

# Grid-aware operating systems

Carla Leite  
200902780

Carlos Castro  
200907147

## Introdução

O conceito de Grid Computing apresenta um novo paradigma computacional, que tem como objectivo a integração de recursos computacionais através de uma infra-estrutura capaz de oferecer confiabilidade, consistência e persistência na execução de aplicações que necessitam de um grande poder computacional[1].

A ideia inicial era permitir a interacção entre centros de computação independentes, que poderiam estar distantes geograficamente de modo a poderem executar tarefas com elevada complexidade de forma distribuída e garantir a maximização da utilização dos recursos. Teoricamente, para se utilizar um sistema de computação Grid o programador apenas teria de escrever o seu algoritmo de computação como se este fosse executado num computador local, submetê-lo para o sistema Grid e este distribuiria a execução da tarefa por vários computadores espalhados pelo Mundo. No final, apenas seria necessário recolher os resultados da computação. Esta é uma ideia sedutora para qualquer utilizador de Informática. No entanto, a concretização de plataformas Grid envolve a resolução de problemas em aberto inerentes aos sistemas distribuídos como por exemplo, sincronização entre processos, transferência e armazenamento de grandes volumes de dados e interoperabilidade entre sistemas heterogéneos.

De forma a minimizar esta complexidade surgiram novos sistemas operativos. Entre eles: XtremOS, Mosix e Vega Grid. O trabalho, de forma geral, consiste em descrever o seu funcionamento, arquitectura e vantagens aplicadas na Grid.

## XtremOS

XtremOS é um sistema operacional baseado em Linux que suporta organizações virtuais em plataformas de computação Grid e visa integrar-se como uma única plataforma de computação em diversos tipos de dispositivos, desde dispositivos móveis a grandes *clusters*[2]. Este sistema oferece à Grid o que um sistema operacional oferece a um único computador: abstracção de *hardware* através da ilusão que o XtremOS transmite aos utilizadores ao usar um computador tradicional enquanto remove os problemas de recursos complexos num ambiente Grid e segurança na partilha de recursos entre diferentes utilizadores uma vez que quer os utilizadores quer os recursos são geridos por meio de uma Organização Virtual (VO) que recorre a mecanismos de segurança como autenticação e autorização.

Para a implementação deste novo sistema operativo a Grid foi sujeita a algumas alterações ao nível da interoperabilidade uma vez que o XtremOS está em conformidade com os principais padrões como o Posix, ao nível da segurança dado que este sistema garante

confiança e integridade de acordo com políticas personalizáveis e a nível de escalabilidade uma vez que o XtreamOS suporta dezenas de nós e utilizadores que entram e saem da Grid de forma dinâmica.

A arquitectura geral deste sistema consiste no empilhamento de duas camadas principais de software: um XtreamOS-F que suporta a plataforma localmente e é específico do dispositivo e um segundo, XtreamOS-G, que se encarrega do nível de rede e integra diferentes dispositivos para uma única plataforma de computação[3].

O XtreamOS tinha o desafio de desenvolver a primeira plataforma de computação em escala planetária, de confiança e de código aberto que explorasse um suporte seguro e escalável para organizações virtuais para permitir federação de recursos. O desafio consistia em fornecer uma facilidade de uso, de administração e de confiabilidade com um sistema operacional comum enquanto explora uma plataforma heterógena e omnipresente com alto desempenho e escalabilidade. O objectivo do projecto para além do desenvolvimento da plataforma passa também por promover uma comunidade de desenvolvedores que explorem as extensões deste XtreamOS para Linux e isso motivou muitas das escolhas de concepção do projecto que visa facilitar a interacção entre a comunidade *open-source* e a adopção das melhores práticas.

## MOSIX

O MOSIX é um pacote de ferramentas que acrescenta ao kernel do Linux funcionalidades de gestão de sistemas cluster. A sua principal característica é fazer com que todos os nós participantes se comportem como se de um único computador se tratasse, semelhante a um SMP (Symmetric Multi-Processing). Por exemplo, os utilizadores podem correr aplicações que criam vários processos, e o MOSIX procura os recursos e distribui de forma automática os processos pelos nós que constituem o sistema, melhorando assim a performance no geral. Como resultado, os utilizadores não têm que modificar as aplicações, não têm que fazer login ou copiar ficheiros para máquinas remotas e estão a par de onde os seus programas estão a ser executados [4].

