测试指令集和第一测试指令集为适配于第一处理器架构的传统测试指令集中的不同指令,多个业务处理器属于第一处理器架构,多个第二实际执行结果和多个业务处理单元一一对应;当多个第二实际执行结果不完全一致时,管理处理单元用于指示多个业务处理单元中存在故障处理单元。

[0042] 也就是说,在该实现方式中,可以业务处理单元可以分批执行测试指令集,每当执行一批测试指令集时,可以进行故障处理单元检测,从而可以及早检测出处理单元集群中是否存储故障处理单元,以及标记出故障处理单元,提高了故障检测效率。

[0043] 在一种可能的实现方式中,第一测试指令集用于测试业务处理单元中的第一运算单元集合,第二测试指令集用于测试业务处理单元中的第二运算单元集合,其中,第一运算单元集合中的运算单元发生故障的可能性高于第二运算单元集合中的运算单元发生故障的可能性。

[0044] 也就是说,可以把传统的测试指令集分为多个子测试指令集,不同的子测试指令集所测试的对象不同或者说所主要针对的运算单元不同。业务处理单元可以先执行其中一个子测试指令集,并且,用于执行的子测试指令集所针对的测试对象,是容易发生故障的运算单元。因此可以在执行少量测试指令的情况下,便有较大几率检测到故障,进而可以及早将故障运算单元所在的处理单元标记为故障处理单元。当第一个子测试指令集测试完成后没有检测到运算单元故障,那么可以针对次容易发生故障的运算单元,进行下一个子测试指令集的测试。以此类推,直至检测到运算单元故障,或者所有子测试指令集测试完成。

[0045] 在一种可能的实现方式中,多个业务处理单元包括第一业务处理单元和第二业务处理单元;当第一业务处理单元为故障处理单元,且第二业务处理单元不是故障处理单元时,第一业务处理单元将当前业务迁移到第二业务处理单元。

[0046] 也就是说,在该实现方式中,故障处理单元可以将其正在执行的业务迁移到未发生故障的处理单元,从而提高了处理单元集群的容错性和可用性,改善了用户体验。

[0047] 在一种可能的实现方式中,故障处理单元下线。

[0048] 也就是说,在该实现方式中,故障处理单元可以下线,以避免执行业务,从而提高了处理单元集群的容错性和可用性,改善了用户体验。

[0049] 在一种可能的实现方式中,管理处理单元还用于:当多个业务处理单元中故障处理单元的数量大于第一阈值时,产生告警;当该数量不大于第一阈值时,不产生告警。

[0050] 也就是说,在该实现方式中,当故障处理单元的数量较多时,可以及时提醒用户,以便用户采取相应措施,从而改善了用户体验。

[0051] 在一种可能的实现方式中,当在多个实际执行结果中,与第三实际执行结果一致的实际执行结果占多数时,管理处理单元用于比较多个实际执行结果,第三实际执行结果为管理处理单元执行第一测试指令集而得到的结果。

[0052] 也就是说,在该实现方式中,管理处理单元也可以执行第一测试指令集,并根据执行结果判断管理处理单元自身是否发生了故障,并在管理处理单元没有发生故障的情况下,比较比较多个实际执行结果,以提高故障检测的效果。

[0053] 在一种可能的实现方式中,管理处理单元用于比较经过第一数据压缩后的多个实际执行结果;当经过第一数据压缩后的多个实际执行结果不完全一致时,管理处理单元用于指示多个业务处理单元中存在故障处理单元。

[0054] 也就是说,在该实现方式中,可以对实际执行结果进行数据压缩,从而可以节省存储资源,以及通过比较经过数据压缩后的多个实际执行结果,进行故障检测,从而可以节省计算资源。

[0055] 在一种可能的实现方式中,数据压缩包括哈希运算和/或MD5散列加密运算。

[0056] 在一种可能的实现方式中,管理处理单元用于采用异或运算,比较多个实际执行结果。

[0057] 在该实现方式中,采用异或运算进行数据比较,可以缩短比较时间,进一步提高故障检测效率。

[0058] 第五方面,提供了一种计算设备,包括:用于存储第一测试指令集的存储器和第四方面提供的处理单元集群。

[0059] 第六方面,提供了一种故障检测方法,处理单元集群包括多个处理单元,多个处理单元包括管理处理单元和多个业务处理单元,该方法包括:多个业务处理单元分别执行第一测试指令集,得到多个实际执行结果,多个实际执行结果和多个业务处理单元一一对应;管理处理单元比较多个实际执行结果;当多个实际执行结果不完全一致时,管理处理单元指示多个业务处理单元中存在故障处理单元。

[0060] 在一种可能的实现方式中,多个业务处理单元包括第一业务处理单元和第二业务处理单元,该方法还包括:当第一业务处理单元为故障处理单元,且第二业务处理单元不是故障处理单元时,第一业务处理单元将当前业务迁移到第二业务处理单元。

[0061] 在一种可能的实现方式中,该方法还包括:故障处理单元下线。

[0062] 在一种可能的实现方式中,该方法还包括:管理处理单元在多个业务处理单元中故障处理单元的数量大于第一阈值时,产生告警;在该数量不大于第一阈值时,不产生告警。

[0063] 第七方面,提供了一种计算机可读存储介质,包括计算机程序指令,当该计算机程

序指令由处理单元集群执行时,该处理单元集群执行如第三方面或第六方面所提供的方法。

[0064] 第八方面,提供了一种包含指令的计算机程序产品,当该指令被处理单元集群运行时,使得该处理单元集群执行如第三方面或第六方面所提供的方法。

[0065] 提供的故障检测方法、处理单元集群及计算设备,通过比较业务处理单元执行测试指令集的实际执行结果和预期执行结果,或者,将多个业务处理单元执行测试指令集的实际执行结果相互比较,以故障检测,提高了故障检测的效率,且降低了故障检测的开销。

## 附图说明

[0066] 图1为一种故障检测方法示意图;

[0067] 图2A为本申请实施例提供的一种处理单元集群结构示意图;

[0068] 图2B为本申请实施例提供的一种计算设备结构示意图;

[0069] 图3为本申请实施例提供的一种故障检测方法流程图;

[0070] 图4A为本申请实施例提供的一种故障检测方案示意图;

[0071] 图4B为本申请实施例提供的一种故障检测方案示意图;

[0072] 图5A为MD5算法效果展示图;

[0073] 图5B为哈希算法效果展示图;

[0074] 图5C为异或运算效果展示图;

[0075] 图6A为本申请实施例提供的一种故障检测方案示意图;

[0076] 图6B为本申请实施例提供的一种故障检测方案示意图;

[0077] 图6C为本申请实施例提供的一种故障检测方案示意图;

[0078] 图7为本申请实施例提供的一种故障检测方法流程图;

[0079] 图8为本申请实施例提供的一种故障检测方法流程图。

## 具体实施方式

[0080] 下面将结合附图,对本申请实施例中的技术方案进行描述。显然,所描述的实施例仅是本申请一部分实施例,而不是全部的实施例。

[0081] 在本说明书的描述中“一个实施例”或“一些实施例”等意味着在本申请的一个或多个实施例中包括结合该实施例描述的特定特征、结构或特点。由此,在本说明书中的不同之处出现的语句“在一个实施例中”、“在一些实施例中”、“在其他一些实施例中”、“在另外一些实施例中”等不是必然都参考相同的实施例,而是意味着“一个或多个但不是所有的实施例”,除非是以其他方式另外特别强调。

[0082] 其中,在本说明书的描述中,除非另有说明,“/”表示或的意思,例如,A/B可以表示A或B;本文中的“和/或”仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。另外,在本申请实施例的描述中,“多个”是指两个或两个以上。

[0083] 在本说明书的描述中,术语“第一”、“第二”仅用于描述目的,而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”的特征可以明示或者隐含地包括一个或者更多个该特征。术语“包括”、“包含”、“具有”及它

们的变形都意味着“包括但不限于”，除非是以其他方式另外特别强调。

[0084] 随着半导体技术的发展，一个处理器往往包括多个计算核心，以实现并行计算，提高处理器的计算能力。其中，处理器的计算核心也可以称为处理器核心、计算核。处理器出厂前，可以对处理器中的计算核进行故障检测，以在应用到计算设备之前，发现故障计算核。目前 出厂前的故障检测方案，难以检测出所有类型的故障；并且，在计算设备使用过程中，计算核也有可能发生新的故障。出厂前未检测出的计算核故障以及计算设备过程中发生的计算核故障，都有可能导出现网故障，影响用户业务的执行，从而响响用户体验。

[0085] 其中，为方便描述，可以将计算核发生的故障称为计算核故障，发生了故障的计算核称为故障计算核。

[0086] 计算核故障往往可导致计算核错误执行计算机指令，也就是说，故障计算核执行计算机指令的执行结果通常是错误的。因此，让计算核执行指令集，得到执行结果，再根据执行结果判断计算核是否发生了故障，是实现计算核故障检测的有效方法。

[0087] 可以将用于检测计算核是否发生了故障的指令集称为测试指令集。对于同一处理器架构的处理器而言，其适配的测试指令集往往是相对固定的，而非随意设计。通常，适配于处理器架构的测试指令集由该处理器架构设计商所设计，可以用于对采用该处理器架构的处理器进行故障检测。为方便描述，可以将该相对固定，通常由处理器架构设计商所设计的测试指令集，称为适配于处理器架构的传统测试指令集或者处理器架构的传统测试指令集。传统测试指令集所能覆盖的测试范围，包括计算核的所有运算单元(或者绝大多数运算单元)。

