**Reto 4 Tópicos Especiales en Telemática**

Sebastián García Valencia

Estudiante Ingeniería de Sistemas

Universidad EAFIT

Medellín, Colombia, Suramérica

**Resumen**

El presente documento contiene el análisis y diseño de un programa para vectores de movimiento usando mpi con c.

**Palabras clave.**

Mpi, c, vectores de movimiento, programacion paralela

1. **Introducción**

Los problemas actualmente, ya sean científicos o comerciales, cada vez necesitan mas procesamiento, ya sea en datos en instrucciones o en ambos. Por esto la programación paralela es cada ves mas importante, en el presente programa se pretende usar mpi para solucionar un problema de programación intensiva en procesamiento sobre vectores de movimiento, estrategia usada en la compresión mpeg.

1. **Algoritmo secuencial**

Antes de iniciar se debe tener un punto de partida, al hacer este algoritmo secuencialmente para un conjunto de datos de 10000 en las matrices y de 100 en los macrobloques se obtuvo un tiempo de 239.2230 segundos.

Para 1000/100

Secuencial 2.420304

1. **PCAM**
   1. **Particionamiento**

A cada tarea se enviara primero toda la matriz de destino, y luego por medio de un round robin se les ira rotando cada macrobloque de la matriz original, cada tarea procesara y devolverá el índice y mejor match de la determinada posición de la matriz de referencias.

* 1. **Comunicación**

Se tendrá una tarea master encargada de llenar las matrices y esta le enviara a los n trabajadores los datos, el control se dara enviando primero el índice de la matriz de referencias que se esta trabajando, la comunicación será solo de 2 niveles y se dara

Master a slaves: 1 a n

Slaves a master: 1 a 1

* 1. **Aglomeracion**

Por el momento no se tendrán submaster que distribuyan los datos

* 1. **Mapeo**

Se tienen 4 nodos, uno será el master (para este caso no se pondrá a procesar) y en los otros 3 se tendrán los n trabajadores repartidos, se dejara que mpi gestione los procesos muoltiples en cada nodo y por ahora será transparente para nosotros.

