

## EXPERIENCIAS DE PRÁCTICAS CON PVM EN LA ASIGNATURA DE ARQUITECTURA DE COMPUTADORES Y SISTEMAS OPERATIVOS 2

Miguel A. Mateo Pla<sup>1</sup>, Julio Pons Terol<sup>1</sup>

<sup>1</sup>*Departamento de Informática de Sistemas y Computadoras (DISCA)  
Universidad Politécnica de Valencia (UPV)  
e-mail: {mimateo,jpons}@disca.upv.es*

**RESUMEN:** En el presente artículo se presentan las experiencias obtenidas durante la preparación y realización de una práctica con PVM [2] (Parallel Virtual Machine) para la asignatura Arquitectura de Computadores y Sistemas Operativos 2 (ACSO2), impartida en octavo semestre en la Escuela Técnica Superior de Ingenieros de Telecomunicación (ETSIT) de la UPV.

Esta práctica se desarrolla en torno a un problema sencillo con necesidades computacionales altas, y con una sustancial ventaja sobre otros problemas estudiados: una salida gráfica. Este hecho permite al alumno apreciar mejor el paralelismo y las diferentes formas de resolver el algoritmo.

Se justificará la elección de PVM como entorno de programación y las herramientas utilizadas como ayuda al desarrollo de la práctica.

### 1.- INTRODUCCIÓN.

La asignatura ACSO2 se imparte por primera vez en el curso 1999-2000. Esta asignatura se imparte en el semestre octavo de todas las especialidades de la titulación de Ingeniero de Telecomunicación de la ETSIT.

Esta nueva asignatura surge de la implantación de los planes de estudio de 1996. En el plan de estudio anterior existía la asignatura anual ACSO de la que se han aprovechado algunos materiales. Se ha aprovechado el paso al nuevo plan de estudios para revisar y actualizar tanto el temario como los materiales y prácticas utilizados. El resultado ha sido la introducción de nuevos temas en el temario y sus prácticas correspondientes.

Uno de los temas que se ha introducido es el de las Redes de Estaciones de Trabajo (en inglés Network Of Workstations ó NOW). Este tema se trata como un aspecto actual de los sistemas operativos y de la arquitectura de computadores.

Para comprender el funcionamiento de las NOW se necesitaba encontrar un entorno de programación en el que preparar la práctica y una aplicación susceptible de paralelizarse de forma sencilla.

Con la experiencia obtenida en las prácticas de asignaturas anteriores [1], se ha observado que los alumnos se motivan más con aquellos problemas en los que pueden ver el resultado de la computación de forma gráfica. Las prácticas de asignaturas anteriores contaban con esa salida gráfica, pero se utilizaba un simulador de multicomputadores por lo que algunos alumnos tenían la impresión de que las ejecuciones no eran realistas.

Teniendo estos conceptos en mente se diseña una práctica que pretende la utilización de una NOW para la solución de un problema sencillo de forma paralela, sin utilizar simuladores y obteniendo resultados gráficos atractivos.

En este artículo presentamos los objetivos concretos perseguidos con esta práctica y las experiencias obtenidas durante su preparación, realización y posterior evaluación.

## 2.- OBJETIVOS

Los objetivos principales de la práctica son:

1. Introducir de forma práctica una NOW.
2. Dar a conocer al alumno una herramienta de programación de NOW.
3. Que el alumno sea capaz de escribir o modificar una aplicación paralela sencilla.
4. Conseguir que el alumno descubra los principales problemas de la programación distribuida.

Para facilitar el trabajo de los alumnos en otros laboratorios o en su casa, es deseable que la práctica se pueda probar en entornos diferentes de los disponibles actualmente en el Laboratorio de Arquitectura de Computadores del DISCA, por ello se ha escogido un entorno de programación de NOW lo más independiente posible del sistema operativo y de la arquitectura de la máquina.

## 3.- ENTORNO

Los dos condicionantes básicos del diseño de la práctica son: el laboratorio y el perfil del alumno. Sin definir con claridad ambos entornos será probable que la práctica no se pueda realizar o que los alumnos no puedan sacarle el aprovechamiento deseado.

Por este motivo se definen a continuación estos dos elementos.