[0088] 在一种方案中，参阅图1，一个计算核可以先后重复执行测试指令集，得到多个执行结果。对多个执行结果进行比较，来判断该计算核是否发生了故障。该方案需要多次重复执行测试指令集，检错时间成本较高，故障检测效率较低。

[0089] 本申请实施例提供了一种故障检测方案，可以应用于包括多个处理单元的处理单元集群。通过该方案，在无需借助专门的设备，以及无需增加额外的硬件的情况下，处理单元集群可自行检测处理单元是否发生了故障，并且故障检测效果高，可以在较短时间内获知处理单元是否发生了故障。

[0090] 图2A示出了一种处理单元集群100。处理单元集群100可以包括多个处理单元，可以用于实施本申请实施例提供的故障检测方案。

[0091] 处理单元也可以称为计算单元，具有数据处理能力，可以执行数据处理任务，实现相关功能，例如数据比较、图形渲染、多媒体播放、数据收发等等。

[0092] 在一些实施例中，处理单元集群100可以为多核处理器，处理单元可以为多核处理器中的计算核。在一个例子中，处理单元集群100可以为中央处理器(central processing unit, CPU)，相应的，处理单元为CPU中的计算核。在一个例子中，处理单元集群100可以为图形处理器(graphics processing unit, GPU)，相应的，处理单元可以为GPU中的计算核。在一个例子中，处理单元集群100可以为采用ARM(advanced RISC machines)架构的多核处理器。在一个例子中，处理单元集群100可以为采用X86架构的多核处理器。

[0093] 在一些实施例中，处理单元集群100可以由多个处理器构成，也就是说，处理单元集群100包括的处理单元可以为处理器。换言之，处理单元可以是处理器(或者处理器中的计算核)。处理单元集群100可以由多个处理器构成的情况下：在一个例子中，处理单元集

群100 中的处理器可以为CPU (或者CPU中的一个内核); 在一个例子中, 处理单元集群100中的处理器可以为GPU (或者GPU中的计算核); 在一个例子中, 处理单元集群100中的处理器可以为专用集成电路(application specific integrated circuit, ASIC); 在一个例子中, 处理单元集群100中的处理器可以为神经网络处理单元(neural-network processing unit, NPU), 或者NPU中的一个计算核。

[0094] 需要说明的是, 上文对处理单元集群100所包括的处理单元进行了示例说明, 并不构成限定, 在其他实施例中, 处理单元集群100还可以包括其他类型的处理单元, 在此不再一一列举。

[0095] 在一些实施例中, 处理单元集群100中的多个处理单元可以集中式部署。也就是说, 处理单元集群100中不同的处理单元可以位于同一计算设备中, 例如, 图2B所示的计算设备 A1。在一些实施例中, 处理单元集群100中的多个处理单元可以分布式部署。也就是说, 不同的处理单元可以位于不同的计算设备中, 或者作为不同的计算节点。本申请实施例对处理单元集群100中多个处理单元的部署方式不做限定。计算设备A1例如是服务器。

[0096] 继续参阅图2A, 处理单元集群100可以包括管理处理单元110和业务处理单元(例如业务处理单元121)。也就是说, 处理单元集群100中的部分处理单元(例如一个处理单元)可以用作或者称为管理处理单元110, 其余的处理单元可以用作或称为业务处理单元。在一些实施例中, 管理处理单元110可以由处理单元集群100中预先指定的处理单元担任。例如, 处理单元集群100可以为多核处理器, 可以预先指定多核处理器中的计算核0(core 0)用作管理处理单元110。在一些实施例中, 管理处理单元110可以是处理单元集群100中的多个处理单元选举出的。具体将在下文进行介绍, 在此不再赘述。

[0097] 其中, 管理处理单元110可以用于故障检测。具体而言, 管理处理单元110可以用于判断处理单元集群100中是否具有故障处理单元, 也可以用于判断业务处理单元是否为故障处理单元。管理处理单元110同时也可以作为业务处理单元, 也就是说, 管理处理单元110可以兼具管理处理单元的功能和业务处理单元的功能。

[0098] 管理处理单元110所执行的操作以及业务处理单元所执行的操作, 将在下文进行具体介绍, 在此不再赘述。另外, 在下文描述中, 当管理处理单元和业务处理单元没有特别区分时, 它们可以被简称为处理单元。

[0099] 继续参阅图2A, 处理单元集群100中的处理单元可以连接存储器200。存储器200为具有数据存储功能的器件或装置。示例性的, 存储器200可以为一个独立的存储器。示例性的, 存储器200可以为多个存储器构成的存储器集群。本申请实施例对存储器200的实现形式不做具体限定。示例性的, 存储器200可以为内存, 例如, 随机存取存储器(random access memory, RAM) 或者其他主存储器(main memory)。示例性的, 存储器200可以为外存储器, 例如, 硬盘。存储器200可以用于存储数据, 以供处理单元调用。其中, 存储器200可以存储测试指令集, 例如测试指令集201。处理单元可以执行测试指令集201, 得到实际执行结果。实际执行结果可以用于管理处理单元110进行故障检测。存储器200还可以存储供处理单元运行测试指令集的程序代码。在一些实施例中, 存储器200还可以存储其他数据, 例如测试指令集201的预期执行结果2011、测试指令集202、测试指令集202的预期执行结果2021等。具体将在下文进行具体介绍, 在此不再赘述。

[0100] 在一些实施例中, 参阅图2B, 处理单元集群100和存储器200可以部署到计算设备

A1 中,其中,处理单元集群100中的处理单元可以作为计算设备A1的处理器(例如CPU、GPU),执行计算设备A1的相关业务,实现计算设备A1的相关功能,例如,处理单元的故障检测、图形渲染、多媒体播放、数据收发功能、导航等等。存储器200可以作为计算设备A1的存储部件,存储计算设备A1的相关数据,例如测试指令集201等。

[0101] 上文示例介绍了本申请实施例提供的故障检测方案可应用的架构。接下来,示例介绍该故障检测方案的流程。

[0102] 参阅图3,可以启动故障检测方案的流程。示例性的,该流程可以自行启动,例如在处理单元集群100上电后,可以启动故障检测方案的流程。流程启动之后,可以执行步骤301,加载配置文件B1。可以设定步骤301可以由处理单元C1执行。在一些实施例中,处理单元C1可以是处理单元集群100中预先指定的处理单元(例如计算核0)。在一些实施例中,处理单元C1可以是处理单元集群100中任意处理单元。在步骤301中,配置文件B1可以被加载到存储器200中,以便在后续步骤中供处理单元调用。示例性的,在步骤301中,当完成配置文件加载时,存储器200可以提供处理单元C1提供配置文件B1的调用接口,以便处理单元C1利用该调用接口调用配置文件B1,以及指示其他处理单元利用该调用接口调用配置文件B1。配置文件B1为预先配置的文件,可以包括测试指令集201等。配置文件B1所包括的信息或者说数据,将在下文结合具体方案进行介绍,在此不再赘述。

[0103] 在一些实施例中,处理单元C1还可以执行步骤302,读取故障检测策略,并根据故障检测策略,指示处理单元集群100中的业务处理单元执行测试指令集,以便进行故障检测。其中,故障检测策略可以包括在配置文件B1中,被加载到存储器200中。由此,在步骤302中,处理单元C1可以从存储器200中读取故障检测策略。示例性的,故障检测策略可以包括故障检测开关。其中,故障检测开关用于指示是否进行故障检测。若故障检测开关指示不进行故障检测,则可以终止故障检测方案的流程。若故障检测开关执行进行故障检测,则可以继续进行后续步骤。示例性的,故障检测策略可以包括待检测处理单元指示信息。待检测处理单元指示信息用于指示哪些处理单元需要故障检测。其中,待检测处理单元指示信息可以指示处理单元集群100中所有业务处理单元都需要进行故障检测,也可以指示处理单元集群100中部分业务处理单元需要进行故障检测。示例性的,故障检测策略可以包括故障检测周期。故障检测周期用于表示每间隔预设时长进行一轮故障检测。其中,每一轮故障检测可以包括下文即将介绍的步骤303和步骤304,还可以包括下文即将介绍的步骤305和步骤306。

[0104] 继续参阅图3,业务处理单元可以执行步骤303,执行测试指令集201,得到实际执行结果。其中,处理单元C1可以向业务处理单元提供调用接口,业务处理单元可以根据调用接口,调用并执行配置文件B1中的测试指令集201,得到实际执行结果。管理处理单元110可以执行步骤304,判断业务处理单元是否发生了故障。

[0105] 接下来,对步骤303和步骤304的具体执行过程进行示例介绍。

[0106] 在一些实施例中,以业务处理单元121为例,在步骤303中,业务处理单元121可以执行测试指令集201,得到实际执行结果E1。在步骤304,管理处理单元110可以比较实际执行结果E1和预期执行结果2011。其中,预期执行结果2011为测试指令集201的预期执行结果。预期执行结果也可以称为正确执行结果,其是预设的,或者是经由其他处理单元正常执行测试指令集201而得到的执行结果。该其他处理单元为无故障处理单元。

[0107] 示例性的,参阅图4A,预期执行结果2011可以包括在配置文件B1中,从而可以被加载到存储器200中。管理处理单元110可以从存储器200调用预期执行结果2011。

[0108] 示例性的,参阅图4A,业务处理单元121可以将实际执行结果E1存放到存储器200中,管理处理单元110在执行步骤304时,可以从存储器200调用实际执行结果E1,以及调用预期执行结果2011,从而可以将实际执行结果E1和预期执行结果2011进行比较。

[0109] 若实际执行结果E1和预期执行结果2011不一致,则意味着执行结果E1是错误的执行结果,那么生成计算结果E1的处理单元就是不可靠的。因此管理处理单元110可以将业务处理单元121标记为故障处理单元。