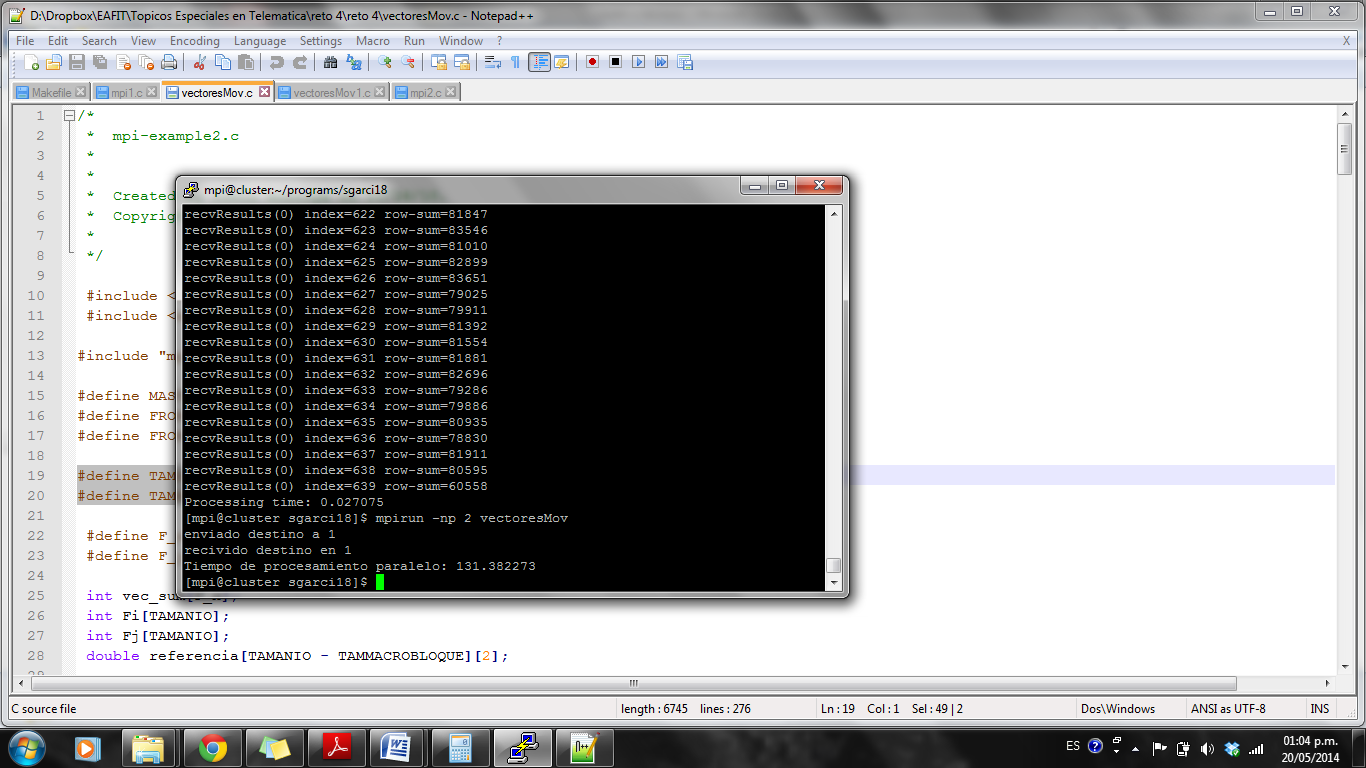
1. **Análisis de tiempos.**

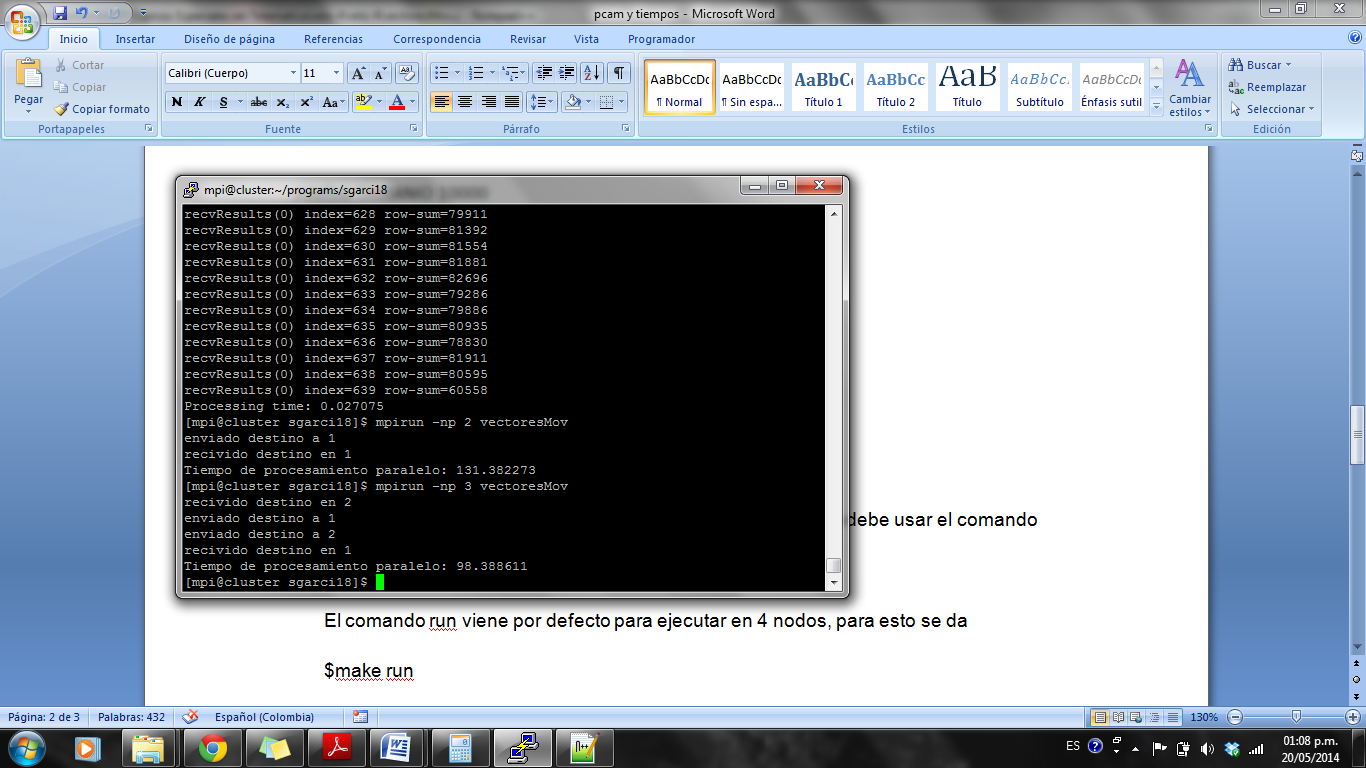
Se tomo como base matrices de 10000 posiciones, lo cual para frames de un video es un poco consrvador pero por cuestiones de tiempo fue suficiente, el macrobloque tiene un tamaño de 100

