



文章编号:1001-9081(2018)01-0044-06

DOI:10.11772/j.issn.1001-9081.2017071948

面向高性能计算的分布式故障定位框架

高 剑*, 于 康, 卿 鹏, 尉红梅

(江南计算技术研究所, 江苏 无锡 214083)

(*通信作者电子邮箱 gaojian_whu@163.com)

摘要:针对高性能计算系统中故障定位难度高且实时性差的问题,提出了一种基于消息传递的故障定位框架(MPFL),包括基于树形拓扑的故障检测(TFD)和故障分析(TFA)算法。首先,在并行作业初始化时,将所有参与计算的节点进行逻辑上的树形划分,生成故障定位树(FLT),并将故障定位任务分布到节点上;然后,当消息库、操作系统等组件检测到节点异常状态时,基于TFD算法分析作业的FLT结构,根据负载平衡、性能开销等因素选择接收异常状态的节点;最后,节点利用TFA算法对接收到的异常状态进行推理得出故障,TFA算法使用基于规则的事件关联,并基于消息传递设计轻量级的主动探测,将两种方式相结合,提高了故障分析的准确性。实验以模拟节点停机故障为定位目标,并以NPB-FT与NPB-IS为基准测试,在集群上对MPFL框架进行了评估。实验结果表明,MPFL框架在故障定位能力与开销节省方面表现突出。

关键词:高性能计算;消息传递;故障定位;事件关联;主动探测

中图分类号: TP302 **文献标志码:**A

Distributed fault localization framework for high performance computing

GAO Jian*, YU Kang, QING Peng, WEI Hongmei

(Jiangnan Institute of Computing Technology, Wuxi Jiangsu 214083, China)

Abstract: To solve the problem of high difficulty and poor real-time in fault localization for high performance computing system, a Message-Passing based Fault Localization (MPFL) framework was proposed, which included Tree-based Fault Detection (TFD) and Tree-based Fault Analysis (TFA) algorithms. Firstly, when the parallel application was initialized, the Fault Localization Tree (FLT) was obtained by logically dividing all the nodes participating in the computing, and the fault localization tasks were distributed to different nodes. Secondly, if the abnormal state of a node was detected by system components such as message-passing library and operating system, the TFD algorithm was used to analyze the FLT structure, and the node responsible for receiving the abnormal state was selected according to factors such as load balancing, and performance cost. Finally, the fault was derived from the received abnormal state, which was reasoned by the node that used TFA algorithm. The rule-based event correlation and the lightweight active probing based on message-passing were used in TFA algorithm, and the accuracy of fault analysis was improved by combining these two approaches. The experimental evaluation was performed on a typical cluster, which demonstrated the capability of MPFL by locating the shutdown simulation nodes. The experimental results on the NPB-FT and NPB-IS benchmarks show that the MPFL framework has excellent performance on fault localization capability and cost saving.

Key words: high performance computing; message-passing; fault localization; event correlation; active probing

0 引言

高性能计算系统广泛应用于国防建设、科学研究以及国民金融等重要领域,随着系统规模的扩大,系统的平均无故障时间(Mean Time Between Failures, MTBF)逐渐降低,为系统可靠性带来了严峻挑战^[1]。根据可靠性理论的论述,若系统组件的故障(fault)在运行时被激活,将导致系统内部出现错误状态(error),错误状态会在组件间不断地传播,最终引发系统的失效(failure),即系统失效是一个由故障引起错误状态并逐渐积累的渐进过程^[2]。

故障管理是维护高性能计算系统可靠性的重要基础,故障定位作为故障管理的核心功能,发挥着关键作用。故障定

位主要包括检测和分析两个主要步骤:故障检测负责及时发现由故障引起的异常表现,也称作症状(symptom);故障分析负责根据检测到的症状快速、准确地推理得出故障,缩短故障响应时间。高效的故障定位有利于系统在失效前采取相应的处理策略以避免故障的扩散,从而提高系统利用率。

1 相关工作

目前,从计算机科学的不同领域派生出的故障定位方法主要分为事件关联^[3]与主动探测^[4]两类。

1.1 事件关联

事件关联是应用最为广泛的故障定位技术,要求被管设备在自身状态出现异常时,能够向外发出症状告警,由中央管

收稿日期:2017-08-08;修回日期:2017-08-24。 基金项目:国家重点研发计划项目(2016YFB0200502)。

作者简介:高剑(1992—),男,云南富宁人,硕士研究生,主要研究方向:并行计算、运行时系统; 于康(1987—),男,江西景德镇人,助理工程师,博士,主要研究方向:并行计算; 卿鹏(1979—),男,四川资阳人,高级工程师,硕士,主要研究方向:并行编译、运行时系统; 尉红梅(1968—),女,江苏无锡人,高级工程师,博士,主要研究方向:并行计算、并行编译。