[0110] 可以设定业务处理单元121所属的处理器架构为处理器架构D1,测试指令D11为适配于处理器架构D1的测试指令集。在一个例子中,处理器架构D1可以为ARM架构。在另一个例子中,处理器架构D1可以为X86架构。等等,此处不再一一列举。如上文所述,测试指令D11通常由处理器架构的设计商所设计,相对固定。即,测试指令D11为处理器架构D1的传统测试指令集。在本申请的一些实施例中,可以将测试指令集D11中的部分测试指令用作测试指令集201,即测试指令集201为适配于处理器架构D1的测试指令集中的部分指令。换言之,测试指令集D11包括测试指令集201,还包括其他测试指令集。在步骤303中,业务处理单元121执行测试指令集D11中的部分测试指令,可以在较短时间内执行完毕,从而管理处理单元110可以及时执行步骤304,通过比较实际执行结果E1和预期执行结果2011,来判断业务处理单元121是否为故障处理单元,提高了故障检测的效率。

[0111] 在一些实施例中,测试指令集D11包括多个指令子集,其中,不同的指令子集可以用于测试业务处理单元的不同运算单元。其中,指令子集也可以称为子测试指令集。运算单元可以为算术逻辑单元(arithmetic and logic unit, ALU)和/或浮点运算单元(float point unit, FPU)。不同的运算单元发生故障的概率不同。相对而言,有的运算单元容易发生故障,有的运算单元不容易发生故障。示例性的,可以将容易发生故障的运算单元称为高危运算单元,不容易发生故障的运算单元称为低危运算单元。其中,高危运算单元可以为业务处理单元中的一个或多个运算单元,低危运算单元可以为该业务处理单元中的另外一个或多个运算单元。相应地,可以将测试指令集D11中用于测试高危运算单元的指令子集称为高危测试指令集,用于测试低危运算单元的指令子集称为低危测试指令集。上文所述的测试指令集201可以为高危测试指令集。

[0112] 在这些实施例的一个说明性示例中,回到图2A,存储器200还可以存储有测试指令集202和测试指令集202的预期执行结果2021。其中,测试指令集202也可以为测试指令集D11中的部分测试指令,且测试指令集202和测试指令集201不同。在实际执行结果E1和预期执行结果2011一致的情况下,业务处理单元121还可以执行测试指令集202,得到实际执行结果E2。管理处理单元110可以比较实际执行结果E2和预期执行结果2021。若两者不一致,则管理处理单元110可以将业务处理单元121标记为故障处理单元。

[0113] 示例性的,测试指令集202也可以为高危测试指令集。也就是说,测试指令集201是测试指令集D11中的高危测试指令集中的一部分,测试指令集202是测试指令集D11中的高危测试指令集中的另一部分。在该示例中,业务处理单元121可以分段执行高危测试指令集,并根据每个阶段的执行结果,判断业务处理单元121是否发生了故障,从而可以在无需等全部高危测试指令集执行完的情况下,及早检测出业务处理单元121发生了故障。

[0114] 示例性的,测试指令集202可以为低危测试指令集。在该示例中,业务处理单元121可以优先执行高危测试指令集,以便及早检测出业务处理单元121发生了故障。在高危运算单元没有发生故障的情况下,业务处理单元121可以执行低位测试指令集,实现全面的故障检测。

[0115] 通常,实际执行结果的比特位数较多(例如,256个比特位),管理处理单元110需要占用较长时间,才能完成实际执行结果和预期执行结果的比较。在一些实施例中,为了提高管理处理单元110在比较实际执行结果和预期执行结果时的比较效率,以及节省存储资源,可以对实际执行结果进行数据压缩。示例性的,业务处理单元可以对其产生的实际执行结果进行数据压缩。示例性的,管理处理单元110在从存储器200读取到实际执行结果时,可以对实际执行结果进行数据压缩。

[0116] 另外,当采用进行数据压缩后的实际执行结果和预期执行结果进行比较时,该预期执行结果也为经过了数据压缩后的执行结果。其中,预期执行结果的数据压缩方式和实际执行结果的数据压缩方式是相同的。示例性的,预期执行结果可以预先进行数据压缩,也就是说,在步骤301中加载的配置文件B1包括了数据压缩后的预期执行结果。接下来,以实际执行结果的数据压缩为例,介绍数据压缩方式。

[0117] 在一些实施例中,可以采用散列加密算法对实际执行结果进行数据压缩。散列加密算法可以将任意长度的二进制数据转化或者说加密为固定长度的二进制数据,也就是说,不同长度的二进制的的数据可以被散列加密算法转化为同样长度的二进制数据。并且,不同的原始数据,被散列加密算法转化后的数据也不相同;相同的原始数据,被散列加密算法转化后的数据也相同。也就是说,散列加密算法可以保证转化前的数据和转化后的数据具有一一对应关系。

[0118] 在一个例子中,本申请实施例具体可以采用MD5信息摘要算法(MD5 message-digest algorithm)实现数据压缩。参阅图5A,MD5信息摘要算法也可以简称为MD5算法,其为一种较为典型的散列加密算法,可以将任意长度的二进制数据转化为32个比特位的二进制数据。如此,利用MD5信息摘要算法可以将比特位数较多的实际执行结果,转化为只有32个比特位数的二进制数据,该数据即为数据压缩后的实际执行结果。由此,可以实现对实际执行结果的压缩。

[0119] 在一些实施例中,可以采用哈希(hash)算法对实际执行结果进行数据压缩。哈希算法的功能与散列加密算法的功能类似,可以将任意长度的二进制数据转化或者说加密为固定长度的二进制数据。并且,不同的原始数据,被哈希算法转化后的数据也不相同;相同的原始数据,被哈希算法转化后的数据也相同。

[0120] 在一个例子中,可以设定采用的哈希算法将实际执行结果转化为一个阿拉伯数字所占据的比特长度。参阅图5B,为直观示出哈希算法的数据压缩效果,采用了阿拉伯数字表示压缩后的实际执行结果。如图5B所示,实际执行结果1可被哈希算法转化为“0”,与实际执行结果1不同的实际执行结果2被哈希算法转化为“1”,与实际执行结果1相同的实际执行结果3被哈希算法转化为“0”。由此,可以实现对实际执行结果的压缩。

[0121] 经过上述数据压缩方案,可以将业务处理单元121的实际执行结果(实际执行结果E1和/或实际执行结果E2)压缩为较短的数据,同理,预期执行结果(预期执行结果2011和/或预期执行结果2011)可以也可以被压缩为与实际执行结果长度相同的数据。管理处理单

元110 可以比较经过数据压缩后的实际执行结果和预期执行结果,可以快速完成比较,提高了比较效率,可以尽快判断出业务处理单元121是否发生了故障。并且,压缩后的实际执行结果占用的存储空间较少,可以节省存储资源。

[0122] 在一些实施例中,为了提高管理处理单元110在比较实际执行结果和预期执行结果时的工作效率,可以采用异或运算的方式,对实际执行结果和预期执行结果进行比较。参阅图5C,对于需要比较的两个或更多个数据,异或运算可以逐比特位比较比特值,当两个或更多个数据在某一比特位上的比特值完全相同时,运算结果为0。当两个或更多个数据在某一比特位上的比特值不完全相同时,运算结果为1。因此,当比较结果出现1的时候,可以判断出这两个或更多个数据不完全相同,从而可以快速完成比较,提高了管理处理单元110的比较效率。

[0123] 在一些实施例中,回到图2A,处理单元集群100还可以包括业务处理单元122等。业务处理单元122也可以执行测试指令集,得到实际执行结果。管理处理单元110可以比较业务处理单元122的实际执行结果和预期执行结果,以判断业务处理单元122是否发生了故障。具体实现过程可以参考上文对步骤303和步骤304的介绍,在此不再赘述。

[0124] 在一些实施例中,如上所述,管理处理单元110可以作为业务处理单元,也执行测试指令集,得到实际执行结果。然后,管理处理单元110可以比较该实际执行结果和预期执行结果,进而判断管理处理单元是否发生了故障。

[0125] 在一个说明性示例中,管理处理单元110在对业务处理单元121等业务处理单元的实际执行结果和预期结果进行比较之前,管理处理单元110可以先比较管理处理单元110的实际执行结果和预期执行结果,在确定管理处理单元110自身没有发生故障的情况下,管理处理单元110再去比较其他处理单元的实际执行结果和预期执行结果,以免发生故障误判。

[0126] 在一个说明性示例中,如上所述,管理处理单元110可以是处理单元集群100中的多个处理单元选举出的。处理单元集群100中发生故障的处理单元通常少数的,基于此,可以进行管理处理单元的选举。具体而言,处理单元集群100中的多个处理单元分别执行测试指令集201,得到实际执行结果。该多个处理单元中的一个处理单元,例如处理单元F1,可以比较多个处理单元的实际执行结果。示例性的,处理单元F1比较多个处理单元的实际执行结果具体可以为,将处理单元F1的实际执行结果分别与其他处理单元的实际执行结果一一进行比较。若在多个处理单元的实际执行结果中,与处理单元F1的实际执行结果相同的实际执行结果占多数,则可以选举处理单元F1为管理处理单元110。若在多个处理单元的实际执行结果中,与处理单元F1的实际执行结果相同的实际执行结果不占多数,则该多个处理单元中的另一个处理单元,例如处理单元F2,可以比较多个处理单元的实际执行结果。示例性的,处理单元F2比较多个处理单元的实际执行结果具体可以为,将处理单元F2的实际执行结果分别与其他处理单元的实际执行结果一一进行比较。若在多个处理单元的实际执行结果中,与处理单元F2的实际执行结果相同的实际执行结果占多数,则可以选举处理单元F2为管理处理单元110。若在多个处理单元的实际执行结果中,与处理单元F2的实际执行结果相同的实际执行结果不占多数,则继续选择其他处理单元进行上述比较,直到选举出管理处理单元110。

