

考虑节点失效恢复能力的网格服务 可靠性建模与分析

郭夙昌, 杨波, 黄洪钟

(电子科技大学机械电子工程学院, 610054, 成都)

摘要: 针对需要较长执行时间和占用较多计算机资源的“大任务”, 其节点可靠性随着时间的增加而呈指数衰减的问题, 给出了一种考虑节点失效恢复能力的网格服务可靠性模型. 在星型网格系统的网格服务可靠性分析中引入了节点失效恢复机制, 并考虑了节点软件可靠性对网格服务可靠性的影响, 同时采用子任务并行处理和子任务冗余方法提高服务的可靠性. 仿真结果验证了引入节点失效恢复机制对提高网格服务可靠性的积极影响, 为解决“大任务”可靠性偏低的问题提供了一种有效的解决方法.

关键词: 网格; 服务可靠性; 节点恢复

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2008)06-0693-05

Modeling and Analysis for Grid Service Reliability Considering Node Recovery

GUO Suchang, YANG Bo, HUANG Hongzhong

(School of Mechatronics Engineering, University of Electronic Science and Technology, Chengdu 610054, China)

Abstract: A reliability model for grid services that considers the fault recovery is presented to solve the problem that the grid service reliability decreases at exponential as the time increases, especially for some large subtasks that need long-lived computations and long-terms data storage. The ability of recovery into the grid nodes is introduced in grid systems with star topology, and the influence of software reliability is also taken into account. In order to improve the reliability of grid services, the grid services are divided into subtasks and then are assigned to different resources for processing in the proposed model. Numerical example is given to show that the recovery has a positive influence on grid service reliability and provides an effective solution to the fault tolerant for the services which consist of some large subtasks.

Keywords: grid; service reliability; node recovery

网格是继万维网之后出现的一种新型网络计算机平台^[1], 它将高速互联网、计算机、数据库等资源融为一体, 随之许多以往看来不可能完成的或是极其复杂的问题都迎刃而解, 但是现有的网格系统在可靠性和安全性等方面仍面临着诸多问题^[2], 因此提高网格系统的可靠性成为当前的一个难题.

网格服务可靠性是指用户提交的服务被网格成

功完成的概率, 是从用户的角度反映网格完成用户提交服务的执行能力^[3]. Dai 等人首先提出了网格服务可靠性的概念, 并采用最小档案扩展树(MFST)方法对网格服务可靠性进行了求解^[3]. Levitin 等人针对星型网格, 在综合考虑了服务可靠性和服务性能的同时, 提出了一种快速评估算法^[4].

在上述研究的基础上, 人们开始研究如何通过

资源优化或容错来提高网格的服务可靠性^[5],从而改善节点可靠性随着执行时间的增加而呈指数衰减的状况^[6],但这种机制未考虑本地节点的失效恢复能力,仅是借助了移动代理或者任务迁移^[7]等方式.邱敏等人提出了容错网格体系结构,并针对应用程序失效采用了本地检查点恢复机制^[8],却没有明确区分各种失效的类别. Heddaya 等人试图给出失效可恢复情况下系统服务可靠性的统一表达形式,但其分析过程出现了明显的错误^[9]. 另一方面,在网格系统服务可靠性中,硬件可靠性固然重要,软件可靠性亦不可忽视^[10].

本文将节点失效可恢复的情况引入到网格服务可靠性分析中,同时考虑节点软件可靠性对网格服务可靠性的影响,并在此基础上建立了网格服务可靠性模型.

1 网格服务执行过程

网格服务是开放式网格服务体系结构^[1]的核心,是在网格资源管理系统(RMS)的控制下,通过执行一系列子任务实现的. 网格在接收到用户提交的服务之后,RMS将服务分解成多个子任务,并将子任务分配到拥有子任务执行所需资源的节点. 节点完成子任务后,将执行结果返给 RMS. RMS 集成所有子任务的结果,并将最终执行结果返给用户. 这种网格服务的执行过程,可以用星型的拓扑结构来描述^[4]. 同时,假设节点与 RMS 之间有惟一一条通信链路,RMS 通过该链路向节点提交子任务,当子任务执行结束时,节点通过该链路将子任务执行结果返回 RMS.