理器负责收集并分析被管设备发出的告警事件。文献[5]对事件关联进行了较为全面的综述,包括基于规则、模型以及案例等具体的实现方式。

基于故障传播模型(Fault Propagation Model, FPM)^[6],也称作“症状-故障”模型的事件关联是高性能计算系统中常用的故障定位方式,该方式通过挖掘历史故障经验建立“症状-故障”之间的映射关系。当故障发生时,以系统日志中的监控状态作为“症状-故障”模型的症状输入,并利用不同的分析算法进行推理和调试。这种方式的不足在于:1)定位不及时,由于故障具备传播性,滞后性可能引发更多的故障;2)随着系统规模的不断扩大,事件数量剧增,构建“症状-故障”模型的复杂度大幅提高;3)事件在传播过程中不可避免地出现延迟、丢失等情况,容易造成故障的误报或漏报;4)管理员的干预调试影响系统的正常运行。

1.2 主动探测

主动探测是近年来的研究热点,这种方式基于系统的拓扑结构在运行时主动地执行不同的探针,根据探针的探测结果实现故障的检测和分析。探针是指执行在特定机器也称作探测站上的一类特殊程序,它通过发送命令或请求到系统组件实现端到端的探测,例如ping和traceroute命令可视作检测网络可用性的探针。文献[7]指出探测站与探针集的选择是影响主动探测效率的关键因素;文献[8]对目前常用的探测站和探针集选择算法进行了总结与对比。

由于能够自适应地选择执行的探针集,主动探测与事件关联相比,具有较强的主动性、实时性以及针对性,能够避免症状延迟或丢失对故障定位准确性的影响,但将主动探测直接应用于高性能计算系统的缺陷^[9]在于:1)系统的规模不断增长,所需探针的数目也随之剧增,且探针的设计复杂度高;2)具备强探测能力的探针是有限的,部署探测站的能力也是有限的;3)探测站和探针集的选择已被证明是NP问题,相关选择算法的执行时间随系统规模的增加呈指数级增长;4)大量探针的执行将加剧网络的负载,占用系统宝贵的计算资源。

针对事件关联和主动探测技术应用于高性能计算系统的问题,本文提出了一种基于消息传递的故障定位(Message-Passing based Fault Localization, MPFL)框架,MPFL框架首次将消息库与故障定位问题联系起来,并采用分布式的设计思想,将故障定位任务分配给计算节点,能够在系统运行时实现异常状态的检测,并且将事件关联与主动探测的优势相结合,提高了故障分析的准确性。

2 MPFL 故障定位框架

高性能计算系统的节点通过特定的硬件以及高速网络互连,大部分节点具有同构性且节点状态在执行计算任务时具备相似性。每个节点独立运行,并与其它节点相互通信来协同完成计算任务,节点的通信机制广泛使用消息传递。

因此,MPFL框架的基本思路是充分利用节点间的消息传递在系统运行时获取节点状态。在此基础上,设计基于树形拓扑的故障检测(Tree-based Fault Detection, TFD)和故障分析(Tree-based Fault Analysis, TFA)算法。

2.1 MPFL 软件架构

如图1所示,MPFL框架主要包括故障检测和故障分析两个功能模块,在系统软件架构中与消息库位于同一层次,两者相互协作为上层并行应用程序提供故障定位服务。此外,

作业管理、网络管理、操作系统及文件系统等系统组件为故障定位提供支持,如提供故障信息和触发故障定位等。

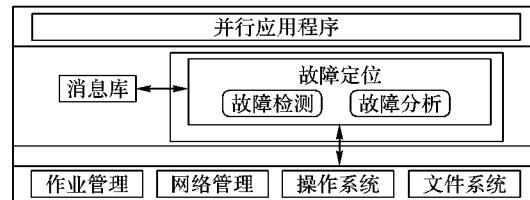


图1 MPFL通用架构
Fig. 1 General architecture of MPFL

MPFL将故障定位任务与消息库紧密联系起来,能够更好地适应高性能计算系统,主要原因包括:

- 1)受作业调度和节点分配策略等因素影响,高性能计算系统的部分故障往往只在特定的运行环境中被激活,故障难以重现和调试;若能在节点运行时获取节点异常状态,有利于解决此类故障。
- 2)参与大规模计算的节点数巨大,且节点之间需要相互通信协作,部分节点的故障更易于在其他节点中体现,此时需要基于消息传递协同多个节点进行综合分析。
- 3)通过与消息库的紧耦合,能够主动地获取并行应用程序运行时的内部状态信息,具备较强的实时性和针对性,能够避免将极大的时间与计算成本消耗在海量系统日志的数据挖掘工作中。
- 4)故障定位可独立于节点计算等工作进程,节点在进行故障检测和分析时不影响系统的正常工作,并且整个过程对用户是透明的,用户只需要关心故障定位结果以采取相应的处理策略。
- 5)作业管理等系统组件虽然能够主动或被动地检测到部分系统异常,但它们之间相互独立,并不共享异常信息。MPFL将不同组件的异常信息进行汇总分析,有利于提高故障定位的准确性。

2.2 基于故障定位树的故障检测

目前,高性能计算系统普遍采用全局的集中式故障管理,随着系统规模扩大,故障概率增加、故障关联性增强,并且故障类型更为复杂多样,这种方式容易陷入性能瓶颈^[10]。

2.2.1 故障定位树

MPFL框架采用层次化和分布式的设计思想,将全局的故障定位任务分配给不同的节点,由各节点运行轻量级的故障定位进程对局部范围内的多个节点进行故障的检测和分析。具体地,当系统在对每个作业进行初始化时,根据相关配置将所有参与计算的节点进行逻辑上的树形划分,逻辑划分得到的树形拓扑称作故障定位树(Fault Localization Tree, FLT)。在系统中并行执行的每个作业都拥有各自的FLT结构,参与多个作业的节点可以属于不同的FLT,但在每个FLT中的位置可能并不相同。

在FLT中,除根节点外,每个节点的父节点是唯一的;除叶节点外,每个节点拥有若干个子节点。MPFL指定由父节点负责所有与子节点故障相关的症状信息收集与分析工作,即父节点是其所有子节点的故障定位节点(Fault Localization Node, FLN)。FLT的结构在作业的生命周期内不会改变,但由于父节点可能失效,同时考虑节点负载、性能开销等因素,每个节点能够根据相关参数及其阈值(如带宽、CPU利用率等)选择替代的故障定位节点(Substitute of Fault Localization



Node, SFLN)。显然,同一子树内的节点状态与存储的症状信息往往具备更强的关联性,失去 FLN 的节点通常选择同一子树的更高层节点作为 SFLN。

在 FLT 中,虽然每个节点的 FLN 是唯一的,但 SFLN 可以有多个。此外,MPFL 指定根节点负责接收不同节点上报的故障定位结果并向用户报告,同时提供接口使得用户能够对故障定位过程进行管理。

故障定位树的优势在于逻辑层次清晰,具备较强的可扩展性,系统可自适应地增加或删减节点;此外,各节点独立工作,能够同时处理多个并发性故障;同时,树形拓扑能够契合绝大多数高性能计算系统的物理拓扑,同一子树的节点可获得较高的通信效率,有利于提高故障定位的整体效率。

2.2.2 故障检测算法——TFD

故障检测的目标是及时地发现由故障引起的症状,而症状的空间性和时间性将直接影响故障分析的准确性。空间性是指收集的症状能否覆盖所有可能的故障,因此需要获取不同硬件部件和不同软件层次的状态,扩大对故障的覆盖范围;时间性指的是症状的收集应当在系统失效之前,并且能够体现节点状态随时间的变化过程。

为满足症状收集的空间性和时间性,结合 FLT 的结构及其工作机制,节点在运行时可利用消息库、网络管理以及操作系统等组件实现对症状的检测。表 1 给出了各组件报告的主要症状模式。

表 1 系统组件报告的主要症状模式

Tab. 1 Symptoms reported by system components

系统组件	症状模式
消息库	消息超时、消息错误
网络管理	端口错误、链路错误
作业管理	节点无心跳
文件系统	I/O 错误、性能降级
操作系统	CPU、内存等节点运行状态异常

算法 1 描述了基于故障定位树的故障检测算法 TFD,作业初始化后,节点正常工作,若系统组件发现可疑节点(Suspected Fault Node, SFN)出现异常症状,计算合适的症状接收节点并发送。

算法 1 TFD 算法。

```

对每个节点;
输入:故障定位树 FLT;
WHILE (Job_Finished! = true) DO
  IF Find_Symp(SFN) THEN
    //检测到故障症状
    IF Available_FLN(SFN) THEN
      //判断可疑节点的 FLN 是否可用
      Send the Symptom to SFN's FLN;
    ELSE
      Select the SFN's SFLN;
      Send the Symptom to SFN's SFLN;
    END IF
  END IF
  IF (Recv_Symp == true) THEN
    Store_Symp(Symptom);
    //将接收到的症状保存到症状集
  END IF
END WHILE