[0127] 在一些实施例中,参阅图4B,可以设定处理单元集群100为多核处理器,处理单元

集群 100 中的处理单元为多核处理器中的计算核。其中,多核处理器中的计算核0可以用作管理 处理单元110,多核处理器中的计算核1 (core 1) 用作业务处理单元121,多个处理器中的 计算核2 (core 2) 用作业务处理单元122,……,计算核N (core N) 用作业务处理单元 M, 等等。

[0128] 如图4B所示,管理处理单元110、业务处理单元121、业务处理单元122等处理单元集 群100中的处理单元可以分别从存储器200读取执行测试指令集201,并将得到的实际执 行 结果存放到存储器200。示例性的,处理单元可以将其实际执行结果进行数据压缩后,再 存放 到存储器200,可以节省存储器200的存储资源,并提高后续步骤中的比较效率。管理 处理单 元110可以从存储器200读取预期执行结果2011和每个处理单元的实际执行结果, 并比较预 期执行结果2011和每个处理单元的实际执行结果,以判断每个处理单元是否发 生了故障。示 例性的,管理处理单元110可以采用异或运算,比较预期执行结果2011和各处 理单元的实际 执行结果,以提高比较效率。示例性的,当任一处理单元的实际执行结果和 预期执行结果2011 一致时,该处理单元还可以从存储器200中读取测试指令集202,并得到 再次实际执行结果。管理处理单元100可以比较再次实际执行结果和预期执行结果2021, 以再次判断该处理单元 是否发生了故障。

[0129] 在上述方案中,通过将业务处理单元的实际执行结果和预期执行结果进行比较, 可以快 速检测出业务处理单元是否发生了故障。

[0130] 本申请实施例还提供了步骤303和步骤304的另一种实现方式。该实现方式也可以 称为 空间冗余方案,可以通过比较多个业务处理单元的实际执行结果,判断处理单元集群 100中 是否存在故障处理单元,以及哪些业务处理单元为故障处理单元。具体而言,在步骤 303,该 多个业务处理单元可以分别执行测试指令集,得到实际执行结果。在步骤304,管理 处理单元 可以比较多个业务处理单元的实际执行结果。若多个业务处理单元的实际执行 结果不完全一 致,则管理处理单元可以判断出该多个业务处理单元中具有故障处理单元。 示例性的,若在 多个业务处理单元的实际执行结果中,与某业务处理单元的实际执行结果 相同的实际执行结 果占少数,则可以将该业务处理单元标记为故障处理单元。

[0131] 接下来,示例描述空间冗余方案。

[0132] 参阅图6A,在步骤303中,业务处理单元121可以执行测试指令集201,得到实际执 行 结果G1。在步骤303中,业务处理单元122也可以执行测试指令集201,得到实际执行结果 G2。在步骤304,管理处理单元110可以比较实际执行结果G1和实际执行结果G2。若实际执 行结果G1和实际执行结果G2不一致,则可以判断出实际执行结果G1或实际执行结果G2为 故障处理单元,也就是说,处理单元集群100中存在故障处理单元。

[0133] 在空间冗余方案中,管理处理单元110可以由预先指定的处理单元担任,也可以由 处理 单元集群100中的多个处理单元选举得到。具体可以参考上文描述,在此不再赘述。

[0134] 在一些实施例中,处理单元集群100还可以包括更多业务处理单元,例如业务处理 单元 M。在步骤303,业务处理单元M也可以执行测试指令集201,得到实际执行结果Gm。管理 处 理单元110可以比较实际执行结果G1、实际执行结果G2和实际执行结果Gm。若实际执行 结 果G1和实际执行结果Gm一致,实际执行结果G2和实际执行结果G1不一致。即在多个实际 执行结果中,与 实际执行结果G2的实际执行结果占少数,则可以判断出业务处理单元122为 故障处理单元,可以将业务处理单元122标记为故障处理单元。

[0135] 在一些实施例中,管理处理单元110也用作业务处理单元,在步骤303中执行测试指令集201,得到实际执行结果G3。管理处理单元110可以比较实际执行结果G1、实际执行结果G2和实际执行结果G3。若实际执行结果G1和实际执行结果G3一致,实际执行结果G2和实际执行结果G3不一致。即在多个实际执行结果中,与实际执行结果G2的实际执行结果占少数,则可以判断出业务处理单元122为故障处理单元,可以将业务处理单元122标记为故障处理单元。

[0136] 在一些实施例中,参阅图6B,业务处理单元可以从存储器200读取测试指令集201,并将实际执行结果存放到存储器200中。管理处理单元110可以从存储器200读取实际执行结果,并进行比较。

[0137] 在一些实施例中,业务处理单元可以对实际执行结果进行数据压缩,然后,在存放到存储器200中,以节省存储器200的存储资源以及提高后续步骤的比较效率。数据压缩具体可以参考上文对图5A和图5B所示实施例的介绍实现。

[0138] 在一些实施例中,管理处理单元110可以采用异或运算,比较多个实际执行结果,以节省。具体可以参考上文对图5C所示实施例的介绍实现。

[0139] 在一些实施例中,可以设定处理单元集群100中的多个业务处理单元采用的处理器架构为处理器架构D1,测试指令集201可以为适配于处理器架构D1的测试指令集中的部分指令。具体可以参考上文介绍,在此不再赘述。

[0140] 在一些实施例中,当多个业务处理单元执行测试指令集201的实际执行结果完全一致时,多个业务处理单元还可以分别执行测试指令集202,得到再次实际执行结果。测试指令集202具体可以参考上文介绍。然后,管理处理单元110可以比较多个业务处理单元的再次实际执行结果。若不完全相同,则可以判断出处理单元集群100中存在故障处理单元。当在多个业务处理单元的再次实际执行结果中,与某业务处理单元的再次实际执行结果相同的实际执行结果占少数时,可以将该业务处理单元标记为故障处理单元。其中,测试指令集201和测试指令集202以及它们的执行过程可以参考上文介绍,在此不再赘述,

[0141] 在一些实施例中,管理处理单元110在比较多个实际执行结果时,可以两两进行比较。例如,可以将一实际执行结果逐一和其他实际执行结果进行两两比较。在一些实施例中,管理处理单元110在比较多个实际执行结果时,可以先将多个实际执行结果进行综合比较,例如利用异或算法进行比较,以快速发现多个实际执行结果是否完全相同。若多个实际执行结果不完全相同,可以采用两两比较的方式,以发现哪些实际执行结果相同,哪些实际执行结果不同。

[0142] 在一些实施例中,参阅图6C,可以设定处理单元集群100为多核处理器,处理单元集群100中的处理单元为多核处理器中的计算核。其中,多核处理器中的计算核0可以用作管理处理单元110,多核处理器中的计算核1(core 1)用作业务处理单元121,多个处理器中的计算核2(core 2)用作业务处理单元122,计算核N(core N)用作业务处理单元M,等等。

[0143] 如图6C所示,管理处理单元110、业务处理单元121、业务处理单元122等处理单元集群100中的处理单元可以分别从存储器200读取执行测试指令集201,并将得到的实际执行结果存放到存储器200。示例性的,处理单元可以将其实际执行结果进行数据压缩后,再存放到存储器200,可以节省存储器200的存储资源,并提高后续步骤中的比较效率。管理

处理单元110可以从存储器200读取各处理单元的实际执行结果,并比较各处理单元的实际执行结果,以判断处理单元集群100中是否存在故障处理单元,以及哪些处理单元为故障处理单元。示例性的,管理处理单元110可以采用异或运算,比较各处理单元的实际执行结果,以提高比较效率。示例性的,当各处理单元的实际执行结果完全一致时,各处理单元还可以从存储器200中读取测试指令集202,并得到再次实际执行结果。管理处理单元100可以比较各处理单元的再次实际执行结果,以再次判断处理单元集群100中是否存在故障处理单元,以及哪些处理单元为故障处理单元。

[0144] 通过空间冗余方案,可以在预设预期执行的结果的情况下,快速判断出处理单元集群是否存在故障单元,以及检测出故障单元。

[0145] 上文示例介绍了步骤303和步骤304的具体实现过程。接下来,描述图3中其余步骤的实现过程。

[0146] 参阅图3,在一些实施例中,本申请实施例提供的故障检测方案还包括步骤305,迁移业务。具体而言,当检测出故障处理单元时,故障处理单元可以执行步骤305,迁移业务。具体而言,故障处理单元可以将其当前业务迁移到其他处理单元。其中,故障处理单元的当前业务是指故障处理单元当前正在执行的业务,例如,图形渲染业务等。示例性的,故障处理单元可以将其当前业务迁移到正常处理单元。其中,正常处理单元可以是指步骤303和步骤304未检测出故障的处理单元。管理处理单元110可以将正常处理单元通知给故障处理单元,例如,将正常处理单元的标识信息通知给故障处理单元。由此,故障处理单元可以将其当前业务迁移到正常处理单元。

[0147] 参阅图3,在一些实施例中,本申请实施例提供的故障检测方案还包括步骤306,故障处理单元下线。示例性的,故障处理单元下线可以是指屏蔽处理单元。在一个例子中,管理处理单元110或其他处理单元(例如故障处理单元)可调用内核层处理器驱动可以使用echo命令屏蔽该故障处理单元。在一个具体例子中,故障处理单元可以为编号为3的计算机,该echo命令可以为echo 0>/sys/devices/system/cpu/cpu3/online。其中,cpu3表示该故障处理单元。还可以采用其他方式使得故障处理单元下线,在此不再一一赘述。

