

Migração de Processos: Panorama de Soluções Populares e Avanços Recentes

Márcio Parise Bouffleur, Rodolfo Leffa de Oliveira,
Andrea Schwertner Charão

Laboratório de Sistemas de Computação - LSC
Curso de Ciência da Computação – Universidade Federal de Santa Maria - UFSM
Informática/CT - UFSM Campus - 97105-900, Santa Maria, RS
{bouffleur, rodox, andrea}@inf.ufsm.br

Introdução

A migração de processos de um computador para outro é um tema recorrente em áreas ligadas à computação distribuída [SMI 88, NUT 94, MIL 2000]. No contexto do processamento de alto desempenho, mecanismos de migração de processos são potencialmente úteis para se alcançar objetivos clássicos, tais como distribuição dinâmica de carga, tolerância a falhas e localidade no acesso a dados.

Ao longo do tempo, as pesquisas nesta área levaram ao desenvolvimento de várias soluções para a migração de processos. Sistemas precursores como Mosix [BAR 98], Sprite [DOU 89] e Mach [MIL 92] foram implementados, mas obtiveram pouca utilização na prática [MIL 2000]. Porém, com o uso mais disseminado de sistemas distribuídos e a mudança do uso dos supercomputadores por redes de estações de trabalho, o tema novamente se mostra atraente.

Atualmente, alguns sistemas com suporte à migração de processos vêm se tornando populares, como é o caso de openMosix [OPE 2002] e OpenSSI [WAL 2001]. Além disso, pesquisas recentes têm produzido novas soluções na área, tais como Zap [OSM 2002], Xen [BAR 2003] e Kerrighed [VAL 2003]. Todos estes sistemas diferem entre si sob vários aspectos, de modo que a seleção de uma ou outra solução é uma tarefa pouco trivial.

Neste trabalho tem-se como objetivo traçar um panorama dos desenvolvimentos recentes e das soluções mais populares sobre migração de processos, fornecendo subsídios para administradores de sistemas e pesquisas futuras. As seções seguintes apresentam as soluções analisadas, organizadas de acordo com a abordagem utilizada para migração de processos.

Migração em *Clusters* SSI

O acrônimo SSI (*Single System Image*) designa sistemas que oferecem a ilusão de que diversos recursos distribuídos (memória, disco, CPU) formam um único e compartilhado conjunto de recursos. Um *cluster* SSI é visto pelo usuário como se fosse uma grande máquina SMP (multiprocessador simétrico), destacando-se pela facilidade de uso e programação [LOT 2005]. Dentre as ferramentas de migração de processos que operam sobre *clusters* SSI, pode-se citar openMosix, Kerrighed e OpenSSI.

Desenvolvido na Universidade Hebraica de Jerusalém, **openMosix** [OPE 2002] é uma implementação livre do sistema Mosix. Em openMosix, a migração de processos é utilizada para distribuir a carga de trabalho uniformemente pelos nós de um *cluster*. Para tal, openMosix utiliza um algoritmo adaptativo de compartilhamento de recursos que leva em conta não somente a carga de CPU, mas também a utilização de memória.

Em openMosix, um processo migrante continua possuindo uma forte relação com a máquina onde foi originalmente criado. Para resolver as chamadas de sistema (comunicações de rede, acesso a arquivos) que não poderiam ser processadas localmente após a migração, o processo migrante se comunica com a máquina de origem.

A implementação de openMosix é intrusiva, pois requer alterações no código do sistema operacional. Porém, a interface de programação de openMosix é bastante simples, sendo disponibilizada no sistemas de arquivos `/proc`.

O sistema **OpenSSI** [WAL 2001] é uma plataforma desenvolvida com o objetivo de integrar diversas tecnologias de código aberto para *clusters*. O mecanismo de migração utilizado em OpenSSI é derivado do Mosix. Quando ocorre uma migração em OpenSSI, todo o processo é enviado para o nó destino. Assim, todas as chamadas de sistema são executadas remotamente, permitindo que comunicações entre processos, acesso a arquivos e chamadas de sistemas não precisem retornar ao nó de origem, acelerando e descentralizando as tarefas.

Kerrighed [VAL 2003] é um sistema formado por um conjunto de serviços encarregados de toda a gerência dos recursos do *cluster*. Este sistema oferece um escalonador de processos global configurável. A migração de processos é baseada na carga da CPU dos nós, e é implementada através de vários mecanismos.

Migração Através de *Checkpoint/Restart*

Nesta abordagem para migração de processos, a idéia de base consiste em salvar em disco uma imagem com o estado atual do processo migrante (*checkpoint*). Esta imagem pode então ser enviada para outra estação, permitindo reiniciar a execução do processo a partir do mesmo estado (*restart*).

Existem vários sistemas baseados em *checkpoint/restart*. Dentre os exemplos mais recentes, destaca-se a ferramenta **Zap** [OSM 2002], desenvolvida na Universidade da Columbia.

