第33卷 第18期 计 算 机 工 程 2007年9月
Vol.33 No.18 Computer Engineering September 2007
·工程应用技术与实现· 文章编号：1000—3428(2007)18—0262—03 文献标识码：A 中图分类号：TP338

分布、实时、容错一体化设计方法研究

李新明[1]，李 艺[1]，王 鹏[1]，刘 东[2]
(1. 装备指挥技术学院国防重点实验室，北京 101416；2. 国防科技大学，长沙 410073)

摘 要：针对分布式航天器系统的运行环境和特点，对嵌入式系统在空间环境、实时、容错、分布上的需求进行了分析，提出了分布、实时、容错一体化的嵌入式系统设计方法，从满足实时要求下的实时容错能力、免疫与自愈相结合的综合容错能力、单节点的容错与节点间容错相结合的分布容错能力和多种容错方法集成等4个方面，对设计方法进行了阐述。

关键词：分布式航天器系统；嵌入式系统；分布；实时；容错

Integrative Design Method of Distribution, Real-time and Fault Tolerance

LI Xin-ming[1], LI Yi[1], WANG Peng[1], LIU Dong[2]
(1. Key Lab of National Defense, Academy of Equipment Command and Technology, Beijing 101416;
2. National University of Defense Technology, Changsha 410073)

【Abstract】 According to the running environment and characteristics of distributed spacecraft system, requirements of embedded systems are analyzed on space environments, real-time, fault tolerance and distribution. An integrative design of distribution, real-time and fault tolerance is proposed. Details of the design are described from four aspects: real-time fault tolerant ability based on accomplishing the real-time requirements, integrated fault tolerant ability integrated immunity and self cure, distributed fault tolerant ability integrated fault tolerance on single node and among multi nodes and the integration of multi fault tolerant techniques.

【Key words】 distributed spacecraft system; embedded system; distribution; real-time; fault tolerance

近十年来，强大的需求成为嵌入式系统发展的巨大动力，而网络、芯片等技术的发展加快了嵌入式系统的发展速度，使得嵌入式系统成为无处不在的计算单元，在航空、航天、军事、通信等各个领域得到了广泛的应用。

分布式微小卫星和编队卫星技术是20世纪90年代以来航天领域的重要研究内容之一，由多个微小卫星编组组成一个分布式航天器系统，各个微小卫星在自主运行的基础上，按照一定规则编队飞行，协同工作，完成预定的任务。与单颗大卫星相比，分布式航天器系统具有成本低、质量轻、体积小、研制周期短、技术含量高、使用灵活等许多优点，而且系统的整体功能远远大于单个卫星功能，能够完成某些单个卫星无法胜任的任务，分布式航天器系统代表了未来航天器技术的发展方向。

本文以分布式航天器系统为主要应用背景，同时也参考车载、舰载等综合电子信息系统，研究满足分布式航天器系统需求的嵌入式系统的设计方法和技术。

1 需求分析

针对分布式航天器系统的运行环境和各种需求，运行于分布式航天器上的嵌入式系统应满足以下的需求：

(1)运行在复杂的空间环境或恶劣环境中。空间环境对卫星上的电子系统、光电系统以及机电系统的设计都有很大影响。空间环境可以概括为：强辐射，粒子碰撞，强电磁场环境，大范围温度变化，微重力和真空环境等。空间强辐射主要来源于宇宙射线，对电子系统产生总辐射剂量和单事件效应(single-event effects，SEE)，后者对空间电子设备的危害性很大，可以导致翻转(single-event upset，SEU)、栓锁(single-event latchup，SEL)和烧毁(single-event burnout，SEB)等。这些复杂的空间环境或恶劣环境可能导致硬件故障、程序跑飞、应用程序数据错误、内部操作系统数据错误以及其他不可预料的后果，在进行嵌入式系统的设计时必须充分考虑复杂空间环境和恶劣环境的影响。

(2)强实时处理能力。分布式航天器系统中的控制和处理对实时有很高的要求，如果超过了预定的死线(deadline)，将会产生严重后果。战场环境瞬息万变，信息量大，通过无线传感器网络获得的信息必须进行实时处理和快速融合，否则就会变成失效的信息，影响作战决策的准确性。嵌入式软件系统的设计需要在时间管理、实时调度、实时通信和实时事件驱动等方面满足实时性的要求，首先，强实时任务必须在预定的生死线之前完成；其次，对系统的各种运行状态、时间甚至路径等特性必须是可预测的，即使在最坏的情况下，也能有效进行处理。