为了提高网格服务的可靠性,充分利用网格资源的多样性,RMS 一般将单个子任务分配到多个节点并行执行,从而通过节点冗余的方法达到容错的目的. 这样,若有一个节点返回子任务的执行结果,则表明该子任务结束;若所有子任务均返回执行结果,则表明网格完成了该服务. 为了更好地描述失效对服务可靠性的影响,应对节点硬件、链路以及节点软件的失效情况做如下的基本假设.

(1)节点在获得了 RMS 分配的子任务之后,立即执行,并且其无失效执行时间服从指数分布^[3,5].

(2)节点软件失效的概率为常数.

(3)节点与 RMS 在交换信息期间,链路无失效工作的时间服从指数分布^[3,5].

(4)各子任务之间相互独立,网格内各组件(包括节点硬件、链路、节点软件)之间的失效相互独立.

由于 RMS 的可靠性比较高,工作时间相对较短,因此可以认为它在服务期间不会发生失效.

2 网格服务可靠性建模

2.1 网格节点的可靠性

假设服务 S 的总复杂度为 C ,RMS 将服务 S 分解成 m 个相互独立的子任务,其中第 i 个子任务 T_i ($i=1,2,\dots,m$)的复杂度为 c_i . 由于子任务之间的独立性, C 可以表示为各子任务的复杂度之和,即

$$\sum_{i=1}^m c_i = C \quad (1)$$

RMS 根据网格资源的分布情况,将子任务 T_i 分配到节点 k 来执行. 如果节点 k 的 CPU 处理速度为 s_k ,那么在不发生失效的情况下,子任务 T_i 在节点 k 的执行时间

$$\tau_{ki} = \frac{c_i}{s_k} \quad (2)$$

在实际的网格系统中,子任务 T_i 的执行往往因节点 k 发生失效而被迫终止. 为此,引入节点失效恢复机制,即当节点发生用户误操作、CPU 资源短时衰竭、网络瞬间堵塞或短期中断等可恢复故障时,节点自动运行失效恢复 JAVA 程序,恢复已停止的子任务执行程序,这样就可以有效解决大任务遇到失效而终止的问题. 当然,失效恢复具有一定的概率,即失效可以恢复,恢复后节点从失效断点继续执行子任务,直到遇到不可恢复的失效或者子任务执行完毕. 反之,失效不可恢复,任务执行被迫终止,在该节点的子任务执行宣告失败.

假定节点 k 的失效恢复概率恒为 x_k ,即对于任意的 j ($j=1,2,\dots$)都有概率 $P(X_k^{(j)}=0)=x_k$ 和 $P(X_k^{(j)}=1)=1-x_k$,且随机变量

$$X_k^{(j)} = \begin{cases} 0 & \text{节点 } k \text{ 发生的第 } j \text{ 次失效可以修复} \\ 1 & \text{节点 } k \text{ 发生的第 } j \text{ 次失效不可修复} \end{cases} \quad (3)$$

由于引入了节点失效恢复机制,节点 k 执行子任务 T_i 的过程被分为执行时间和恢复时间 2 个阶段. 设 $E_k^{(j)}$ 表示节点 k 在执行子任务 T_i 期间第 j 个执行阶段, $C_k^{(j)}$ 表示节点 k 在执行子任务 T_i 期间第 j 个恢复阶段,由于失效恢复后节点的失效率并不发生改变^[6],因此除最后一个子任务执行阶段外,其他 $E_k^{(j)}$ 都服从指数分布且相互独立. 根据指数分布的无记忆性,定义事件 U 为在“理想服务时间” τ_{ki} 内发生 α 次失效,那么在事件 U 发生的条件下, α 次失效均可恢复的概率为

$$P\left(\sum_{j=1}^{\alpha} X_{ki}^{(j)} = 0 \mid U\right)P(U) = \frac{(x_k \lambda_k \tau_{ki})^{\alpha}}{\alpha!} e^{-\lambda_k \tau_{ki}} \quad (4)$$