[0148] 继续参阅图3,本申请实施例提供的故障检测方案还包括步骤307,判断处理单元集群100中的故障处理单元的数量是否大于门限数量。其中,门限为预设值。示例性的,步骤307可以由管理处理单元110执行。

[0149] 本申请实施例提供的故障检测方案还包括步骤308,当处理单元集群100中的故障处理单元的数量大于门限数量时,可以产生告警。其中,当处理单元集群100中的故障处理单元的数量不大于门限数量时,可以不产生告警。示例性的,步骤308可以由管理处理单元110执行。告警信息用于向用户发出告警,以使用户对处理单元集群100采取相应措施,例如维修处理单元集群100,或者停用处理单元集群100,等等。

[0150] 由此,本申请实施例提供的故障检测方案可以在实现高效检测故障处理单元的同时,还可以对故障处理单元进行处理,以提高用户的体验。

[0151] 综合以上,申请实施例提供的故障检测方案具体如下优势。

[0152] 首先,本申请提供的故障检测方案无需增加或借助外部的硬件,可以通过软件调度实现故障处理单元检测,从而具有更广泛的应用范围,不但可以应用于新型号的处理单元,也可以在不更换硬件的情况下应用于原来型号的处理单元。

[0153] 其次,本申请实施例提供的故障检测方案可以采用将实际执行结果和预期执行结果进行比较的方式,来实现故障处理单元的检测。从而无需实时计算预期执行结果,节省了计算资源以及提高了故障检测效率,并且,可以保障预期执行结果的准确,提高了故障检测效果。本申请实施例提供的故障检测方案还可以采用空间冗余方案,实现故障处理单元的检测,可以提高故障检测效率,特别是在处理单元集群中处理单元个数较多的情况下,可以快速判断处理单元集群是否存在故障处理单元,以及哪些处理单元为故障处理单元。例如,当处理单元集群具有100个处理单元时,可以毫秒级判断出处理单元集群中是否存在故障处理单元,以及秒级检测出故障处理单元。

[0154] 并且,本申请实施例提高的故障检测方案可以通过部分测试指令集,实施故障检测,进一步提高了检测效率,可以快速检测出故障处理单元。

[0155] 以及,本申请实施例的故障检测方案可以对实际执行结果进行压缩,从而可以节省存储资源,以及提高故障检测的效率。具体可以将实际执行结果所占用的存储空间减少为传统方案的10%。并且,将压缩后的实际执行结果用于故障检测,进一步提高了检测效率。

[0156] 而且,本申请实施例提高的故障检测方案可以对故障处理单元进行进一步处理,提高处理单元集群的系统可用性,降低处理单元集群的返修率。

[0157] 基于上文所描述的故障检测方案,本申请实施例提供了一种故障检测方法。可以理解的是,该方法是上文所描述的故障检测方案的另一种表达方式,两者是相结合的,该方法中的部分或全部内容可以参见上文对故障检测方案的描述。

[0158] 本申请实施例提供的故障检测方法可以应用于处理单元集群,处理单元集群包括多个处理单元,该多个处理单元包括管理处理单元和第一业务处理单元。参阅图7,该方法包括如下步骤。

[0159] 步骤701,第一业务处理单元执行第一测试指令集,得到第一实际执行结果。具体可以参考上文对图3和图4A中的步骤303、以及图4B的介绍。

[0160] 步骤702,管理处理单元比较所述第一实际执行结果和所述第一测试指令集的第一预期执行结果。具体可以参考上文对图3和图4A中的步骤304、以及图4B的介绍。

[0161] 步骤703,当所述第一实际执行结果和所述第一预期执行结果不一致时,所述管理处理单元将所述第一业务处理单元标记为故障处理单元。具体可以参考上文对图3和图4A中的步骤304、以及图4B的介绍。

[0162] 在一些实施例中,第一业务处理单元为多核处理器中的计算核、CPU、GPU、专用集成电路ASIC、神经网络处理单元NPU中的任一种。具体可以参考上文对图2A中业务处理单元的介绍。

[0163] 在一些实施例中,第一业务处理单元属于第一处理器架构,第一测试指令集为适配于第一处理器架构的传统测试指令集中的部分指令,传统测试指令集还包括其他指令集。具体可以参考上文对测试指令集201和处理器架构D1的介绍。

[0164] 在一些实施例中,当第一实际执行结果和第一预期执行结果不一致时,管理处理单元110用于将第一业务处理单元标记为故障处理单元;当第一实际执行结果和第一预期执行结果一致时,第一业务处理单元还用于执行第二测试指令集,得到第二实际执行结果,第二测试指令集和第一测试指令集为适配于第一处理器架构的传统测试指令集中的

不同指令,第一业务处理器属于第一处理器架构;当第二实际执行结果和第二测试指令集的第二预期执行结果不一致时,管理处理单元用于将业务处理单元标记为故障处理单元。具体可以参考上文对测试指令集201和测试指令集202的介绍。

[0165] 在一些实施例中,第一测试指令集用于测试第一业务处理单元中的第一运算单元集合,第二测试指令集用于测试第一业务处理单元中的第二运算单元集合,其中,第一运算单元集合中的运算单元发生故障的可能性高于第二运算单元集合中的运算单元发生故障的可能性。具体可以参考上文对高危测试指令集和低危测试指令集的介绍,在此不再赘述。

[0166] 在一些实施例中,该多个处理单元还包括第二业务处理单元;当第一实际执行结果和第一预期执行结果不一致时,第一业务处理单元将当前业务迁移到第二业务处理单元。具体可以参考上文对图3中步骤305的介绍。

[0167] 在一些实施例中,当第一实际执行结果和第一预期执行结果不一致时,第一业务处理单元下线。具体可以参考上文对图3中步骤306的介绍。

[0168] 在一些实施例中,管理处理单元在该多个处理单元中故障处理单元的数量大于第一阈值时,产生告警;在该多个处理单元中故障处理单元的数量不大于第一阈值时,不产生告警。具体可以参考上文对图3中步骤307的介绍。

[0169] 本申请实施例提供的故障检测方法在进行故障检测时,可以通过比较测试指令集的实际执行结果和预期执行结果,进行故障检测,可以在降低故障检测开销的同时,快速检测出故障处理单元,提高了故障检测效率。

[0170] 基于上文所描述的故障检测方案,本申请实施例还提供了一种故障检测方法。该方法中的部分或全部内容可以参见上文对故障检测方案的描述。

[0171] 该方法可应用于处理单元集群,处理单元集群包括多个处理单元,多个处理单元包括管理处理单元和多个业务处理单元。参阅图8该方法包括如下步骤。

[0172] 步骤801,多个业务处理单元分别执行第一测试指令集,得到多个实际执行结果,多个实际执行结果和多个业务处理单元一一对应。具体可以参考上文对图3、图6A、图6B中步骤303,以及图6C的介绍。

[0173] 步骤802,管理处理单元比较多个实际执行结果。具体可以参考上文对图3、图6A、图6B中步骤304,以及图6C的介绍。

[0174] 步骤803,当多个实际执行结果不完全一致时,管理处理单元指示多个业务处理单元中存在故障处理单元。具体可以参考上文对图3、图6A、图6B中步骤304,以及图6C的介绍。

[0175] 在一些实施例中,业务处理单元包括多核处理器中的计算核、CPU、GPU、专用集成电路ASIC、神经网络处理单元NPU中的任一种。具体可以参考上文对图2A中业务处理单元的介绍。

[0176] 在一些实施例中,业务处理单元属于第一处理器架构,第一测试指令集为适配于第一处理器架构的传统测试指令集中的部分指令,传统测试指令集还包括其他指令集。具体可以参考上文对测试指令集201和处理器架构D1的介绍。

[0177] 在一些实施例中,多个业务处理单元包括第一业务处理单元,第一业务处理单元对应多个实际执行结果中的第一实际执行结果;管理处理单元用于:在多个实际执行结果

中,当与 第一实际执行结果相同的实际执行结果占少数时,将第一业务处理单元标记为故障处理单元。具体可以参考上文对图3、图6A、图6B中步骤303,以及图6C的介绍。

[0178] 在一些实施例中,当多个实际执行结果完全一致时,多个业务处理单元还用于分别执行 第二测试指令集,得到多个第二实际执行结果,第二测试指令集和第一测试指令集为适配于 第一处理器架构的传统测试指令集中的不同指令,多个业务处理器属于第一处理器架构,多 个第二实际执行结果和多个业务处理单元一一对应;当多个第二实际执行结果不完全一致时, 管理处理单元用于指示多个业务处理单元中存在故障处理单元。具体可以参考上文对测试指 令集201和测试指令集202的介绍。

[0179] 在一些实施例中,第一测试指令集用于测试业务处理单元中的第一运算单元集合,第二 测试指令集用于测试业务处理单元中的第二运算单元集合,其中,第一运算单元集合中的运 算单元发生故障的可能性高于第二运算单元集合中运算单元发生故障的可能性。具体可以参 考上文对高危测试指令集和低危测试指令集的介绍,在此不再赘述。

[0180] 在一些实施例中,多个业务处理单元包括第一业务处理单元和第二业务处理单 元,该方 法还包括:当第一业务处理单元为故障处理单元,且第二业务处理单元不是故障 处理单元时, 第一业务处理单元将当前业务迁移到第二业务处理单元。具体可以参考上文 对图3中步骤305 的介绍。