(3)强容错能力。分布式航天器系统是一个自主运行的系统，各节点在不依赖于地面设施的情况下，能够自主确定星座状态和维持星座构型，完成飞行任务。这种自主性要求嵌入式系统能够保证分布式航天器系统在出现故障的情况下，不需要地面的人工干预，通过自身的容错功能，按时准确地完成系统预先制定的飞行任务。嵌入式系统软件是分布式航天器系统的重要组成部分，对系统的自主运行发挥了关键的

基金项目：国家“973”计划基金资助项目(51312020－3)
作者简介：李新明(1965－)，男，教授，主研方向：系统软件，网络系统；李 艺，教授；王 鹏，硕士、助教；刘 东，博士研究生
收稿日期：2006-09-26 **E-mail：**liyi1221@163.net

作用。因此，嵌入式系统软件在保证航天器系统能够完成任务，对航天器系统进行容错的同时，其本身也应该是一个强免疫系统，对在空间环境下遇到的高辐射、强电磁等情况有充分的考虑，以保证自身在空间的运行，即应该具有很强的自愈能力，能够容忍其本身所出现的错误。复杂的空间环境或恶劣环境，使得系统出现故障的可能性大大增加，必须及时发现并处理故障。并且，分布式航天器系统上的故障类型、处理方式和普通的地面系统有很多不同，要提供有效的 SEU 故障和 SEL 故障的恢复处理机制。

(4)体现分布式结构特点。分布式航天器系统由多个微小卫星构成，每个节点都不是孤立的，而是紧密联系、互相关联的，各个节点按照既定的拓扑结构相互连接，具有物理上分布的特点。在地面控制中心的整体控制下，内部系统是一个互连、互通的自治系统，每个节点具有自主性，所以设计的嵌入式系统必须体现分布式结构的特点。同时这种紧耦合形式的分布也可以充分利用，对完成容错、实时等需求提供平台。

2 分布、实时、容错一体化设计方法

通常，分布、实时和容错作为系统不同的需求，被分别处理，有各自不相关的解决方案，同时由于各种方案相互冲突，很难取得一个满足所有需求的整体解决办法。针对分布式航天器系统的特点，本文提出了分布、实时、容错一体化设计的思想，把它们作为一个整体需求，进行统一规划、综合考虑，权衡处理，满足系统在实时要求下的容错能力，而分布式环境为实时容错提供了很好的平台。

2.1 满足实时要求的实时容错能力

实时容错是指把容错技术应用于实时系统中，使得系统在出现故障进行容错处理后，仍能保证计算正确且满足实时性要求。故障的检测与冗余容错方法将不可避免地带来空间上或时间上的额外开销，可能会使实时任务的执行产生延迟，从而对系统性能和实时性产生较大的影响。由于故障产生的随机性以及节点之间实时任务迁移的时间延迟的不确定性等因素，可能对实时系统运行的可预测性产生影响，因此，满足实时要求下的容错能力的设计需要考虑以下原则：(1)在系统正常运行时，对硬件和软件故障的监测任务必须与实时任务等统一进行调度；(2)故障监测算法、实时任务调度算法以及其他故障处理算法必须是实时的；(3)为支持容错处理所消耗的系统资源，如存储空间等必须满足实时系统余量的要求。这样，由于单节点上运行的都是实时操作系统，因此保证在单节点上运行的任务和任务出错时重新运行的任务能够按时完成。对整个分布式系统来说，当节点失效时，通过每个节点上负载的评价，从负载平衡的角度，把失效节点上的任务迁移到负载最小的节点上去，在保证任务的按时完成的前提下，使整个系统资源利用率最优。对节点负载的评价，除了考虑吞吐量、运行能力等因素，还必须考虑实时要求，如节点上实时任务的数量、任务迁移后系统资源的余量控制等。任务迁移也要根据实时性，按急缓程度，有序进行。

2.2 免疫与自愈相结合的综合容错能力

参照生物学概念，在容错设计中引入了“免疫”和“自愈”的概念，系统要具有免疫与自愈相结合的综合容错能力。免疫是指系统具有防范某些故障发生的能力。自愈是指系统具有监测、发现、容忍、处理故障和恢复正常的能力。在分布式航天器系统的容错处理中，首先，系统运行在复杂、恶劣的空间环境，必须提高系统的免疫能力，尽量保证某些错误或者恶劣环境不会导致系统产生故障。然后提高系统的自愈能力，一旦发生故障，系统在不需要外部干预的前提下，能及时监测到故障的产生，并采取相应的处理措施，使系统恢复正常。