因为每一次失效、恢复都会给节点带来较大的开销,所以有必要限定节点的恢复次数.假设网格内所有节点的失效恢复次数不超过 L_c (L_c 为正整数),这样节点的可靠性将转变成节点在执行子任务期间不发生失效,或者是在发生的失效全部可恢复且失效次数不超过 L_c 的条件下,成功执行子任务的概率,即

$$R_{ki} = P(\alpha = 0) + \sum_{\alpha=1}^{L_c} P\left(\sum_{j=1}^{\alpha} X_{ki}^{(j)} = 0 \mid U\right)P(U) = e^{-\lambda_k \tau_{ki}} + \sum_{\alpha=1}^{L_c} \frac{(x_k \lambda_k \tau_{ki})^{\alpha}}{\alpha!} e^{-\lambda_k \tau_{ki}} = \sum_{\alpha=0}^{L_c} \frac{(x_k \lambda_k \tau_{ki})^{\alpha}}{\alpha!} e^{-\lambda_k \tau_{ki}} \quad (5)$$

根据不完整 Gamma 函数定义,可得

$$\Gamma[L_c + 1, x_k \lambda_k \tau_{ki}] = \int_{x_k \lambda_k \tau_{ki}}^{\infty} e^{-x} x^{L_c} dx = e^{-x_k \lambda_k \tau_{ki}} [(x_k \lambda_k \tau_{ki})^{L_c} + L_c (x_k \lambda_k \tau_{ki})^{L_c-1} + L_c (L_c - 1) (x_k \lambda_k \tau_{ki})^{L_c-2} + \dots + L_c!]$$

式(6)的变形为

$$\sum_{\alpha=0}^{L_c} \frac{(x_k \lambda_k \tau_{ki})^{\alpha}}{\alpha!} = \{(x_k \lambda_k \tau_{ki})^{L_c} + L_c (x_k \lambda_k \tau_{ki})^{L_c-1} + L_c (L_c - 1) (x_k \lambda_k \tau_{ki})^{L_c-2} + \dots + L_c!\} / L_c! = \{e^{x_k \lambda_k \tau_{ki}} \Gamma[L_c + 1, x_k \lambda_k \tau_{ki}]\} / L_c! \quad (7)$$

由于 L_c 为正整数,由 Gamma 函数的性质

$$\Gamma[L_c + 1] = L_c \Gamma[L_c] = L_c! \quad (8)$$

结合式(7),则式(5)可以写为

$$R_{ki} = \sum_{\alpha=0}^{L_c} \frac{(x_k \lambda_k \tau_{ki})^{\alpha}}{\alpha!} e^{-\lambda_k \tau_{ki}} = \frac{e^{x_k \lambda_k \tau_{ki}} e^{-\lambda_k \tau_{ki}} \Gamma[1 + L_c, x_k \lambda_k \tau_{ki}]}{\Gamma[1 + L_c]} \quad (9)$$

在更为一般的情况下,当 $L_c \rightarrow \infty$ 时,节点 k 的可靠性可以表示为

$$R'_{ki} = P(\alpha = 0) + \sum_{\alpha=1}^{\infty} P\left(\sum_{j=1}^{\alpha} X_{ki}^{(j)} = 0 \mid U\right)P(U) = \sum_{\alpha=0}^{\infty} \frac{(x_k \lambda_k \tau_{ki})^{\alpha}}{\alpha!} e^{-\lambda_k \tau_{ki}} = e^{x_k \lambda_k \tau_{ki}} e^{-\lambda_k \tau_{ki}} = e^{(x_k - 1) \lambda_k \tau_{ki}} \quad (10)$$

由式(10)可知,对于固定的子任务、节点,子任务的节点可靠性只取决于节点的失效率和失效恢复概率.特别是,当失效恢复率 $x_k = 0$ 时,节点 k 的可

靠性将转变为节点不具备失效恢复能力的情况^[4],这说明以往不具备节点失效恢复的网格可靠性分析方法本文方法的一种特例.

2.2 链路的可靠性

设节点 k 与 RMS 之间传输的总数据量为 a_i ,链路通信速度为 y_k ,那么节点 k 与 RMS 的通信时间

$$l_{ki} = \frac{a_i}{y_k} \quad (11)$$

根据第1节假设(2),若节点 k 与 RMS 的链路失效率为常数 π_k ,那么节点 k 与 RMS 通信时不发生失效的概率为

$$L_{ki} = P\left(l_{ki} = \frac{a_i}{y_k}\right) = e^{-\frac{\pi_k a_i}{y_k}} \quad (12)$$

节点 k 与 RMS 之间只有一条通信链路,如果在节点 k 与 RMS 交换信息期间通信链路发生了失效,子任务在节点 k 的执行将失败,子任务 T_i 也随之被终止.

