一种面向自适应的动态配置方法

文志华，周序生，廖立君，李长云

（湖南工业大学 计算机与通信学院，湖南 株洲 412008）

摘 要：针对动态配置过程中更新时机选取、一致性保证和容错管理问题，提出一种自适应动态配置算法，算法采用阻塞和强制中断方式更新构件。以此算法为核心，运用AOP技术构建了一个自适应动态配置系统。

关键词：自适应；动态配置；系统一致性；AOP

中图分类号：TP311.52 **文献标识码**：A **文章编号**：1673-9833(2009)01-0050-04

Oriented Self-Adapting Approach for Dynamic Reconfiguration

Wen Zhihua, Zhou Xusheng, Liao Lijun, Li Changyun

（School of Computer and Communication，Hunan University of Technology，Zhuzhou Hunan 412008，China）

Abstract：A self-adapting dynamic reconfiguration algorithm on the base of system updating opportunity getting, system consistency and fault-tolerance management in the process of dynamic reconfiguration is presented, which used obstruction and forced interruption to replace components. Based on the algorithm and AOP, a self-adapting dynamic reconfiguration system is put forward finally.

Key words：self-adapting；dynamic reconfiguration；system consistency；AOP

在开放、动态的网络环境中，软件应具有动态调整和重配置的能力。在软件生命周期内，软件必须保持不断的进化，以修正软件故障，扩展服务功能，提高系统性能，适应新的运行环境和用户需求。金融数据处理系统、生命支持系统一旦发生停机事件，将会给经济和生命带来巨大损失，因此这些系统需要具有在不中断系统服务前提下更新功能的能力，即要求软件系统对变化具有自适应性。动态配置技术是实现软件自适应的一种重要手段，主要包含构件删除、构件添加、构件替换、构件迁移、连接删除、连接建立、连接重定向等[1]。其研究需要考虑如下关键问题：系统一致性问题，系统在动态配置期间和动态配置后必须处于正常运行状态中的约束；配置时机选取问题，如何在动态配置时选择构件更新的正确时机，保证动态配置正确开启；容错管理问题，如何在更新失败的情况下处理异常，使系统保持正常稳定的运行。

1 动态配置思路

为使得系统结构能动态调整，达到自适应的目标，动态配置需要关注配置的合法性，系统配置前后的正确性、一致性和配置的容错管理。一般自适应系统在实施时主要侧重在两个方面：一个是实现正确的新组件功能或组件的新功能；另一个是替换组件、更换版本、维护整个应用系统的正常运转[2]。根据内外环境的变化分析产生的自适应策略，用来指导如何调整系统配置，动态配置思路如图1所示。

收稿日期：2008-11-25

基金项目：国家自然科学基金资助项目（60773110），湖南省教育厅科学研究基金资助项目（07C234），湖南工业大学研究生创新基金资助课题（CX0811）

作者简介：文志华（1982-），男，湖南益阳人，湖南工业大学教师，硕士，主要研究方向为可信软件和软件自适应，
E-mail：wenzhihua@163.com
周序生（1970-），男，湖南常德人，湖南工业大学副教授，主要研究方向为网络安全与管理，数据库.

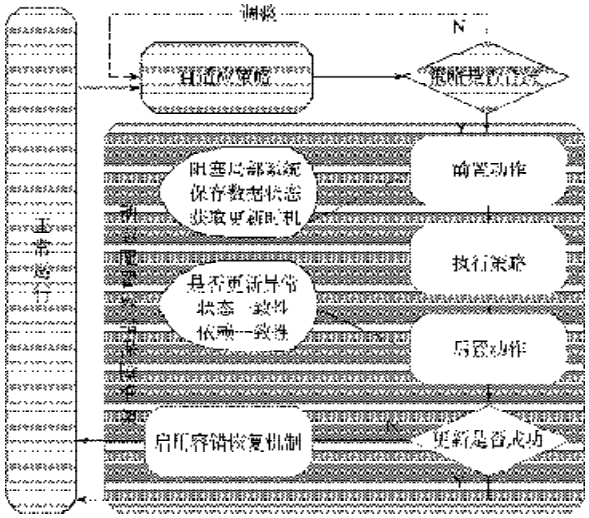

图 1　动态配置思路

Fig. 1　The idea of dynamic reconfiguration

动态配置框架获得自适应策略后，需要对其进行合法性判断。合法性判断是对策略的一种预评估，主要判断演化更新后系统能否达到原来系统所预定的目标状态。合法性判断成功后，系统进入局部阻塞状态。运行中的系统在更新时往往所需调整的组件处于运行状态，此时需要对调整的组件进行监视和控制，阻止其它依赖于此组件的调用。当配置时机成熟，系统实施更新操作，执行自适应策略。执行动作完成后，需执行后置条件，看系统是否达到了一致性要求和其它一些要求条件。一旦更新出现问题或后置条件不成立，则由更新保障器负责恢复系统到原来状态。