例如，空间强辐射会导致发生门击、烧毁或锁定等单粒子栓锁 SEL 现象。统计表明，SEL 虽然发生的次数很少，但却是不可避免的，所以，设计的系统必须具备对 SEL 的容错能力。在分布式航天器系统的硬件平台中设计抗 SEL 过流检测和电路保护，通过在大电流发生后重启电源来消除 SEL，针对嵌入式系统的硬件平台中各子板和关键芯片的供电问题，都要采用相应的抗 SEL 防护电路，使得整个卫星系统对 SEL 有免疫和自愈能力，使得 SEL 不会导致严重的系统故障。

航天器系统的硬件平台采用基于温备的系统级双机容错方案。单节点中有 2 台工作机组，对于每台机组，均在扩展板上设有外部 watchdog 电路用于监控机组的运行。一旦某一台机组发生故障，经过一定的时间后，故障机组相应的外部 watchdog 将会检测到故障的出现，并重新复位故障机组，同时通知另一台机组接管工作。

2.3 单节点的容错与节点间容错相结合的分布容错能力

分布式航天器系统的分布环境为系统的容错处理提供了很好的平台。单节点的容错是分布式航天器系统容错的基础，此外，利用分布式航天器系统的多节点分布特点，通过对其他节点状态、节点通信状态、节点上运行飞行任务的状态进行实时监控，可以及时发现并处理错误，从而极大地提高系统的整体容错能力。

在分布式航天器系统的软件平台上，设计分布容错处理软件层，用于完成节点间的容错处理。在分布式航天器系统中，没有中心控制节点，所以采用对称分布式模型、全局信息收集方式来处理各节点间的容错。系统的每个节点都对其他节点的运行状况进行实时记录，从而对节点间的通信状态、节点的状态、节点上运行任务的状态等进行准确的判断，进行正确的容错处理。由于没有中心控制节点，因此分布容错处理软件在系统每个节点上运行的程序都是相同的，采用多线程的结构，每个节点上都运行如下线程：注册及异常任务信息更新线程，节点状态应答线程，系统信息更新线程，系统任务信息查询线程，系统信息查询及恢复线程，系统信息显示线程，系统通信故障检测线程，远程加载与系统更新线程等，这些线程分成两种类型：一种是在指定的端口上等待消息的线程；另一种是定时唤醒的线程。

当进行远程加载和升级处理时，通过分布容错处理软件层，可以有效地保证加载节点上软件的版本一致性。分布式航天器系统的每个节点，虽然有效载荷不同，但软件平台是相同的，当对操作系统核心模块、分布容错处理程序、系统函数库、系统程序、应用程序等进行加载和升级时，必须保证所有节点上相同的软件具有相同的版本。这种版本一致性的控制，不是通过地面站对不同的节点同时发送多条命令来实现的，而是由分布容错软件层自动处理完成的。

2.4 多种容错方法的集成

针对不同的错误，要采取不同的容错处理方法，如硬件处理方法、硬件和软件相配合的处理方法、软件处理方法等。各种不同的容错处理方法，有效地集成在一起，提高系统整体容错能力。

如对存储器的冗余处理采用硬件容错方法。为了消除单事件翻转 SEU 使存储器出现错误，提高硬件系统存储器模块

对空间环境的抗辐照能力，通过基于三选二逻辑的硬件冗余方法来处理。

(1)对硬件平台的系统监控采用硬件与软件相结合的容错方法。在硬件平台中，采用内部和外部两级 watchdog 监控策略。内部 watchdog 监控利用 AT91RM9200 处理器内部软件 watchdog 功能实现，外部 watchdog 监控利用接口板中的 FPGA 和专用 watchdog 芯片实现。其中，专用 watchdog 芯片是外部 watchdog 监控功能的主体，当专用 watchdog 芯片损坏时，则采用位于 FPGA 内部的利用硬件描述语言实现的 watchdog 接管外部 watchdog 监控功能。内部软件 watchdog 的优先级低于接口板上硬件 watchdog 的优先级。当接口板上 watchdog 失效，从而不能够达到重启 CPU 板的目的时，AT91RM9200 内部 watchdog 将会发生作用，重启 CPU。这是通过控制两种 watchdog 计数时间的长短来实现的。由此构成了硬件平台中的二级监控、一级冗余的 watchdog 监控机制。

(2)对飞行控制程序的监控采用软件容错方法。飞行程序作为分布式航天器系统软件平台上的用户程序运行，对其状态的监控由软件平台中的操作系统、分布容错处理软件、分布容错 API 协同完成。当飞行控制程序异常退出时，操作系统首先及时发现，并通知其他节点。目前的 Linux 等操作系统，进程退出只通过信号通知其父进程，其他进程无法及时发现，所以在分布式航天器系统的操作系统中增加容错处理功能，任何进程异常退出时，都要把相关的信息以信号等方式通知分布容错处理软件。分布容错处理软件将把系统中所有节点的状态变化信息、通信状态变化信息、飞行控制程序状态变化信息、软件加载和更新信息等通知飞行控制程序，以便可以在用户层进行相应的处理。