2.3 网格的子任务可靠性

由于节点失效和链路失效具有独立性,所以子任务 T_i 能被节点 k 成功执行的概率为

$$R_{i,\{k\}} = P(E_{i,\{k\}}) = R_{ki} L_{ki} \quad (13)$$

式中: $E_{i,\{k\}}$ 表示事件中子任务 T_i 能被节点 k 成功执行.

为了提高子任务的可靠性,一个子任务一般分配到多个节点,当其中一个节点返回结果,则表示该子任务被成功执行.因此,网格的子任务可靠性为子任务 T_i 能够被网格成功执行的概率.为了简化起见,假设每个节点只执行一个子任务,又设执行子任务 T_i 的节点集合为 $D(i)$,那么对任意 2 个不相同的子任务 T_i, T_j ($i \neq j$),有 $D(i) \cap D(j) = \emptyset$, $\sum_{i=1}^m D(i) = \Omega$, 其中 Ω 表示全集.因此,在节点软件完全可靠的情况下,只要有一个子任务被某节点成功执行,该子任务可视为已成功完成.

软件出现错误是不可避免的,由软件错误导致的软件失效也是不可避免的^[10].为了更好地描述软件可靠性对子任务的影响,假设执行子任务 T_i 时的固定失效概率为 ϕ_i ,又设执行同一子任务 T_i 的节点具有相同的软件运行环境,则执行子任务 T_i 的所有节点软件具有相同的失效概率 ϕ_i .综合节点失效、链路失效和软件失效的影响,网格子任务的可靠性可表示为

$$R(T_i) = (1 - \phi_i) \left[1 - P\left(\bigcap_{k \in D(i)} \bar{E}_{i,\{k\}}\right) \right] \quad (14)$$

对于子任务 T_i , 由于各分配节点的运行情况相互独立, 因此网络的子任务可靠性可表示为

$$R(T_i) = (1 - \phi_i) \left[1 - P \left(\bigcap_{k \in D(i)} \bar{E}_{i, \langle k \rangle} \right) \right] = (1 - \phi_i) \left[1 - \prod_{k \in D(i)} (1 - R_{ki} L_{ki}) \right] \quad (15)$$

2.4 网络的服务可靠性

由于子任务之间相互独立, 因此网络的服务可靠性 $R(S)$ 可由服务 S 分解出的所有子任务在网格系统能被成功执行的概率来表示, 即

$$R(S) = \prod_{i=1}^m R(T_i) \quad (16)$$

式中: m 表示由服务 S 分解出的相互独立的子任务数目。

把式(15)代入式(16), 得到网格服务可靠性的表达式

$$R(S) = \prod_{i=1}^m (1 - \phi_i) \left[1 - \prod_{k \in D(i)} (1 - R_{ki} L_{ki}) \right] \quad (17)$$

2.5 参数确定

涉及网格服务可靠性的参数包括节点的 CPU 速度 s_k 、链路的带宽 y_k 、子任务的复杂度 c_i 、节点与 RMS 的通信信息量 a_i 、链路的失效率 π_{ki} 、节点的失效率 λ_k 及失效恢复率 x_k 、子任务执行软件的失效概率 ϕ_i 等。为了提供健壮、可靠、有效的网格环境, 全球网格论坛 (GGF) 制定了网格监控体系结构 (GMA)^[11], 其目标是对特定地点的资源状况及时地进行度量并发布, 这些资源包括软件资源 (应用、服务、进程和操作系统)、节点硬件资源 (CPU、内存、硬盘等) 和网络资源 (带宽、延迟等) 等。为了得到所需的参数, 可以适当完善网格监控系统的监控内容, 具体措施如下。