[0181] 在一些实施例中,该方法还包括:故障处理单元下线。具体可以参考上文对图3中 步骤 306的介绍。

[0182] 在一些实施例中,该方法还包括:管理处理单元在多个业务处理单元中故障处理 单元的数量大于第一阈值时,产生告警;在该多个业务处理单元中故障处理单元的数量不 大于第一 阈值时,不产生告警。具体可以参考上文对图3中步骤307的介绍。

[0183] 本申请实施例提供的故障检测方法在进行故障检测时,可以将多个业务处理单元 的实际 执行结果进行相互比较,以进行故障检测,从而可以快速检测出该多个业务处理单 元中是否 存在故障处理单元,降低了故障检测的开销,并提高了故障处理单元检测效率。

[0184] 本申请的实施例中的方法步骤可以通过硬件的方式来实现,该硬件可以包括处理 单元以 及耦合至处理单元的存储器。处理单元可以从存储器读取信息,以及可以向存储器 写入信息, 以执行上述方法步骤。

[0185] 可以理解的是,在本申请的实施例中涉及的各种数字编号仅为描述方便进行的区 分,并 不用来限制本申请的实施例的范围。

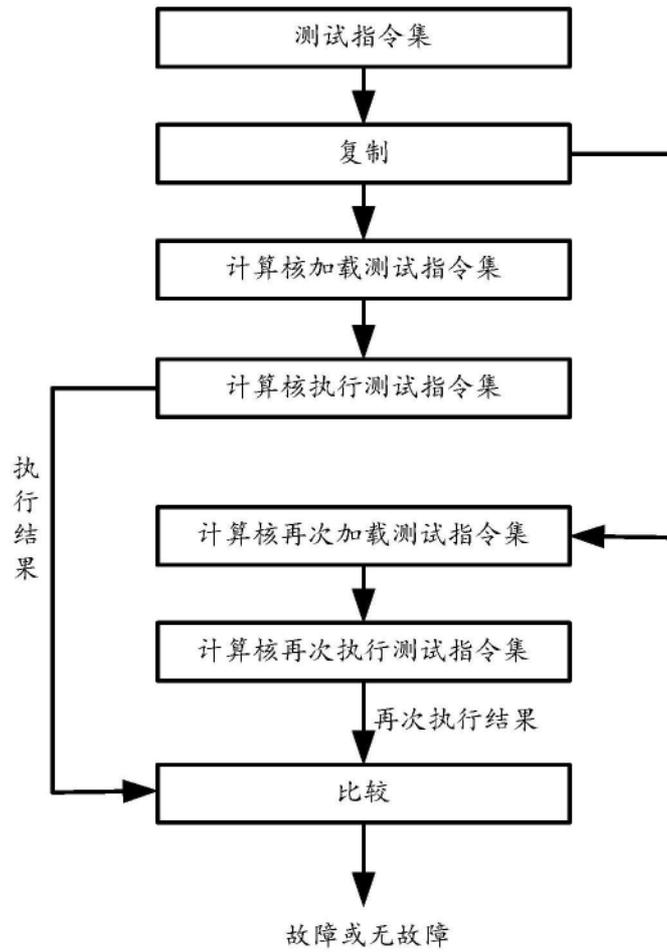


图1