Zap utiliza o conceito de domínios de processos (*process domains - pods*) a fim de que, quando ocorrer uma migração, todos os processos que dependem do processo migrante sejam migrados também. Para isso, Zap utiliza o mecanismo de *checkpoint/restart* provido pela ferramenta CRAK [ZHO 2001]. Esse mecanismo, associado ao redirecionamento das conexões de rede, garante que nenhum resíduo do processo migrante permaneça na máquina de origem.

Zap não requer nenhuma modificação no núcleo do sistema operacional, apenas a carga do módulo dinâmico do CRAK. Porém, como CRAK utiliza-se de algumas funções do núcleo que não são acessíveis, precisa-se de alguns artifícios para usá-las.

Sua interface de programação é feita através de uma biblioteca, provida no mesmo pacote.

Migração Baseada em Máquinas Virtuais

Com o aumento do desempenho dos computadores modernos, os ambientes de execução baseados em máquinas virtuais vêm gradualmente despertando o interesse de administradores de redes e sistemas distribuídos. Máquinas virtuais provêm uma abstração do hardware e arquitetura subjacentes, permitindo que múltiplos sistemas e aplicações executem concorrentemente sobre os recursos virtualizados.

Em sistemas baseados em virtualização, pode-se realizar a migração da própria máquina virtual, o que permite reduzir as dependências residuais entre o processo migrante e a máquina de origem. Esta abordagem é explorada, por exemplo, em soluções baseadas no sistema **Xen** [BAR 2003], desenvolvido na Universidade de Cambridge.

Xen é um sistema de virtualização de código aberto. Para funcionar adequadamente sobre Xen, o sistema operacional precisa ser modificado, tornando-se consciente de sua execução sobre uma máquina virtual. Atualmente, o núcleo do sistema Linux (versões 2.4 e 2.6) possui suporte à arquitetura de virtualização do Xen, sendo que o porte de outros sistemas operacionais também está em desenvolvimento.

Com Xen é possível realizar a migração de uma máquina virtual para outro hospedeiro físico. Para tal, utiliza-se a técnica de "migração viva", que permite migrar o sistema operacional inteiro para outro hospedeiro físico, precisando fazer apenas uma pequena parada, quase imperceptível do ponto de vista do usuário [CLA 2005].

A interface de controle da migração em Xen é feita por uma máquina virtual onde o console administrativo está instalado.

Conclusão

Com este trabalho, buscou-se apresentar uma breve introdução sobre o contexto de utilização de ferramentas que efetuam migração de processos. Neste contexto, discutiu-se as soluções mais populares e as de pesquisa mais recente.

Fica a critério do usuário eleger a solução que melhor atende às suas necessidades, visto que cada ferramenta possui suas aplicações e particularidades.

Referências

- [BAR 98] BARAK, A. The MOSIX multicomputer operating system for high performance cluster computing. **Future Generation Computer Systems**, v.13, n.4–5, Mar. 1998.
- [BAR 2003] BARHAM, P. et al. Xen and the art of virtualization. In: NINETEENTH ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES (SOSP), 2003, Bolton Landing, NY, USA. **Proceedings...** ACM, 2003. p.164–177.
- [CLA 2005] CLARK, C. et al. Live migration of virtual machines. In: SECOND SYMPOSIUM ON NETWORKED SYSTEMS DESIGN AND IMPLEMENTATION (NSDI '05), 2005, Boston, MA, USA. **Proceedings...** Usenix, 2005.

- [DOU 89] DOUGLIS, F. Experience with process migration in Sprite. **Workshop on Experience with Building Distributed (and Multiprocessor) Systems**, p.59–72, Oct. 1989.
- [LOT 2005] LOTTIAUX, R. et al. Openmosix, OpenSSI and Kerrighed: a comparative study. In: IEEE INTERNATIONAL SYMPOSIUM ON CLUSTER COMPUTING AND THE GRID (CCGRID '05), 2005, Cardiff, UK. **Proceedings...** IEEE, 2005.
- [MIL 2000] MILOJICIC, D. et al. Process migration. **ACM Computing Surveys**, v.32, n.3, Sept. 2000.
- [MIL 92] MILOJICIC, W.; ZINT, W.; DANGEL, A. **Task migration on top of the Mach microkernel - design and implementation**. Kaiserslautern: [s.n.], 1992.
- [NUT 94] NUTTALL, M. **Survey of systems providing process or object migration**. London, England: Imperial College London, DOC, 1994. (94/10).
- [OPE 2002] OPENMOSIX. <http://openmosix.sourceforge.net/>.
- [OSM 2002] OSMAN, S. et al. **The design and implementation of Zap: a system for migrating computing environments**.
- [SMI 88] SMITH, J. M. A survey of process migration mechanisms. **ACM SIGOPS**, v.22, n.3, July 1988.
- [VAL 2003] VALLÉE, G. et al. A case for single system image cluster operating systems: Kerrighed approach. **Parallel Processing Letters**, v.13, n.2, June 2003.
- [WAL 2001] WALKER, B. J. **OpenSSI Linux cluster project**. <http://openssi.org>.
- [ZHO 2001] ZHONG, H.; NIEH, J. **CRAK: Linux checkpoint/restart as a kernel module**.