```

TFD 算法在一个作业中的示例如图 2 所示,假设该作业

在一个集群的 9 个计算节点上执行,图中的树形拓扑为逻辑拓扑,与物理连接无关,即为该作业的故障定位树,序号表示了事件发生的顺序:

1) 节点 n 在与节点 m 通信时,发现来自节点 m 的消息错误,节点 n 将此症状报告给节点 m 的 FLN;同时,当节点 y 在给节点 x 发送点对点消息时,发现节点 x 响应超时。

2) 节点 y 试图将节点 x 响应超时的症状报告给其 FLN,但通过可用性探测发现其 FLN 已经过载,不再接收新症状;作业管理同样对节点 x 的 FLN 进行了探测并排除,为简单起见,图中并未标出。

3) 根据系统配置,节点 y 选择根节点作为节点 x 的 SFLN,并将其响应超时症状发送到根节点;同时,作业管理向根节点报告节点 x 无心跳信息。

TFD 算法的优势在于节点只需负责局部范围内的故障,并且能够自适应地选择故障定位节点,有效缓解了单点瓶颈问题;同时,基于消息传递能够获取节点运行时的状态,并且支持并行处理多个症状报告,提高了故障定位的效率。

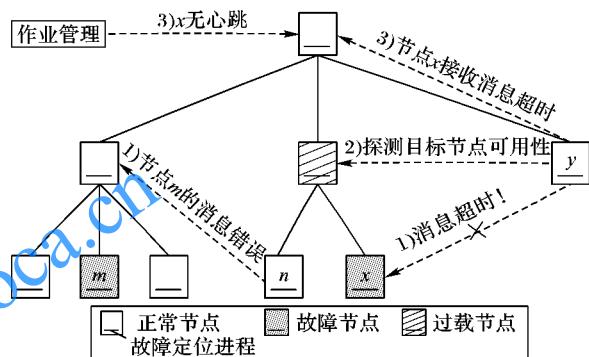


图 2 TFD 算法的示例
Fig. 2 A typical example of TFD algorithm

2.3 故障分析算法的设计

当节点接收到的症状集满足用户设定的触发条件时,例如与某一节点相关的症状数量达到设定的阈值时,节点的故障定位进程将进入分析阶段。

高性能计算系统规模庞大且结构复杂,为故障分析带来了许多挑战:1) 相同的症状可能是由不同组件的故障引起;2) 同一组件的故障也可能引发多种不同的症状;3) 某些故障可能导致其余多个故障的发生,甚至引发事件风暴^[11],即将一个症状的发生称作一个事件,由于故障的关联性,同一时刻出现大量重复、冗余的事件导致系统性能严重下降。

为应对上述挑战,本文将事件关联与主动探测两种方法的优势相结合,在 TFD 算法的基础上,提出基于故障定位树的分析(TFA)算法。图 3 描述了 TFA 算法的结构,首先,故障定位节点使用基于规则的事件关联对 TFD 算法检测到的症状集进行推理,获得多个不同的候选故障集;然后,利用消息探测分别对不同的候选故障集进一步地分析,最终得到若干个不同的故障。

1) 基于规则的事件关联。

基于规则的实现是事件关联应用最为广泛的一种方式,这种方式预先建立规则库,每条规则包含控制逻辑,规则形式为:IF condition A THEN action B,在进行分析时,采用前向链推理机制,选择满足条件的规则并执行相应的动作。

根据文献[12]对高性能计算系统的故障概率、故障位置、时间分布等特征的分析与论述,结合文献[13]对事件压



缩、聚类以及泛化等规则的分类和总结,TFA 算法的事件关联主要包括 3 个步骤:

a) 排重。排除事件集合中大量重复、相似及冗余的事件,有效地减少事件的数量。

b) 组合。将具备关联性的不同事件归并为同一事件组,事件集被划分为互不相交的事件组,充分增强事件语义。

c) 分析。对各事件组分别进行推理,得到相应的候选故障集,每个候选故障集中包含所有可能的事件原因。

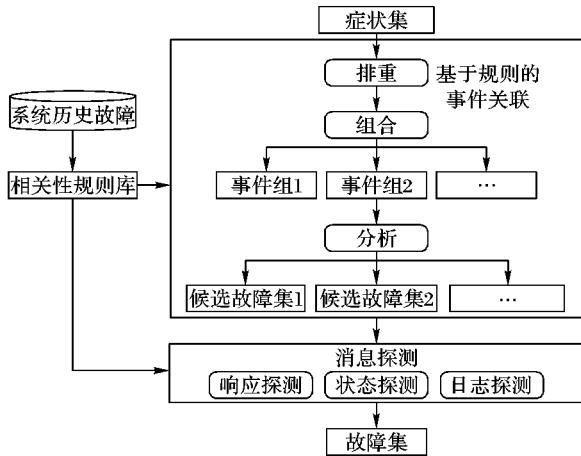


图 3 TFA 算法结构
Fig. 3 Structure of TFA algorithm

2) 消息探测。

TFA 算法中的消息探测借鉴了主动探测的思想,并结合高性能计算系统天然的节点通信优势,可以视作基于消息传递设计实现的轻量级主动探测。

消息探测的主要目标是针对候选故障集主动获取更多的节点状态信息,以对多个故障假设进行筛选,进一步地确定故障。消息探测主要包括三类消息探针:

a) 响应探测。判断与目标节点是否能够正常通信,即探测目标节点的可用性。

b) 状态探测。负责获取目标节点的特定性能指标,如带宽、CPU 利用率等;状态探测常用于节点的 SFLN 选择。

c) 日志探测。要求目标节点返回某个时间范围内的消息日志^[14],消息日志通常包含了消息的类型、标记、时间戳以及完成状态等信息。

在消息探测阶段,故障定位节点需要针对候选故障集使用不同的探测策略,包括不同的消息探针组合及其执行顺序。消息探测的优势在于探针实现简单,每个计算节点都可作为探测站发送探针,并且探针的执行不会增加过大的网络负载,也无需占用计算资源。

无论是基于规则的事件关联中的三个分析步骤,还是消息探测中探针的组合及其执行顺序的选择,都需要得到相关性规则库的支持。相关性规则库通常是指系统历史故障记录中的关联性建立的,与 TFA 算法的分析效率紧密相关,可基于关联规则、频繁序列模式等算法充分挖掘事件记录间的关联性。同时,相关性规则库也为用户提供了接口,支持用户动态部署和更新规则。以节点的停机故障为例,表 2 给出了部分相关规则。

TFA 算法的描述见算法 2。特别地,虽然针对不同的候选故障集,消息探测分别执行相应的探测策略并根据探测结果确定故障,但从不同候选故障集推导出的故障可能相同。

算法 2 TFA 算法。

```

输入: 症状集  $Symp\_Set$ 。
输出: 故障集  $Fault\_Set$ 。
Initialization:  $Fault\_Set = \emptyset$ ;
 $Purify\_Set = Event\_Purify(Symp\_Set)$ ; // 排重
 $Grouping\_Set = Event\_Grouping(Purify\_Set)$ ; // 组合
FOR  $i = 0$  to  $|Grouping\_Set|$  DO
  // 对每个事件组进行分析, 得到相应候选故障集
   $Candidate\_Faults_i = Event\_Reasoning(Group_i)$ ;
  Add  $Candidate\_Faults_i$  to  $Candidate\_Fault\_Sets$ ;
END FOR
FOR  $i = 0$  to  $|Candidate\_Fault\_Sets|$  DO
  // 对每个候选故障集进行消息探测
  FOR  $j = 0$  to  $|Candidate\_Faults_i|$  DO
    // 对候选故障集中的多个故障假设进行分析
    IF  $Msg\_Probing(Candidate\_Faults_{ij})$  THEN
      Add  $Candidate\_Faults_{ij}$  to  $Fault\_Set$ ;
      break;
    END IF
  END FOR
END FOR
Return( $Fault\_Set$ );
  
```

TFA 算法有效缓解了事件关联的效率下降问题;同时,消息探测在高性能计算系统中适应性强,主动、及时且有针对性地获取节点的运行时状态,能够避免症状延迟或丢失对故障定位准确度的影响。

表 2 与节点停机故障相关的规则示例

Tab. 2 Some rules related to node shutdown fault

分析阶段	规则形式
排重	IF 节点 x 报告节点 y 消息超时的次数 $> N$ THEN 仅保留一次
组合	IF M 个不同症状出现在同一时间窗口 T THEN 归并为同一事件组
分析	IF S 个节点报告 y 超时 and 节点 y 无心跳 THEN 节点 y 停机 or 链路及端口故障
消息探测	IF Q 个响应探测均返回节点 y 无响应 THEN 节点 y 停机

3 实验评价

本章通过模拟实验对 MPFL 框架的故障定位能力及其对应用程序的性能影响进行评价。

本实验的平台是一个具有 10 个计算节点的典型集群,每个节点拥有 64 GB 内存,2 个 8 核心 Intel Xeon 处理器,节点使用 InfiniBand 高速网络互连;操作系统为 Red Hat Enterprise Linux Server release 6.3;消息库的版本为 MVAPICH2-2.2。