(1) 利用网格监控系统, 获得节点的 CPU 速度 s_k 、链路的带宽 y_k 、子任务的复杂度 c_i 、节点与 RMS 的通信信息量 a_i 等。

(2) 对特定的网格服务, 监控网格节点和链路的失效行为并记录节点和链路的工作时间。

(3) 监控网格节点的失效恢复能力。

(4) 子任务执行软件的失效概率 ϕ_i 可以通过监控软件执行结果来获取。

3 应用举例

给定一个网格服务 S , 即比较某汽车制造商历年来自设计零部件与采购零部件带来的利润, 并采用具有 5 个节点的星型网格系统对其进行计算。首先将服务分解为 2 个相互独立的子任务 T_1 (计算自主设计零部件产生的总利润) 和 T_2 (计算采购零部件产生的总利润), 并把子任务 T_1 分配给节点 1 和节点 2, 把子任务 T_2 分配给节点 3~节点 5, 各节点与子任务的属性如表 1 所示。

图 1 给出了节点失效恢复率在无约束条件下, 节点、子任务以及服务的可靠性随节点失效恢复率的变化趋势。从中看出, 当 $x_k = 0$ 时 (此时节点不具备失效恢复机制), 节点子任务、网格子任务及服务的可靠性都比较低, 而随着 x_k 的增加, 这些可靠性随之增加。由此得出, 如果节点具有一定的失效恢复能力, 网格服务可靠性将会大幅提高。若节点失效恢复次数不受限制, 当 $x_k \rightarrow 1$ 时, 节点发生的失效将全

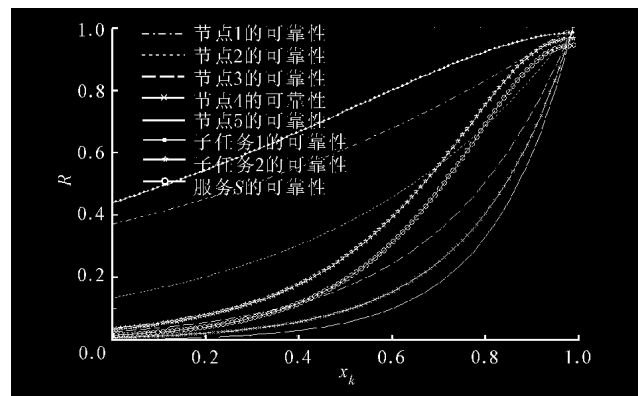


图1 失效次数不受约束时可靠度随节点失效恢复率的变化

表1 节点与各子任务的属性表

属性	子任务 T_1		子任务 T_2		
	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
τ_{ki}/s	1 000	1 002	1 200	1 190	1 170
l_{ki}/s	15	18	20	14	13
λ_k/s^{-1}	0.001	0.002	0.003	0.004	0.005
π_{ki}/s^{-1}	0.001	0.002	0.003	0.004	0.001
ϕ_i	0.02	0.02	0.04	0.04	0.04

部恢复,此时节点、子任务以及服务的可靠性都趋向1.图2给出了节点的失效次数不超过 $L_c=3$ 的情况,从中看出,由于节点4~节点6的失效率比较大,当 $x_k \rightarrow 1$ 时,节点的可靠性仍然很低,此时选择最优的失效次数阈值较为重要.

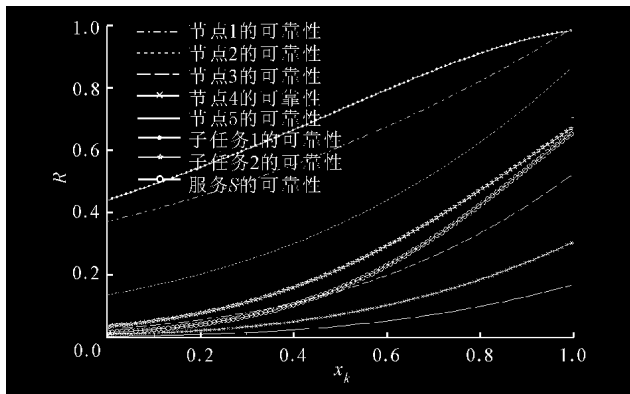


图2 失效次数不超过3次时可靠度随节点失效恢复率的变化

为了描述子任务的理想执行时间对服务可靠性的影响,先假定子任务 T_1 、 T_2 在不同节点上具有相同的理想执行时间,同时其他条件一样以及无失效次数约束的情况下,得到了服务可靠性与理想执行时间和节点失效恢复率之间的变化关系,如图3所示.从中看出,当子任务的执行时间为1200s时,如果节点不具备失效恢复能力,网格的服务可靠性几乎接近于0;当 $x_k=0.8$ 时, $R(S)=0.67$.由此可知,节点的失效恢复能力的大小对网格的服务可靠性起着极其重要的作用.