3 结束语

嵌入式系统在航天和车载、舰载、机载等军事应用中，针对恶劣和特殊的环境的容错处理是必须重点考虑的问题之一，提供强容错能力是嵌入式系统成败的关键。而强实时性也是非常关键的成败因素。在航天等应用中，嵌入式系统早已不是单个孤立节点，而是互相配合的分布式、综合处理系统。由于嵌入式系统从硬件平台(甚至可以从芯片级)开始，到软件结构、应用程序，都是针对具体应用的，具有很强的针对性，这使得经过一体化的精心设计来满足各种特定的需求成为可能。本设计方法应用到分布式航天器系统协同控制与信息处理子系统的设计与实现中，取得了较好的效果。

参考文献

1 Jensen J D. Command and Data Handling Subsystem Design for the Ionospheric Observation Nanosatellite Formation (ION_F)[R]. Utah State University, 2000.

2 Patterson P. Spacecraft Effects and Low Cost Components Capable of Sustaining Low Earth Orbits for a One Year Lifetime[R]. Utah State University, 1999.

3 Li Haiyan, Li Xinming. Embedded Operating System Design: The Resolved and Intelligent Daemon Approach[C]//Proc. of the 1st International Conference on Embedded Software and System, Hangzhou. 2004.

4 刘　东, 李　瑞. 空间计算机系统双机容错解决方案[C]//中国宇航学会首届学术年会, 北海. 2005.

(上接第 250 页)

3 结论

本文通过对非劣解集中粒子密度信息的估计算法、为群体中的粒子选择其 *gBest* 粒子的 Pareto 最优解搜索算法和非劣解集的多样性保持算法等方面的研究，提出了一种新的 MOPSO 算法，该算法有以下特点：(1)通过自适应网格方法有效地估计非劣解集中粒子的密度信息；(2)平衡全局和局部搜索能力的 Pareto 最有解搜索算法；(3)删除品质不好的多余粒子保持 Archive 集一定的规模，以改善算法的收敛性能。最后以三峡梯级优化调度问题为应用背景，通过本算法得到了协调三峡梯级电站年发电量和保证出力的非劣调度方案，为三峡梯级电站的调度决策提供科学的依据。同时也表明了 MOPSO 算法在求解大规模非线性多目标优化问题时的有效性。

参考文献

1 Zizlter E, Thiele L. Multiobjective Evolutionary Algorithm: A Comparative Case Study and the Strength Pareto Approach[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257-271.

2 Srinivas X, Deb K. Multiobjective Optimization Using Nondominated Sorting Genetic Algorithms[J]. Evolutionary Computation, 1995, 2(3): 221-248.

3 Deb K, Pratap A, Agarwal S. A Fast and Elitist Multiobjective Genetic Algorithms NSG-II[J]. IEEE Transaction on Evolutionary Computation, 2002, 6(2): 82-197.

4 Zitzler E, Laumanns M, Bleuler M. A Tutorial on Evolutionary Multiobjective Optimization[M]. Berlin, Germany: Springer-Verlag, 2004: 3-38.

5 Zitzler E, Deb K, Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results[J]. Evolutionary Computation, 2000, 8(2): 173-195.

6 Knowles J, Corne D W. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(2): 100-116.

7 Corne D, Knowles J, Oates M. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization[C]//Proceedings of the 6th International Conference on Parallel Problem Solving from Nature. 2000: 839-848.

8 Kennedy J, Eberhart R. Particle Swarm Optimization[C]//Proceedings of the IEEE International Conference on Neural Networks. 1995.

9 Mostaghim S, Teich J. Strategies for Finding Good Local Guides in Multi-objective Particle Swarm Optimization(MOPSO)[C]// Proc. of IEEE 2003 Swarm Intelligence Symposium. 2003: 26-33.

10 Everson J E, Fieldsend J E, Singh S. Using Unconstrainted Elite Archives for Multi-objective Optimization[J]. IEEE Transaction on Evolutionary Computation, 2003, 7(3): 305-323.

11 Coello C A C, Pulido G T, Lechuga M S. Handling Multiple Objectives with Particle Swarm Optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 256-279.

12 陈洋波, 王先甲, 冯尚友. 考虑发电量与保证出力的水库多目标优化方法[J]. 系统工程与实践, 1998, 3(4): 95-101.