### a) El laboratorio de Arquitectura de Computadores

Todas las prácticas de ACSO2 se realizan en el laboratorio de Arquitectura de Computadores. Este laboratorio cuenta con 27 puestos de prácticas con las siguientes características:

- puestos con AMD K6-2 450 MHz.
- 3 puestos con Pentium II a 350 MHz.
- 6 puestos biprocesadores Pentium II a 350 MHz.
- 4 puestos biprocesadores Pentium 200 MMX
- 3 puestos con Pentium 166 MMX.

Todos los puestos cuentan con 64 MB de RAM, 8 GB de disco duro y monitor de 17". Todos los puestos están conectados mediante una red Ethernet a 100 Mb.

Los grupos de prácticas previstos para la realización de prácticas son 12 grupos de 2 personas. De esta forma se consigue tener grupos de prácticas reducidos y que todos alumnos tengan oportunidad de trabajar. Los puestos del laboratorio que quedan libres son los de menor calidad y aquellos que se utilizarán como nodos adicionales de las máquinas virtuales de PVM.

En cuanto a sistema operativo, todos los puestos disponen de arranque en Windows y en Linux. En las prácticas de ACSO2 se utiliza sólo LINUX.

El laboratorio dispone, además de los puestos para los alumnos, de un servidor LINUX para docencia. En este servidor existe una cuenta para cada uno de los alumnos matriculados en las asignaturas que utilizan LINUX en el laboratorio. El servidor está conectado 24 horas para que los alumnos se puedan conectar y utilizarlo cuando quieran de forma remota.

La utilización de NFS ( Network File System) en todos los equipos permite que un alumno siempre acceda a la misma información, sea cual sea el puesto de trabajo que ocupe durante una sesión de prácticas. La información de cada usuario se encuentra almacenada en el servidor.

## **b) Alumnos**

Los alumnos de ACSO2 llegan a la asignatura con poca experiencia en programación. Esto significa que no se puede exigir grandes desarrollos software, limitando las prácticas a modificaciones en programas ya escritos por los profesores, ya sea de forma completa o parcial.

En cualquier caso, a la altura del curso en la que se realiza la práctica, los alumnos han realizado dos prácticas utilizando el lenguaje de programación C, unas 10 horas de laboratorio. El trabajo realizado en estas prácticas así como el interés mostrado de los alumnos resuelven en parte este problema.

Teniendo en cuenta estos condicionantes, se plantea la práctica de forma que aquellos alumnos con menos experiencia en programación puedan sacar partido de la misma, al mismo tiempo que los que si tienen experiencia se motiven.

## **4.- DESARROLLO**

Tomando en cuenta todas las consideraciones anteriores se decide:

1. Utilización de PVM como entorno de desarrollo de aplicaciones para NOW. Este entorno es de libre distribución y está disponible para las arquitecturas y sistemas operativos más comunes. Dispone así mismo de una amplia documentación.
2. Utilizar como problema base la implementación de un programa que dibuje las figuras de Mandelbrot. Se suministra a los alumnos una versión secuencial y otra preparada para su ejecución en PVM.
3. Separar en la aplicación la parte gráfica. De esta forma los alumnos no tendrán que enfrentarse a los problemas inherentes a la programación gráfica, complicada en cualquier sistema operativo.

Todas las aplicaciones suministradas al alumno imprimen en su salida estándar el tiempo de ejecución del algoritmo. Este valor está claramente influenciado por la carga en la máquina, pero es más sencillo de medir y entender por los alumnos que otros parámetros más exactos pero más difíciles de obtener: utilización de CPU por parte de los procesos que componen la aplicación, tiempo de sobrecarga introducido por PVM, tiempos de comunicaciones...

### a) Estructura aplicación base

A los alumnos se les proporciona una primera versión de la aplicación PVM completamente funcional. Con esta aplicación se realizarán tareas de monitorización de PVM y se enseña a los alumnos cuales son las llamadas fundamentales de la librería de una forma práctica.

La estructura de esta aplicación se puede observar en la Figura. Los elementos que componen la aplicación son:

- *Maestro*. Este proceso se encarga de inicializar el conjunto de tareas necesarias, recibir los resultados y decidir cuando finaliza la aplicación. Sólo existe un *Maestro* que se ejecuta en la misma máquina donde se encuentra el alumno.
- *Trabajador*. Proceso que calcula una parte del problema. El *Maestro* arranca  $N$  procesos de este tipo. Pueden ejecutarse en cualquier máquina de la NOW.
- *Dibujador*. Proceso que dibuja en pantalla las líneas ya calculadas. Cuando el *Maestro* detecta que se ha calculado completamente una línea se la manda a este proceso. Se ejecuta en la misma máquina que el *Maestro*.