3.1 功能评价

高性能计算系统的故障模式主要有通信网络故障、计算节点故障及存储节点故障^[15],其中计算节点的停机故障是影响并行程序运行稳定性,甚至引发系统失效的主要原因^[16],因此,本实验以定位节点的停机故障为目标,以证明 MPFL 框架的故障定位能力。

为模拟节点的停机故障,本文对消息传递接口(Message Passing Interface, MPI)的典型开源实现,即 MVAPICH 源码进行简单的修改:在作业初始化时,所有进程从配置文件获取模拟节点停机故障的进程号,并且进程在执行消息发送操作前需满足要求:

1) 若本进程模拟故障,不执行任何动作,直接结束。



2)若目标进程模拟故障,对于探测消息,默认本次探测超时并结束发送操作;否则不执行任何动作,结束操作。

3)本进程与目标进程均正常,正常发送消息。

这些要求保证了模拟故障进程不发送任何消息,也不可能接收到消息。此外,节点无心跳、带宽下降、CPU利用率过高等多个症状在程序运行时被注入,以模拟其余系统组件的行为,从而达到模拟节点停机故障的效果。

通常,一个节点的停机故障将导致运行在该节点上的所有程序都终止,因此运行一个测试程序且仅有一个进程模拟节点停机故障即可。同时,为满足进程间的通信量需求,测试程序需进行多次全交换(all-to-all)通信,并且基于多线程实现症状信息的收集与分析。

假设程序的进程规模为 P ,实验分为 N 组,每组实验的进程规模不同,且重复测试 M 次,每次实验随机选择1个进程 P_f 模拟节点停机故障。本实验的参数如表3所示。

表3 功能评价的实验参数

Tab. 3 Specific parameters of functional evaluation

参数	含义	取值
N	实验组的数量	4
M	每组实验重复的次数	100
P	测试程序的进程规模	16, 32, 64, 128
P_f	模拟节点停机故障的进程	randomly

实验结果表明,进程 P_f 总是在几秒的时间内被找到,与通常情况相比,不再需要提供冗长的运行状态上下文给用户进行人工分析,用户只需要关心故障定位的结果,这极大地减轻了用户的负担,因此,可以认为MPFL框架是有效的。

实际上,故障定位的准确性与相关性规则紧密相关,但规则库往往无法覆盖所有的系统异常,故障定位不可能做到100%正确。例如,表2的组合规则将时间窗口 T 内的事件划分为一个事件组,若 T 值过小,由于消息存在延迟性,某些症状可能在 T 之外到达,导致故障信息不足;若 T 值过大,事件之间的关联性将降低,影响故障分析的准确性。

3.2 性能评价

本节使用美国航空航天局在NAS(Numerical Aerodynamic Simulation)项目中开发的面向高性能计算的并行基准测试集(NAS Parallel Benchmark, NPB)^[17]中的两个核心程序:快速傅里叶变换(NPB Fast Fourier Transformation, NPB-FT)与整数排序(NPB Integer Sort, NPB-IS)对MPFL的性能进行测试:

1)NPB-FT利用快速傅里叶变换来解决三维的偏微分方程,其初始阶段包含大量的迭代,每次迭代包含大量的all-to-all通信。

2)NPB-IS用于求解基于桶排序的二维大整数排序,同样包含大量的all-to-all通信。

实验分别对部署MPFL前后的NPB-FT、NPB-IS计算性能进行比较。与3.1节类似,为了模拟各系统组件发布症状的行为,当测试部署MPFL的NPB-FT和NPB-IS时,在运行时周期性地注入不同的症状。不同的是,本实验注入的症状不会使得TFA算法推导出故障。

此外,NPB-FT要求进程规模为2的幂次,本实验将沿用表3所示的参数设置, P_f 除外,即无需模拟故障。

实验结果如图4~5所示,本文进行了4组对比实验,测试程序规模(CLASS)选择A规模,进程规模依次为16,32,64以及128。可以看出,MPFL部署前后的NPB-FT、NPB-IS计算

性能无明显差距,在图中用程序的每秒百万次浮点运算(Million Floating-point Operations per Second, MFLOPS)进行表示。究其原因,这是由于在正常程序的运行过程中不会触发完整的故障分析过程。同时,测试程序的性能减速没有随着进程规模的持续增加而呈上升趋势。经过统计分析,部署MPFL仅仅分别给NPB-FT和NPB-IS带来了5.68%和2.12%的运行开销,这说明MPFL对系统的性能影响较小,具备较强的可扩展性。