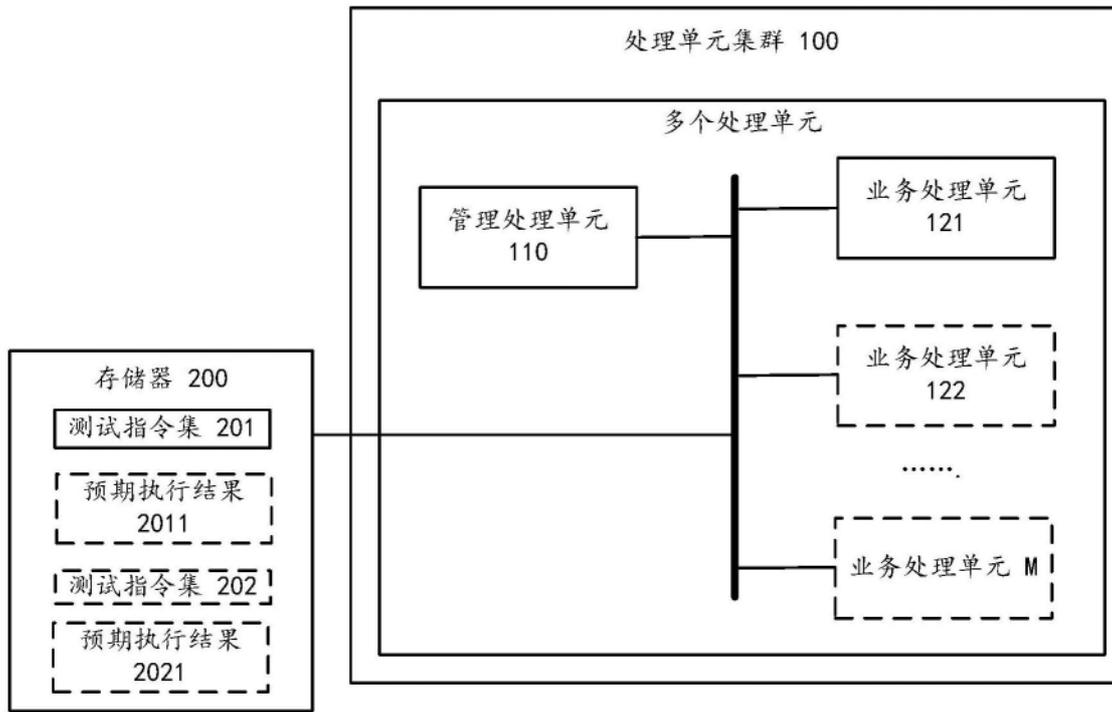


图2A

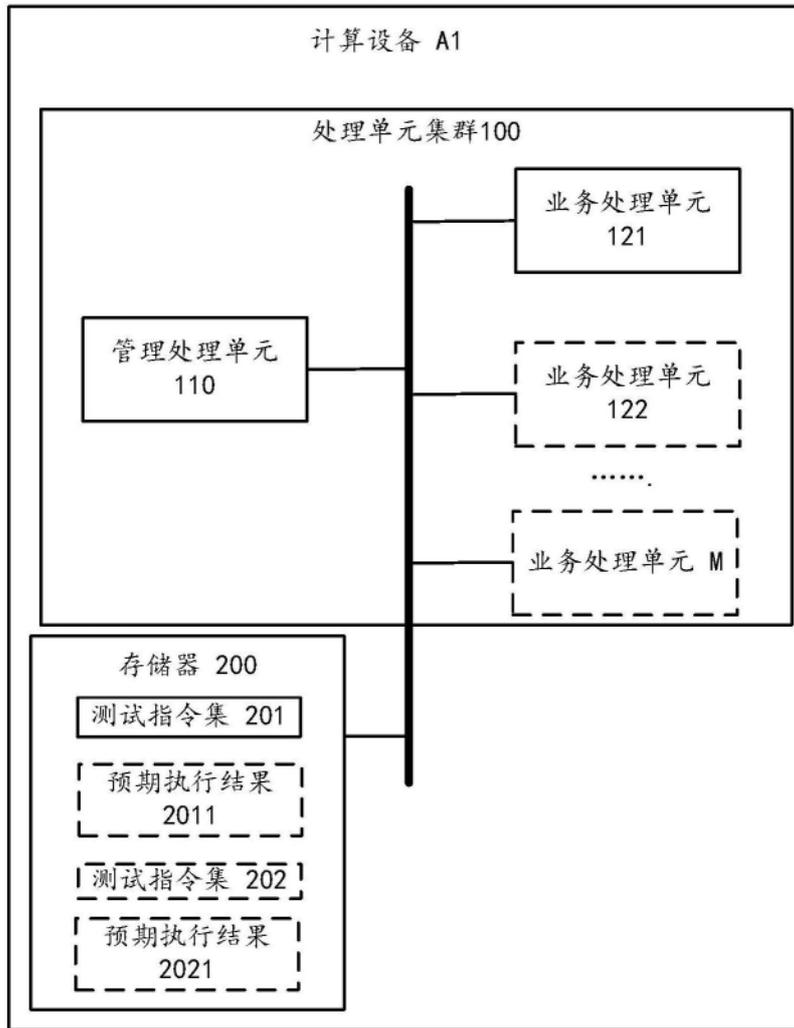


图2B

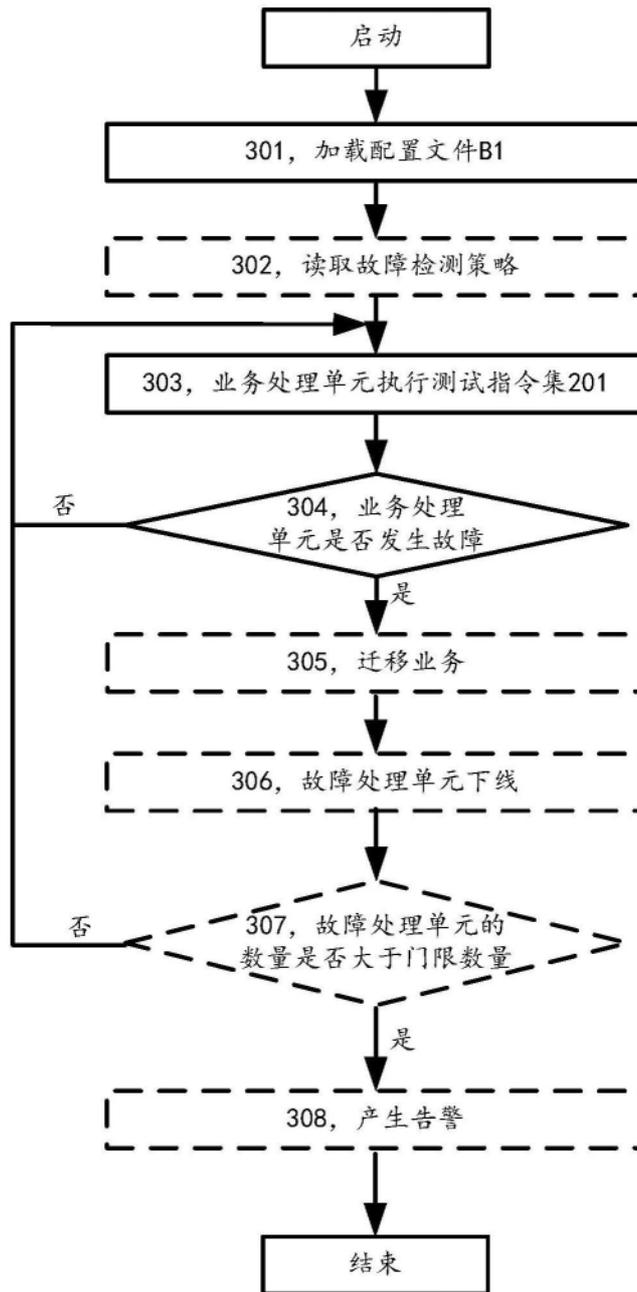


图3

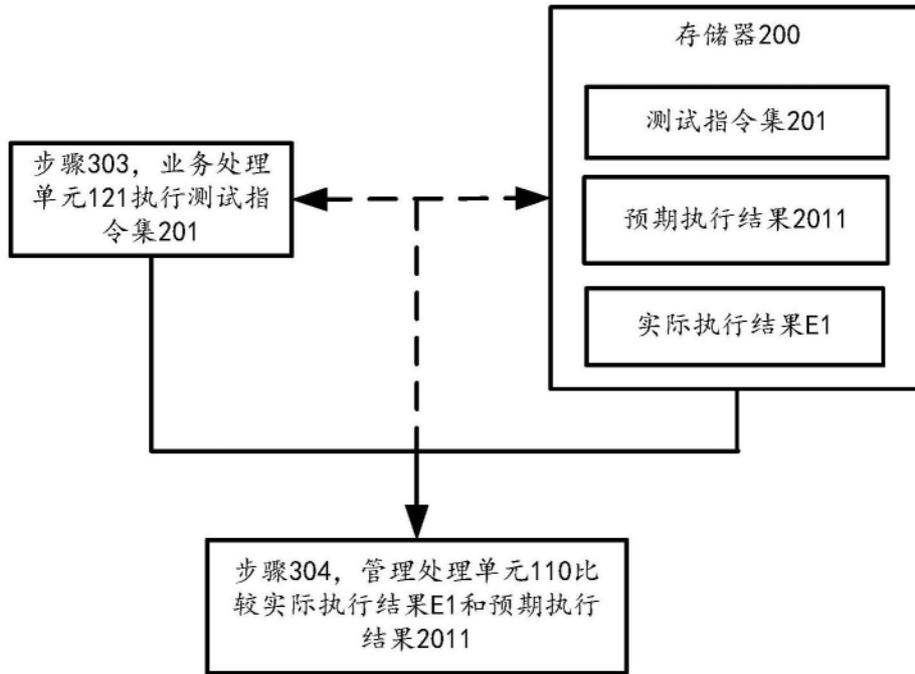


图4A

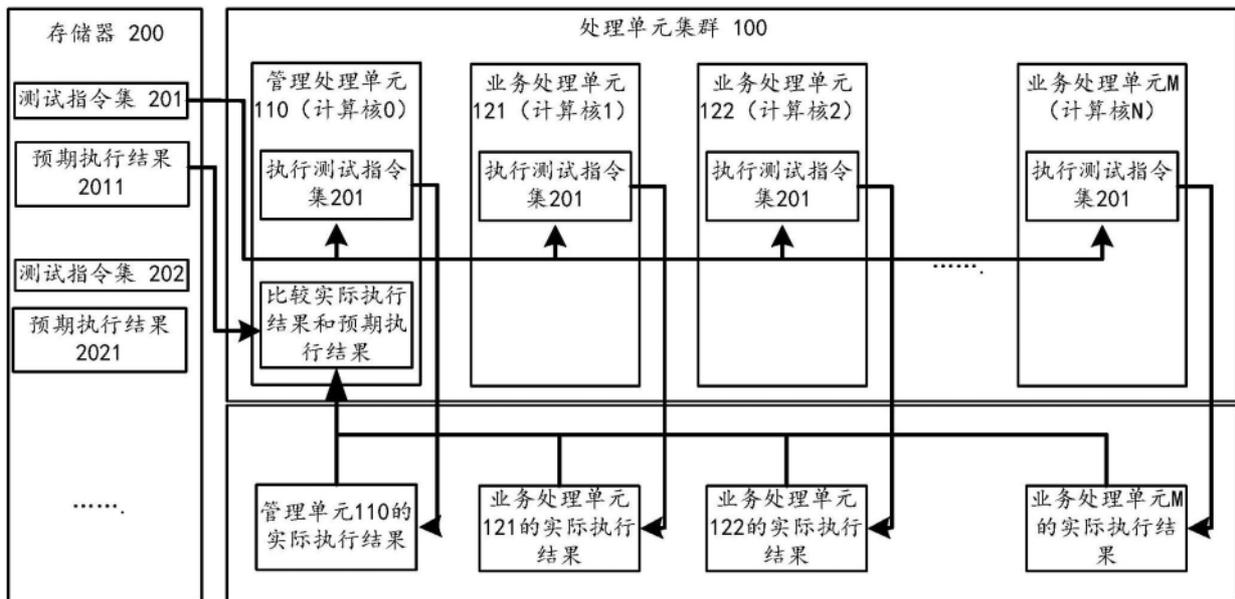


图4B

MD5 ("") = d41d8cd98f00b204e9800998ecf8427e  
MD5 ("a") = 0cc175b9c0f1b6a831c399e269772661  
MD5 ("abc") = 900150983cd24fb0d6963f7d28e17f72  
MD5 ("message digest") = f96b697d7cb7938d525a2f31aaf161d0  
MD5 ("1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890") = 57edf4a22be3c955ac49da2e2107b67a