2　动态配置算法

2.1　动态配置前准备

动态配置时机需要通过构件调用管理器监视每个构件的调用行为来主动获取。其模型基本分成 2 种：调用模型和中断模型[3]，一般采用后者。构件调用管理器中记录了每个构件当前被动调用的关系和次数以及主动调用的关系和次数。当被更新的构件当前空闲，构件调用管理器通知构件管理器马上替换当前构件；如果构件不空闲，由调用阻塞器阻止当前构件的调用和被调用，直到获得构件空闲的时机进行更新操作；如果被更新构件调用超时，启用强制中断来获取更新时机。为了保证更新操作的正确、安全和一致，所有的操作都需要启动动态配置容错机制做保障。

2.2　配置算法

动态配置过程中涉及的元素：构件，构件提供的服务，构件引用，等待时间上限等。描述如下：

C 表示被更新的构件，C' 表示用来替换 C 的构件；S 表示构件提供的服务或功能，S_c 表示构件 C 提供的服务；N 表示依赖或调用，N_c=5 表示构件 C 的调用有 5 次，C_n 表示构件 C 的第 n 个实例；t 表示更新等待时间上限；P 表示构件数据状态或参数，$P(C_x, \cdots, C_y)_{\text{in}}$ 表示构件实例 $C_1, \cdots, C_n$ 的输入参数。

动态配置算法步骤：

1）启动更新，获取被更新构件 C，监视 $C_1, \cdots, C_n$，启动更新容错保障；

2）启动同时设定更新等待时间 t，启动倒计时，拦截构件 C 的调用 N_c'，并保存输入状态数据 $P(C_i')_{\text{in}}$；

3）对构件 C 的调用 N，即对 N_c 在 $t>0$ 时段里进行监视，拦截调用并对构件 C 的已调用计算，每完成一次调用 N_c 自减 1 次，并保存返回参数 $P(C_j)_{\text{out}}$；

4）如果 N_c=0，$t \geqslant 0$，构件 C' 替换 C；替换失败或产生异常进入第 7）步；

5）如果 t=0，N_c>0，构件 C' 替换 C，记录被强制替换为 $C_x, \cdots, C_y$；替换失败或产生异常进入第 7）步；

6）完成替换后，如果是由第 4）步完成替换，则对新构件 C' 恢复被拦截的调用 N_c'，并给予缓存状态数据 $P(C_i')_{\text{in}}$；如果是由第 5）步完成替换，则首先启用新构件实例 $C_x', \cdots, C_y'$，然后输入缓存状态数据 $P(C_1, \cdots, C_n)_{\text{in}}$，再恢复拦截调用；产生异常进入第 7）步，正常则进入第 8）步；

7）启动恢复机制，换回构件 C，回滚到更新初始状态；

8）关闭更新容错保障，进入正常运行状态。

在上述过程中，构件 C' 替换 C，要求新构件提供的服务或功能至少比替换前的构件要多，即 $S_c' \geqslant S_c$。

3　动态配置容错管理

动态配置操作的目的是修复现有系统的故障或扩散现有系统的功能，当然不可避免地会引入新的甚至更多的 bug。产生故障后为了恢复系统，最简单的方法就是重新启动服务。虽然重新启动服务也能把系统恢复到更新前的某一正常状态，但这种方式不能保证恢复到最近故障发生前的正常状态，而且重启服务这种方式无论从时间性能还是存储消耗方面来说都不是最佳途径[4]。从最安全可靠的角度来说，整个系统的状态应该被恢复到更新前的某个合法的、一致的状态，这不仅需要保存老的代码版本以防需要回滚，而且老状态数据也必须被保存，这样一般是需要消耗大量的存储空间的，此外可能会导致某些重要数据的丢失[5]。所以，在动态配置过程中系统需要定义实现相应的机制来实现更新中的容错，一旦更新被检测是非法的、不一致的，则需要进行相应操作的撤消、回滚。

在启动动态配置同时，系统启动更新容错保障，被替换构件 C 的实例 $C_1, \cdots, C_n$ 保存备份，同时从缓存中获取 $C_1, \cdots, C_n$ 的输入状态数据 $P(C_1, \cdots, C_n)_{\text{in}}$ 并保存备份，设置此状态 St_0。在更新过程中如果采用强