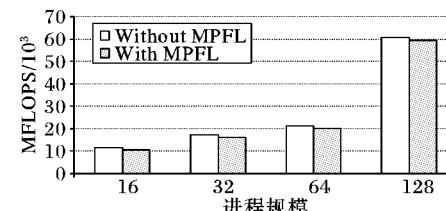


Fig. 4 Results of NPB-FT benchmark (CLASS = A)

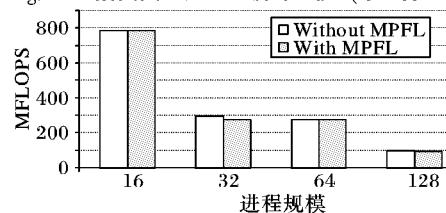


Fig. 5 Results of NPB-IS benchmark (CLASS = A)

4 结语

故障定位作为故障管理的核心,对提高系统可靠性有重要意义,但当前的故障定位技术难以有效地直接应用于高性能计算系统。本文首次将故障定位问题与消息库紧密联系起来,且考虑充分利用各系统组件独立获取的异常信息,提出一种基于消息传递的故障定位框架MPFL,并设计实现基于故障定位树的故障检测与分析算法,能够为高性能计算系统提供实时的轻量级分布式故障定位服务。本文通过定位模拟的节点停机故障对MPFL的功能进行了原型实验证,并分别利用NPB-FT与NPB-IS基准测试程序对MPFL进行了性能评价。实验结果表明,MPFL框架是有效的,并且具备较强的可扩展性。

下一步的工作重点包括:1)对历史故障经验进行更深度的挖掘以提高故障定位的准确性;2)除消息库外,开发网络管理、操作系统等系统组件对MPFL框架的支持;3)针对MPFL框架的故障定位准确率、性能开销、可扩展性等方面进行更全面的评价。

参考文献(References)

- [1] ZHENG Z, LI Y, LAN Z. Anomaly localization in large-scale clusters [C]// Proceedings of the 2007 IEEE International Conference on Cluster Computing. Piscataway, NJ: IEEE, 2007: 322–330.
- [2] AVIZIENIS A, LAPRIE J C, RANDELL B. Fundamental concepts of dependability [R]. Newcastle: LAAS-CNRS, 2001: 4.
- [3] JORDaan J F, PATEROK M. Event correlation in heterogeneous networks using the OSI management framework [C]// Proceedings of the IFIP TC6/WG6.6 Third International Symposium on Integrated Network Management with Participation of the IEEE Communications Society CNOM and with Support from the Institute for Educational Services. Amsterdam: North-Holland Publishing Co., 1993: 683–695.



- [4] NATU M, SETHI A S. Active probing approach for fault localization in computer networks [C]// Proceedings of the 2006 4th IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services. Piscataway, NJ: IEEE, 2006: 25 – 33.
- [5] LGORZATA STEINDER M, SETHI A S. A survey of fault localization techniques in computer networks [J]. Science of Computer Programming, 2004, 53(2): 165 – 194.
- [6] KATKER S, PATEROK M. Fault isolation and event correlation for integrated fault management [C]// Proceedings of the 5th IFIP/IEEE International Symposium on Integrated Network Management. Berlin: Springer, 1997: 583 – 596.
- [7] CHENG L, QIU X, MENG L, et al. Efficient active probing for fault diagnosis in large scale and noisy networks [C]// Proceedings of the 29th IEEE International Conference on Computer Communications. Washington, DC: IEEE Computer Society, 2010: 1 – 9.
- [8] PATIL B M, PATHAK V K. Survey of probe set and probe station selection algorithms for fault detection and localization in computer networks [J]. IEEE Transactions on Networks and Communications, 2015, 3(4): 57.
- [9] 孟洛明, 黄婷, 成璐, 等. 支持多故障定位的探测站点部署方法 [J]. 北京邮电大学学报, 2009, 32(5): 1 – 5. (MENG L M, HUANG T, CHENG L, et al. Probe station placement for multiple faults localization [J]. Journal of Beijing University of Posts and Telecommunications, 2009, 32(5): 1 – 5.)
- [10] HUKERIKAR S, DINIZ P C, LUCAS R F, et al. Opportunistic application-level fault detection through adaptive redundant multithreading [C]// Proceedings of the 2014 International Conference on High Performance Computing & Simulation. Piscataway, NJ: IEEE, 2014: 243 – 250.
- [11] GARDNER R D, HARLE D A. Network fault detection: a simplified approach to alarm correlation [C]// Proceedings of the 16th IEEE Global Telecommunications Conference. Washington, DC: IEEE Computer Society, 1997: 44 – 51.
- [12] SCHROEDER B, GIBSON G. A large-scale study of failures in high-performance computing systems [J]. IEEE Transactions on Dependable and Secure Computing, 2010, 7(4): 337 – 350.
- [13] JAKOBSON G, WEISSMAN M. Real-time telecommunication network management: extending event correlation with temporal constraints [C]// Proceedings of the Fourth International Symposium on Integrated Network Management IV. London: Chapman & Hall, 1995: 290 – 301.
- [14] LEMARINIER P, BOUTEILLER A, KRAWEZIK G, et al. Coordinated checkpoint versus message log for fault tolerant MPI [J]. International Journal of High Performance Computing and Networking, 2004, 2(2/3/4): 146 – 155.
- [15] SCHROEDER B, GIBSON G A. Understanding failures in petascale computers [C]// Proceedings of the 6th Scientific Discovery through Advanced Computing Conference. Bristol: IOP Publishing Ltd, 2007: 2022 – 2032.
- [16] 武林平, 孟丹, 梁毅, 等. LUNF——基于节点失效特征的机群作业调度策略 [J]. 计算机研究与发展, 2005, 42(6): 1000 – 1005. (WU L P, MENG D, LIANG Y, et al. LUNF—a cluster job schedule strategy using characterization of nodes' failure [J]. Journal of Computer Research and Development, 2005, 42(6): 1000 – 1005.)
- [17] BAILTY D, HARRIS T, SAPHIR W, et al. The NAS parallel benchmarks 2.0: NAS-95-020 [R]. Washington: NASA Ames Research Center, 1995: 12.