La idea de esta primera versión es dividir la pantalla en tantas zonas como trabajadores, todas con el mismo tamaño. Estas zonas se componen de líneas horizontales consecutivas.

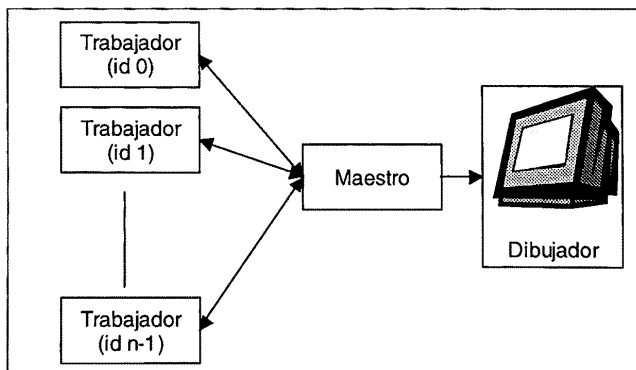


Figura 1: Estructura de la aplicación base

### b) Ejercicios guiados.

El primer ejercicio consiste en ejecutar la versión secuencial del algoritmo de Mandelbrot (sin PVM). Variando la resolución y el número de iteraciones del algoritmo se obtienen diferentes tiempos de ejecución. Estos tiempos se tomarán como base para los siguientes ejercicios.

Esta primera versión carece de salida gráfica para que los alumnos puedan comprender mejor su funcionamiento. Para dar una idea del resultado existen multitud de programas que dibujan el conjunto de Mandelbrot, entre ellos *kfract* instalado en las máquinas del laboratorio.

El siguiente ejercicio consiste en la ejecución de la versión PVM del algoritmo. Para ello, se enseña al alumno como poner en marcha PVM y la forma de añadir y quitar máquinas de la misma, es decir, los mecanismos para determinar y modificar la configuración de la máquina paralela virtual.

Tras esto se prueba la salida gráfica del algoritmo para que el alumno pueda apreciar como las líneas se calculan y dibujan de forma "paralela". Esta salida gráfica no se deberá realizar a la hora de tomar tiempos para poder comparar mejor los tiempos debidos al algoritmo de Mandelbrot.

El siguiente ejercicio es comparar los tiempos de ejecución de esta versión con la versión secuencial, modificando el número de trabajadores y el número de máquinas que componen PVM con la que está trabajando.

De esta forma se demuestra al alumno dos cosas:

1. Que el algoritmo secuencial funciona mejor cuando sólo hay un procesador.
2. Que la existencia de  $N$  procesadores no significa una mejora de  $N$  en el tiempo de ejecución, pero que existe una mejora.

Así mismo, se invita al alumno a probar la consola XPVM. Con esta aplicación se puede obtener una traza de las llamadas a la máquina virtual y observar el intercambio de mensajes entre los componentes de la aplicación.

Estos mecanismos de monitorización serán útiles en el desarrollo de las soluciones propias de los alumnos, ya que permite descubrir con mayor facilidad en que parte del código se encuentran los programas al presentar errores.

Como comentario adicional se muestra como la consola XPVM consume parte de los recursos de la máquina, disminuyendo la velocidad de la ejecución efectiva de nuestra solución. Con esta reflexión deben decidir si dejar la consola en marcha o pararla antes de tomar tiempos.

### c) Ejercicio propuesto

El ejercicio propuesto es la modificación de la aplicación suministrada para mejorar el tiempo total de la aplicación. Para que los alumnos entiendan por donde pueden introducir mejoras se les indican los siguientes fallos en la aplicación base:

1. Todos los procesadores calculan el mismo número de líneas.
2. No todas las líneas tienen el mismo coste computacional. En la figura suministrada las líneas más "caras" se encuentran en el medio, por lo que algunos Trabajadores tienen que hacer más trabajo.
3. No todos los ordenadores tiene las mismas prestaciones.

Éstos no son los únicos aspectos mejorables pero son los que creemos que se entienden mejor por los alumnos. Con estas dos pistas, las soluciones planteadas por los alumnos se basan en una combinación de las siguientes estrategias:

1. Que la asignación de líneas no haga que aquellas zonas más caras sean asignadas a un único Trabajador.
2. Agrupar mayor cantidad de información en los mensajes que los trabajadores intercambian con el servidor.
3. Que el reparto de las líneas no sea estático. De esta forma un Trabajador le pide líneas al Maestro y este le informa de la siguiente línea o grupo de líneas a calcular.