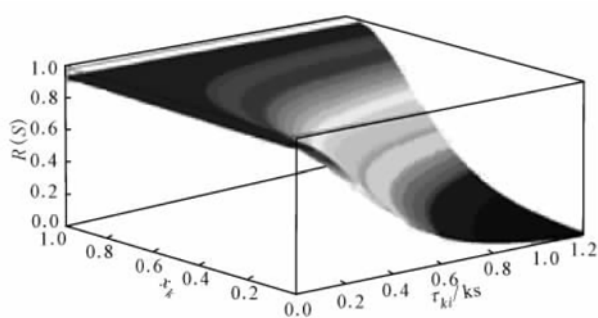


图3 服务可靠性与节点失效恢复率、任务时间的关系

4 结论与展望

本文将节点失效可恢复的情况引入到网格服务可靠性分析中,同时考虑了节点软件可靠性对网格服务可靠性的影响,在此基础上建立了星型网格的服务可靠性模型.由于网格系统的复杂性,仍然有许多问题等待解决.从模型上,子任务的执行相互独立、网格为星型拓扑结构等假设与实际网格系统并

不相符,因此研究广泛意义的网格服务可靠性更具有实际意义.软件冗余的特性在改进软件数值失效问题上没有意义,网络各节点的软件在系统平台、语言类别等方面存在着很大的差异,因此如何寻找网格系统中更好的软件失效处理方法也是值得研究的问题之一.在功能实现上,寻求合理的失效恢复的方法以及优化失效恢复次数阈值,仍需要深入研究.另外,在考虑服务可靠性的同时,服务性能也不能忽视,因此综合考虑网格服务可靠性和性能因素也将是今后网格可靠性研究的热点之一.

参考文献:

- [1] FOSTER I, KESSELMAN C. The grid 2: blueprint for a new computing infrastructure [M]. San Francisco, CA, USA: Morgan-Kaufmann, 2004.
- [2] FOSTER I. The grid: a new infrastructure for 21st century science [J]. Physics Today, 2002, 55(2):42-47.
- [3] DAI Yuansun, XIE Min, POH K L. Reliability of grid service systems [J]. Computers and Industrial Engineering, 2006, 50(1/2):130-147.
- [4] LEVITIN G, DAI Yuansun. Service reliability and performance in grid system with star topology [J]. Reliability Engineering and System Safety, 2007, 92(1):40-46.
- [5] DAI Yuansun, LEVITIN G, WANG Xiaolong. Optimal task partition and distribution in grid service system with common cause failures [J]. Future Generation Computer Systems, 2007, 23(2):209-218.
- [6] YANG Bo, XIE Min. A study of operational and testing reliability in software reliability analysis [J]. Reliability Engineering and System Safety, 2000, 70(3):323-329.
- [7] JOZSEF K, PETER K. A migration framework for executing parallel programs in the grid [C]//Proceedings of 2nd European Across Grids Conference. Berlin, Germany: Springer-Verlag, 2004:80-89.
- [8] 邱敏, 桂小林. 实现可靠计算的容错网格结构 [J]. 微电子学与计算机, 2005, 22(7):99-102.
- [9] QIU Min, GUI Xiaolin. Study of a fault-tolerant grid framework for dependable grid computing [J]. Microelectronics & Computer, 2005, 22(7):99-102.
- [10] HEDDAYA A, HELAL A. Reliability, availability, dependability and performability: a user-centered view [EB/OL]. [2006-11-08]. <http://www.cs.bu.edu/techreports/pdf/1997-011-reliability-def.pdf>.
- [10] XIE Min. Software reliability modeling [M]. Singa-

pore: World Scientific Publishing Company, 1991.

pers/GWD-GP-16-3. pdf.

- [11] Global Grid Forum Performance Working Group. A grid monitoring architecture [EB/OL]. [2006-08-27]. <http://www-didc.lbl.gov/GGF-PERF/GMA-WG/pa->

(编辑 苗凌)