This work is partially supported by the National Key Research and Development Program of China (2016YFB0200502).

GAO Jian, born in 1992, M. S. candidate. His research interests include parallel computing, runtime system.

YU Kang, born in 1987, Ph. D., assistant engineer. His research interests include parallel computing.

QING Peng, born in 1979, M. S., senior engineer. His research interests include parallel compilation, runtime system.

WEI Hongmei, born in 1968, Ph. D., senior engineer. Her research interests include parallel computing, parallel compilation.

(上接第30页)

- [8] ASWATHY M C, TRIPTI C. A cluster based enhancement to AODV for inter-vehicular communication in VANET [J]. International Journal of Grid Computing & Applications, 2012, 3(3): 41 – 50.
- [9] THOMAS N D, GRAY K. SDN: Software Defined Networks [M]. Sebastopol: O'Reilly Media, 2014, 8(7): 31 – 38.
- [10] KU I, LU Y, GERLA M, et al. Towards software-defined VANET: architecture and services [C]// Proceedings of the 2014 13th Annual Mediterranean Ad Hoc Networking Workshop. Piscataway, NJ: IEEE, 2014: 103 – 110.
- [11] SHIN M K, NAM K H, KIM H J. Software-Defined Networking (SDN): a reference architecture and open APIs [C]// ICTC 2012: Proceedings of the 2012 International Conference on ICT Convergence. Piscataway, NJ: IEEE, 2012: 360 – 361.
- [12] LI H, DONG M, OTA K. Control plane optimization in software-defined vehicular Ad Hoc networks [J]. IEEE Transactions on Vehicular Technology, 2016, 65(10): 7895 – 7904.
- [13] TRUONG N B, LEE G M, GHAMRI-DOUDANE Y. Software defined networking-based vehicular Ad Hoc network with fog computing [C]// Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management. Piscataway, NJ: IEEE, 2015: 1202 – 1207.

- [14] 崔翠梅, 杨德智, 姜程鑫. 基于 NS3 的移动认知网络仿真系统 [J]. 通信技术, 2016, 49(11): 1509 – 1513. (CUI C M, YANG D Z, JIANG C X. NS3-based Simulation System for Mobile Cognitive Radio Networks [J]. Communications Technology, 2016, 49(11): 1509 – 1513.)

This work is partially supported by the National Natural Science Foundation of China (6137915), the Science and Technology Planning Project of Guangdong Province (2015B010111001).

DONG Baihong, born in 1993, M. S. candidate. His research interests include vehicular Ad Hoc network.

DENG Jian, born in 1993, M. S. candidate. His research interests include vehicular Ad Hoc network, named data network.

ZHANG Dingjie, born in 1992, M. S. His research interests include vehicular Ad Hoc network.

WU Weigang, born in 1976, Ph. D., professor. His research interest include vehicular Ad Hoc network, cloud computing.