La forma de evaluación de estos ejercicios es una competición. Se informa de una fecha tope para preparar la aplicación y de la configuración que se utilizará. Llegada la fecha se ejecutan todas las soluciones planteadas por los alumnos en una PVM, aquel grupo cuya solución obtenga mejores tiempos obtendrá la mejor nota.

Los grupos tienen asegurado el aprobado siempre que la solución funcione.

## **5.-PROBLEMAS DETECTADOS**

El año de implantación de las prácticas es el año en el que mayor número de problemas se detectan en la misma, incluso en aquellas prácticas con gran tiempo de dedicación en su preparación.

Surgen normalmente dos tipos de problemas: aquellos relacionados con las suposiciones que se hacen de antemano sobre el entorno y aquellos problemas en los que no se había pensado.

En el caso de la presente práctica han aparecido ambos tipos de problemas. Aunque creemos que ninguno de ellos es insalvable ni ha influido en los objetivos principales de la práctica si que ha podido desanimar a los alumnos en algunos momentos.

### **a) Entorno**

El principal problema debido al entorno ha sido imposibilidad de montar a tiempo un "cluster" independiente de máquinas Linux que se dedicara exclusivamente a la ejecución de las aplicaciones de los alumnos. Esto ha provocado que los alumnos tuvieran que utilizar como nodos de sus PVM, los computadores que otros compañeros estaban utilizando para la edición de trabajos. Este hecho provocaba una gran variabilidad de los resultados obtenidos con la versión paralela del algoritmo.

### **b) Problemas nuevos**

El principal problema ha sido no establecer de antemano las características de la PVM donde se probarían las soluciones de los alumnos. Este hecho provocaba en los alumnos cierta ansiedad sobre la hondad de la solución en la que estaban trabajando.

Por otra parte el tiempo que los alumnos han dedicado a la práctica ha sido mucho mayor del esperado. El que los alumnos dedicasen más tiempo del esperado se debe a dos factores independientes: la poca experiencia de programación en C y el afán, a veces obsesión, de superación de la solución obtenida.

El problema de la experiencia de programación en C es difícil de superar, aunque se mantienen conversaciones con profesores de asignaturas de programación para que se solucione año a año.

El problema del tiempo dedicado por intentar mejorar la solución se debe en parte a la forma propuesta de evaluación y la inexistencia de tiempos de referencia que indicasen la bondad de la solución. Este hecho podría tener solución con una tabla de tiempos de referencia y un conjunto de máquinas dedicadas de forma exclusiva a la ejecución de las soluciones de los alumnos, problema muy relacionado con el entorno.

## **6.-CONCLUSIONES**

Con el trabajo realizado se ha desarrollado una práctica que ha permitido al alumno conocer una herramienta de trabajo con redes de estaciones de trabajo. Así mismo el alumno ha resuelto un problema sencillos construyendo una aplicación paralela.

El alumno ha realizado variantes de los programas estudiando los parámetros que influyen en las prestaciones de una NOW.

La práctica puede adaptarse fácilmente a otras asignaturas que toquen el tema de las redes de estaciones de trabajo. Por ejemplo, en la unidad docente de Arquitectura de Computadores, se podría adaptar a la asignatura Sistemas Multiprocesadores, optativa de 9º cuatrimestre de la Facultad de Informática de Valencia.

En general se han cumplido los objetivos, por lo que junto con la motivación que se ha observado en los alumnos, nos permite calificar la experiencia de satisfactoria.

## **7.-BIBLIOGRAFIA**

- [1] "Prácticas con el Simulador de Multicomputadores". José Duato y Pedro Lopez. 1993.
- [2] Página principal del proyecto PVM: <http://www.epm.ornl.gov/pvm/>
- [3] A.Geist et. al. "PVM3 User's Guide and reference Manual" Septiembre 1994.
- [4] "Parallel Programming Languages and Environments" Ira Pramanick. 2º Capítulo de "High Performance Cluster Computing" Editor Rajkumar Buyya. Editorial Prentice Hall.
- [5] "MPI and PVM programming" Ira Pramanick. 3º Capítulo de "High Performance Cluster Computing" Editor Rajkumar Buyya Editorial Prentice Hall.