

ARQUITECTURA DE UN SISTEMA OPERATIVO WEB

JOSE AGUILAR

aguilar@cemisid.ing.ula.ve

EDGAR FERRER

eferrer@cemisid.ing.ula.ve

NIRIASKA PEROZO

nperozo@cemisid.ing.ula.ve

JUAN VIZCARRONDO

jvizcarrondo@cemisid.ing.ula.v

Investigadores Centro de investigaciones en Micro-computación y Sistemas Distribuidos- CEMISID -.
Dpto. de Computación, Facultad de Ingeniería, Av. Tulio Febres.
Universidad de los Andes
Mérida 5101

VENEZUELA

RESUMEN

La cantidad de sistemas, servicios y aplicaciones desarrolladas para la web han crecido considerablemente, en algunos casos el soporte, por parte de los sistemas operativos existentes, a cada uno de ellos, no es el esperado. Como alternativa de solución a esta necesidad, se plantea un modelo del sistema operativo denominado SOW, el cual soporta y maneja un conjunto de servicios en un contexto heterogéneo, dinámico y adaptativo, bajo el enfoque de reconfiguración de las aplicaciones, conformado por cuatro subsistemas que llevan a cabo una serie de funciones coordinadas que permiten un uso eficiente de los recursos sobre Internet, a pesar de sus características dinámicas y de heterogeneidad.

PALABRAS RELEVANTES

- Sistema operativo
- Objetos web
- Sistema manejador de objetos web
- Sistema manejador de comunidades

1. INTRODUCCIÓN

Dada la amplia variedad de servicios inimaginables que surgen cada día en la web, resulta difícil diseñar un sistema operativo que apoye/use a cada uno de esos servicios. De éstas necesidades surge una corriente de desarrollo llamada Sistema Operativo Web (SOW), que tiene como objetivo principal proveer una plataforma que permita a los usuarios beneficiarse del potencial computacional ofrecido en la web, a través del compartimiento de recursos y de la resolución de los problemas de heterogeneidad y adaptabilidad dinámica presentes en la misma. Así, para alcanzar un rendimiento óptimo en un ambiente dinámico de recursos distribuidos como la Internet, el SOW debe ser configurable y capaz de adaptarse a los cambios en cuanto a la disponibilidad de recursos (de software y de hardware). Teniendo en cuenta esas consideraciones, el modelo de SOW presentado en este trabajo propone una serie de aspectos para proveer servicios que se adecuen a esos rasgos especiales de la web. Así, el SOW presenta un diseño que cuenta con las herramientas asociadas para permitir el uso transparente e interactivo de los recursos accesibles a través de la red, en cualquier momento que un usuario lo requiera. Esos servicios pueden ser hardware, software, o una combinación de ambos. El usuario sólo necesita comprender la interfaz del SOW, sin importarle como su solicitud es satisfecha.

Existen varias propuestas para el manejo e integración de los recursos computacionales disponibles en la Web. Quizás el proyecto más general sea el WOS™ [4], ya que permite el manejo e integración de los recursos tratando el problema de la heterogeneidad y volatilidad en la Web. Este proyecto, al igual que la presente propuesta, está basado en la idea del uso de versiones como solución a esos problemas. Otra propuesta para el manejo de recursos en la Web parecida, es la arquitectura Jini de SUN Microsystems [9]. Jini provee servicios de localización de recursos (conocidos como *lookups*), y aplica la idea de agrupar recursos en federaciones.

Otros esfuerzos para explotar los recursos distribuidos en Internet incluyen 2k [16], el cual es

una arquitectura de un sistema operativo distribuido para el manejo de recursos en redes heterogéneas que utiliza CORBA y la configuración de aplicaciones distribuidas basada de componentes para enfrentar los problemas de heterogeneidad; **NetSolve**, el cual es un sistema de cómputo distribuido que ofrece funcionalidades para aplicaciones científicas [7]; **Globe**, el cual es un sistema para manejo de recursos globales donde se implementan estrategias adaptativas de replicación de recursos web [11]; **Legión**, el cual es un metasisistema que permite interconectar un conjunto de computadores heterogéneos y geográficamente distribuidos, ofreciendo una máquina virtual coherente y simple [12]; **Globus**, que construye una rejilla (grid) computacional que permite el acceso a los recursos computacionales independientes de la posición geográfica [8]; y **WebOS**, que provee servicios de un sistema operativo básico necesitados para construir aplicaciones que están distribuidas geográficamente y deben ser reconfigurables dinámicamente [3]. En contraste con la presente propuesta, la mayoría de estos sistemas requieren privilegios de entrada (login) en las máquinas participantes y un software acorde a las arquitecturas participantes.

Existen otras propuestas en donde estos problemas de interoperabilidad de plataformas han sido manejados exitosamente a través de Java. Entre éstas tenemos Charlotte, el cual es un sistema que provee facilidades a los programadores de aplicaciones paralelas para escribir programas en Java y ejecutarlos desde un navegador web [6]; **ATLAS**, que es un sistema para ejecutar programas paralelos multi-hilos [5]; **Popcorn**, que es un sistema de metacomputación que permite que cualquier computador que disponga de una máquina virtual Java pueda participar en el metasisistema [13]; y **Javelin**, que es un sistema que provee un ambiente para la ejecución de aplicaciones paralelas, donde el cómputo paralelo se distribuye entre nodos de Internet [10]. Esos proyectos están dirigidos principalmente a modelos de programación orientados en Java para computación paralela basada en Internet. SOW no es dependiente del modelo de programación, aunque los modelos de programación orientada a Java pudieran perfectamente ser integrados.

Como se mencionó anteriormente, el modelo propuesto, al igual que el WOS™ y jini, es diferente al resto de los proyectos mencionados, en el sentido de que no se requiere ningún catálogo global