制中断，即由动态配置算法中第5步来完成构件替换，则在 t=0 的时刻保存备份实例 C_x，…，C_y，同时备份输入状态数据 $P(C_x，…，C_y)_{in}$，并设置此状态为 St_1。如果由第4步完成构件替换后产生异常，则启用备份并恢复到状态 St_0，如果由第5步完成构件替换后产生异常，则首先用备份并恢复到状态 St_1，如果失败，再恢复到状态 St_0。如图2所示。

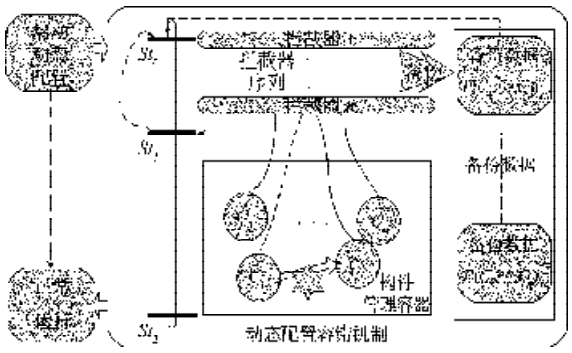

图2 动态配置容错管理

Fig. 2 Fault management in dynamic reconfiguration

4 自适应动态配置系统

自适应动态配置系统以动态配置算法为核心，主要分为4大部分：基础平台模块、可视化管理模块、自适应模块和动态配置模块，如图3所示。基础平台选择成熟的JBoss和Tomcat服务器，整个系统运行在其上。可视化管理模块采用Eclipse插件形式实现，主要是把系统中的一些信息反射出来，为用户提供方便。比如把元层视图以及对应的基层视图展现出来，包括构件之间的依赖关系，运行时刻的构件视图等。还包括系统进程数、资源消耗情况等一些系统状态的反映。

自适应模块主要包括上下文环境监视器、自适应策略制定器、自适应策略评估器和自适应效果反馈器。上下文环境监视器对系统运行的内外环境进行监视，把监视的数据保存，当遇到异常时，对异常数据和其它一些重要信息进行综合分析，把分析结果送给自适应策略制定器。自适应策略制定器根据分析结果，结合系统形式化约束和经验知识制定出自适应配置策略，再由自适应策略评估器进行预评估，预评估通过后由自适应动态配置系统负责执行。动态配置完成后，在完成时刻 St_2 起之后的 t 时间内，由自适应效果反馈器对动态配置的效果进行各方面的评估，并把结果反馈给自适应策略制定器形成经验知识。

动态配置模块主要包括容错管理框架、构件管理器和AOP机制几大部分。容错配置框架是动态配置得以正确完成的重要保障。构件管理器是构件替换、删除等操作的管理和执行者，其管理的构件采用反射技术分为2层：元层和基层。通过xml配置文件对元层构件进行描述，构件管理器通过xml配置文件负责对元层构件的维护和管理。基层构件实例通过图的形式进行描述。构件之间的通信和调用都需要通理器、需要通过AOP机制进行拦截和管理。主要包括构件调用管用计数器和调用阻塞器等，负责管理构件之间的调用依赖、调用的启动/阻塞、通信数据的截取等。元层和基层的交互(隐性)都需通过AOP机制里的各种拦截器进行调控。