Inicialmente, quando MOSIX foi desenvolvido, 1977, só geria um único cluster. Só em 2004, numa versão que pode ser chamada de MOSIX-2 é que passou a suportar clusters e multi-clusters, como por exemplo, diferentes grupos numa organização, ou computação grid . O objectivo de um MOSIX grid é permitir aos proprietários dos nós partilhar os seus recursos computacionais de tempos a tempos, podendo a qualquer altura desligarem esses nós da grid sem sacrificar os processos migrados de outros nós. Para que isto seja possível, o MOSIX é composto por algumas características direccionadas a sistemas grid como por exemplo[5]:

- **Clusters lógicos.** Os nós são divididos em clusters lógicos que depois são alocados a um utilizador ou objectivo específico. Cada utilizador só corre os seus processos num único cluster lógico.
- **Descoberta de recursos de forma automática.** Algoritmos encarregues de verificar os estados dos nós, assim como a entrada ou saída de nós na grid.

- **Migração de processos.** Algoritmos que fazem a migração de forma automática dos processos de um nó para o outro. Estes algoritmos tentam sempre melhorar o desempenho dos processos, como por exemplo, migrar um processo que necessite de mais memória do que aquela que tem disponível no nó em que está a ser executado.
- **Checkpoint e recuperação.** Quando um processo entra em estado de checkpoint, todos os dados correspondentes a esse processo são guardados num ficheiro sendo posteriormente lido esse mesmo ficheiro para resumir o processo. Os checkpoints podem ser feitos manualmente ou automaticamente.
- **Batch jobs.** Os batch jobs são a execução de binários num nó em que só podem ler e escrever no standard input/output.
- **Monitor.** Mostra informação relativo ao estado dos recursos do sistema, como por exemplo, número de nós activos, memória usada, memória disponível, estado dos CPUs.

## Vega GOS

O Vega Grid Operating System foi desenvolvido pelo Instituto da Tecnologia e Computação da Academia das Ciências Chinesa e fundado pelo Ministério da Ciência e Tecnologia Chinês no âmbito do projecto Vega Grid. O projecto Vega Grid tinha como objectivo aprender as propriedades fundamentais sobre computação grid e desenvolver técnicas importantes que seriam essenciais para a construção de sistemas e aplicações grid [6].

O Vega GOS é uma framework que segue os princípios de uma arquitectura orientada a serviços (SOA). Esta framework é dividida em camadas de acordo com as diferentes funcionalidades, camada core, camada system e camada aplicação. A camada core é responsável por gerir os requisitos das aplicações grid como por exemplo, memória, processos, etc. A camada system tem como função disponibilizar um conjunto de bibliotecas e toolkits que são desenvolvidos para o correcto funcionamento da camada core. Esta camada é semelhante ao conjunto de ferramentas e ficheiros de configuração que é chamado de software do sistema nos tradicionais sistemas operativos. A camada aplicação é onde o utilizador corre as suas aplicações num ambiente grid mas sendo este abstraído do que se está a passar nas camadas inferiores. Para as aplicações poderem comunicar com a camada core, foi desenvolvida uma API, Vega GOS API, onde estão escritas as chamadas de funções para a interacção.

## Conclusão

Os sistemas operativos vocacionados para o ambiente grid vieram trazer uma nova abordagem à forma como os sistemas grid são utilizados no sentido de facilitar de forma segura e fidedigna a execução das aplicações por parte dos utilizadores. Todos os processos complexos necessários para o funcionamento de um sistema grid como procura de recursos,

verificar os estados dos nós, são abstraídos dos utilizadores, facilitando assim o desenvolvimento e a execução das aplicações. Estes sistemas operativos disponibilizam ainda ferramentas gráficas que permitem um controlo e visualização relativa aos estados dos processos.

## **Referências**

- [1]** What is Grid? A Three Point Checklist, Ian Foster, Argonne National Laboratory & University of Chicago, 20 Julho 2002.
- [2]** XtreamOS: a vision for a Grid Operating System, ToniCortes,Carsten Franke, Yvon Jégou, Thilo Kielmann, Domenico Laforenza,Brian Matthews,Christine Morin, Luis Pablo Prieto, and AlexanderReinefeld, 14 Maio 2008.
- [3]** XtreamOS: from Grid to Clouds, Christine Morin, Illinois-INRIA workshop – Bordeaux, 23 Junho, 2010.
- [4]** Cluster Operating Systems, Russell W Clarke, Benjamin T B Lee, School of Computer Science and Software Engineering, Monash University.
- [5]** MOSIX for Organizational Grids - A White Paper, A. Barak and A. Shiloh, Agosto 2006.
- [6]** CNGrid Software 2.0: Service Oriented Approach to Grid Computing, X. Xie, N. Xiao, Z. Xu, L. Zha, W. Li, H. Yu, Institute of Computing Technology, Chinese Academy of Sciences.