图5A

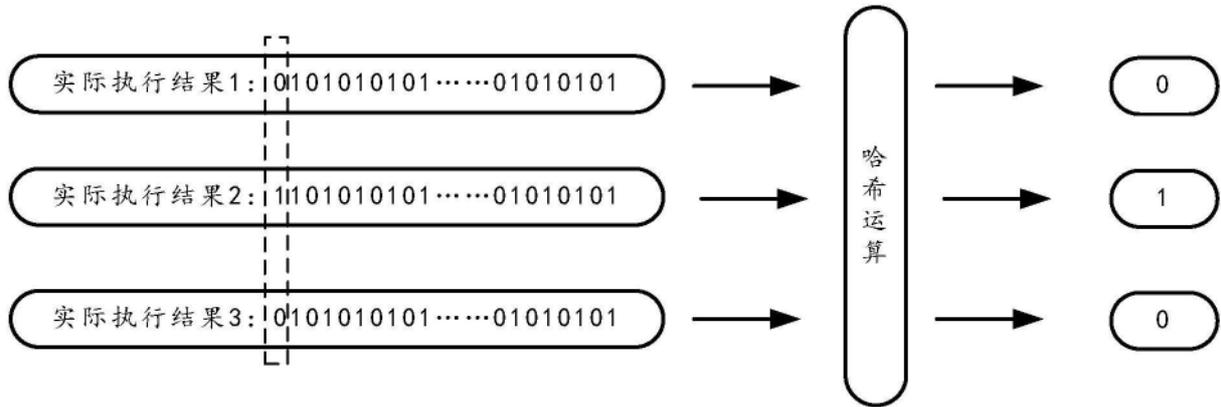


图5B

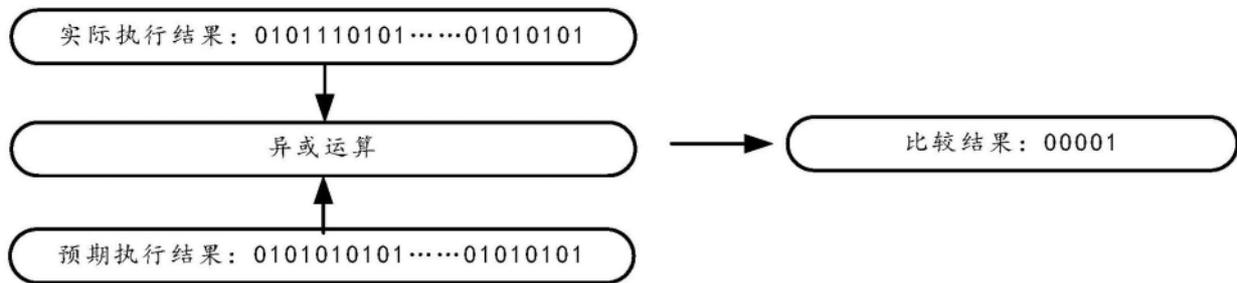


图5C

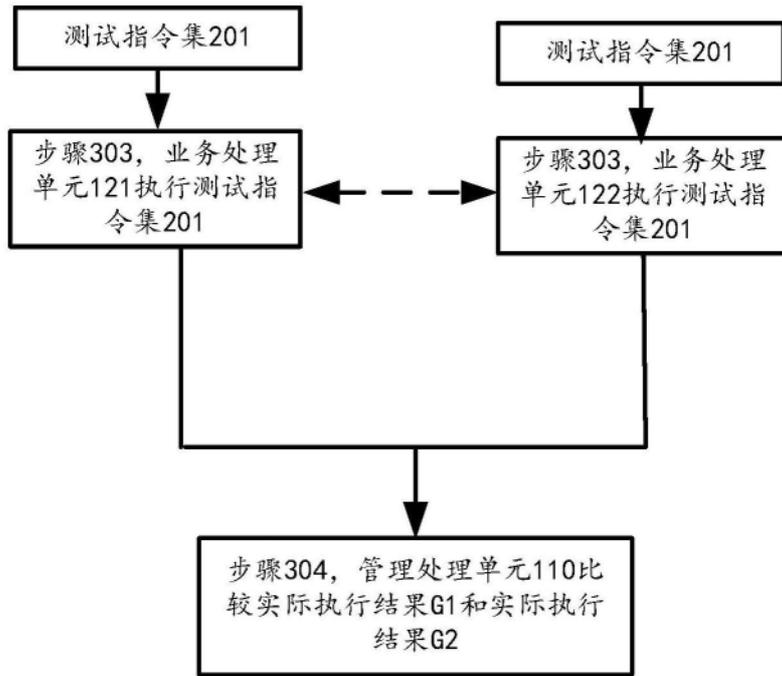


图6A

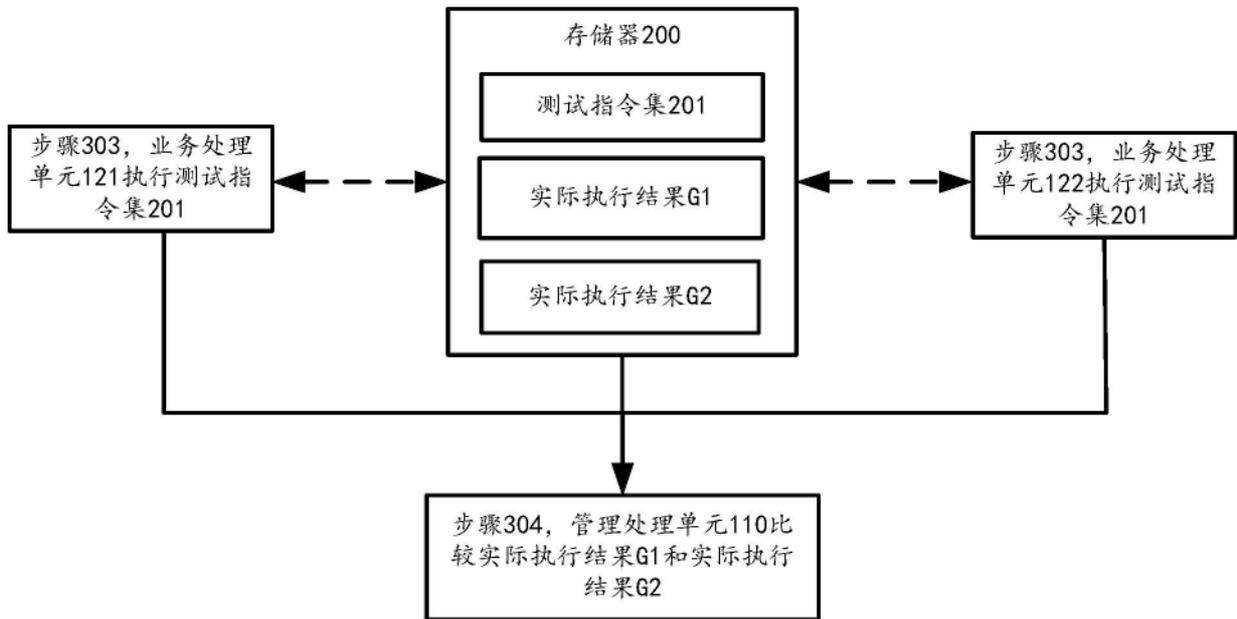


图6B

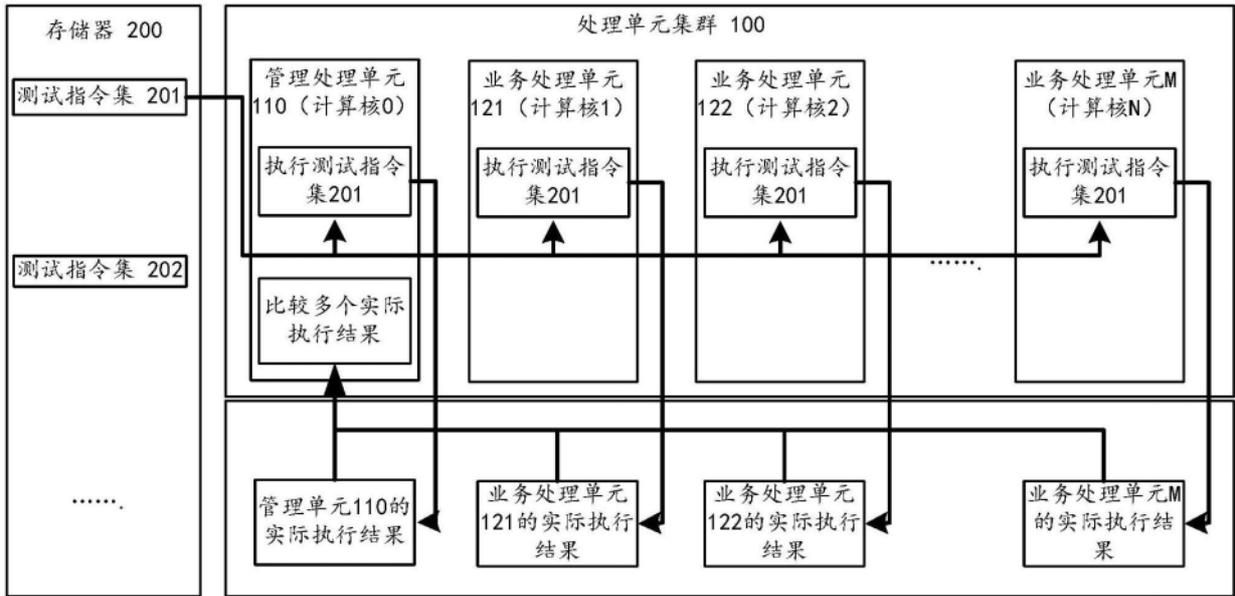


图6C

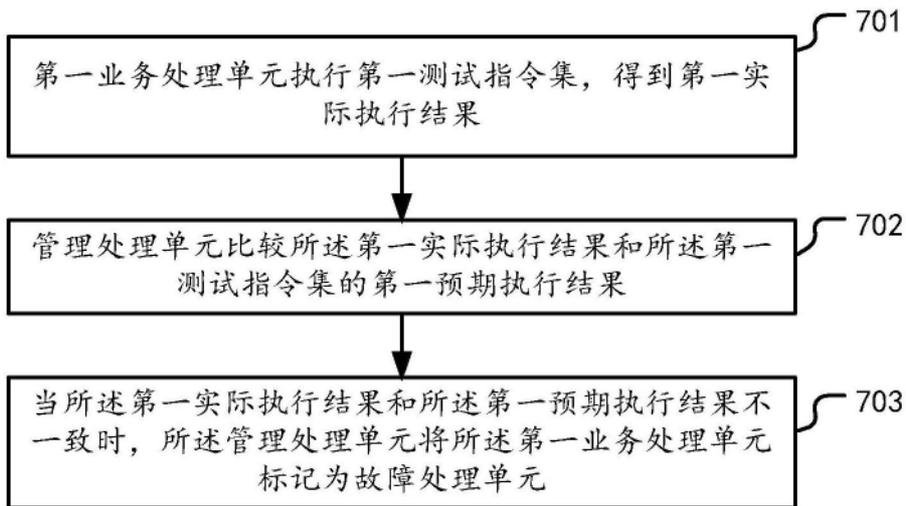


图7

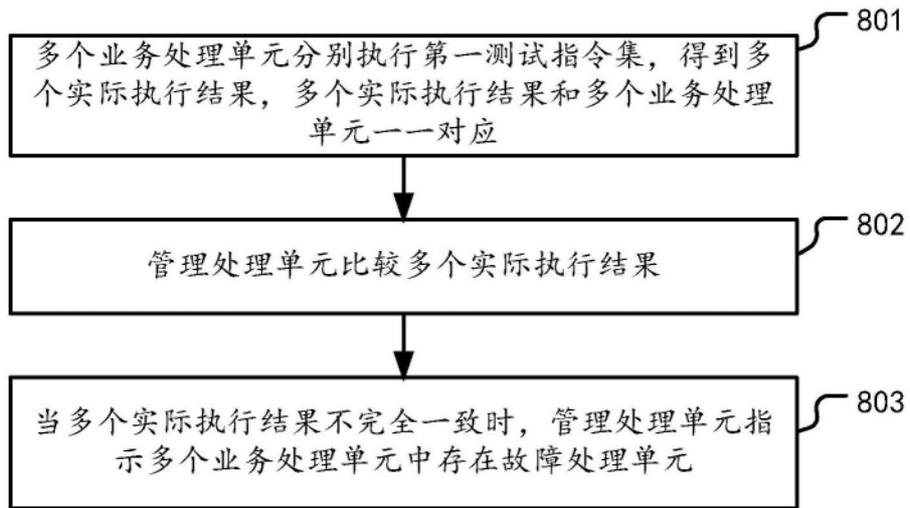


图8