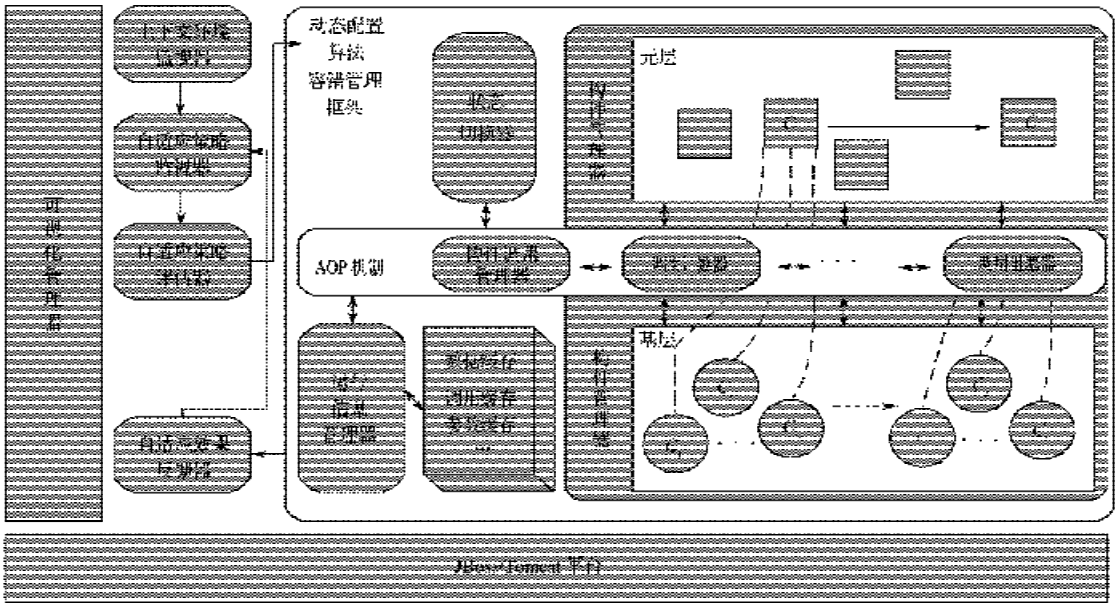

图3 自适应动态配置系统

Fig. 3 Adaptive dynamic reconfiguration system

5 进一步的工作

自适应动态配置的关键在于动态配置过程中的关键问题能否很好地解决。文章分析了自适应动态配置过程中的关键问题、一致性问题和更新时机获取问题，为了达到系统自适应的要求，提出了动态配置思

路，并在其指导下提出了具体的自适应动态配置算法。对动态配置过程的异常和故障处理提出了具体的动态配置容错管理方法，保证了动态配置前后的各种一致性。自适应动态配置平台保证了整个自适应全过程正确、安全和一致。在下一步的研究工作中，将在自适应环境感知、策略定制的全过程中引入形式化方法，从整体层面保证演化变更的正确性和一致性。

参考文献：

[1] Ajmani S. Automatic Software Upgrades for Distributed Systems [D]. Massachusetts： Department of Electrical Engineering and Computer Science，Massachusetts Institute of Technology (MIT)，2004.

[2] 李长云. 基于体系结构的软件动态演化研究[D]. 杭州：浙江大学，2005.
Li Changyun. Research on Architecture-Based Software Dynamic Evolution[D]. Hangzhou：Zhejiang University，2005.

[3] Hicks M. Dynamic Software Updating[D]. Philadelphia： Department of Computer and Information Science，University of Pennsylvania，2001.

[4] 王德俊，黄林鹏，徐晓辉，等. 分布式动态更新支持系统：研究综述[J]. 计算机科学，2007，34(11)：19-26.
Wang Dejun，Huang Linpeng，Xu Xiaohui，etl al. Survey of Supporting System for Dynamically Updating Distributed dystem[J]. Computer Science，2007，34(11)：19-26.

[5] EbraertP. Pitfalls in UnantieiPated Dynamic Softtware Evolution[C] // The Proceedings of the Workshop on Reflection，AOP and Meta-Data for Software Evolution (RAM-SE'05). Conjunction： ECOOP，2005：41-5l.

（责任编辑：罗立宇）

（上接第 35 页）

$z=\frac{x}{1+cx}\mp 1$，则 $y-\left(\frac{\alpha D}{P}\right)z-y_3$

$\frac{x}{1\ cx}\mp 1-\frac{1}{x}\pm 1=\frac{x^3-cx-1}{x+cx^2}$。

参考文献：

[1] 赵临龙. 可积的 Riccati 微分方程的不变量变换[J]. 数学的实践与认识，2008，38(16)：205-209.
Zhao Linlong. A Study on Invariant Transform of Riccati Differatial Equation[J]. Mathematics in Practice and Theory，2008，38(16)：205-209.

[2] 王玉萍，卢　琨，史胜楠. Riccati 方程的可积条件及通积分[J]. 陕西科技大学学报：自然科学版，2007，25(2)：136-138.
Wang Yuping，Lu Kun，Shi Shengnan. Integral Conditions and Representatives of Riccati Equation[J]. Journal of Shanxi University of Science & Technology：Natural Science Edition，2007，25(2)：136-138.

[3] Zhao Linlong. The Intergrable Condition of Riccati Differential Equation[J]. Chinese Quarterly Journal of Mathematics，1999，14 (3)：67-70.

[4] Zhao Linlong. A New Integrability Condition for Riccati Differeutial Equation[J]. Chinese Science Bulletin，1998，43 (5)：439-440.

（责任编辑：罗立宇）