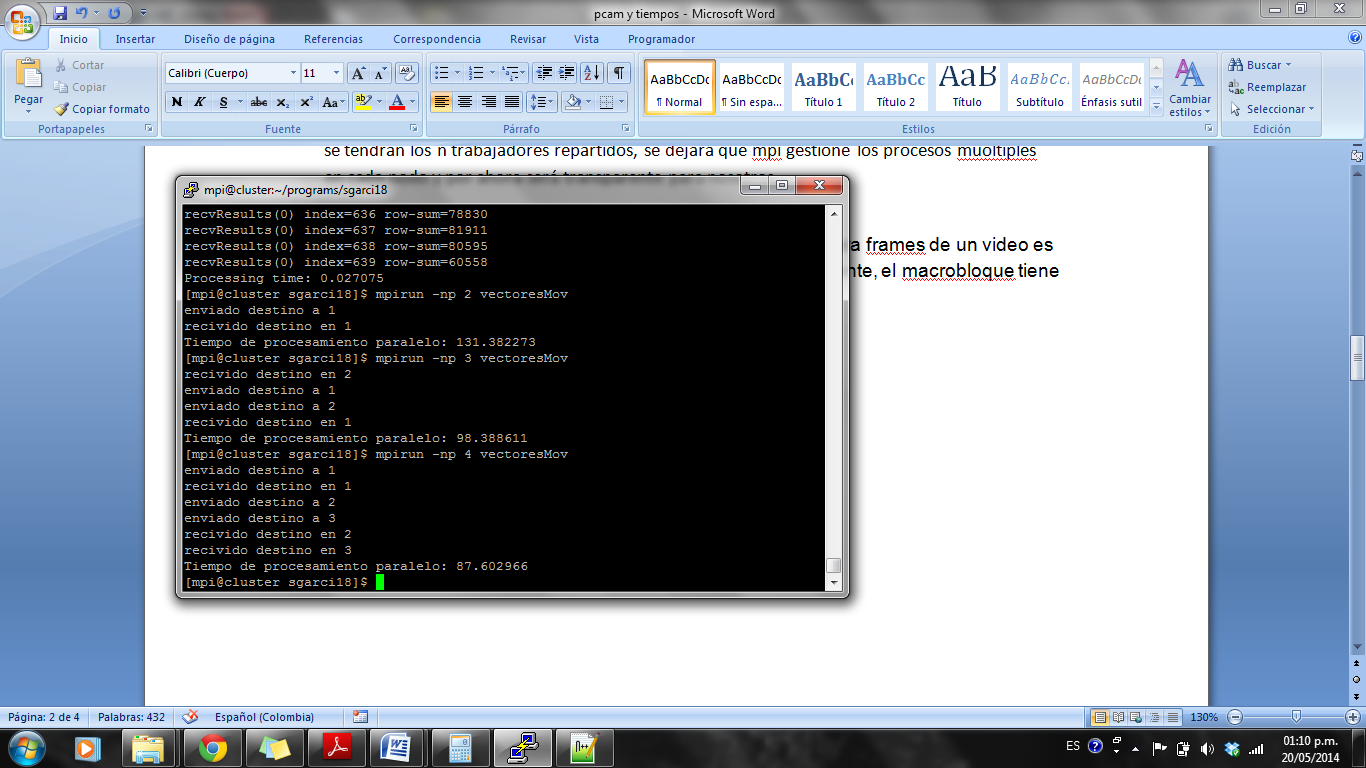
#define TAMANIO 10000

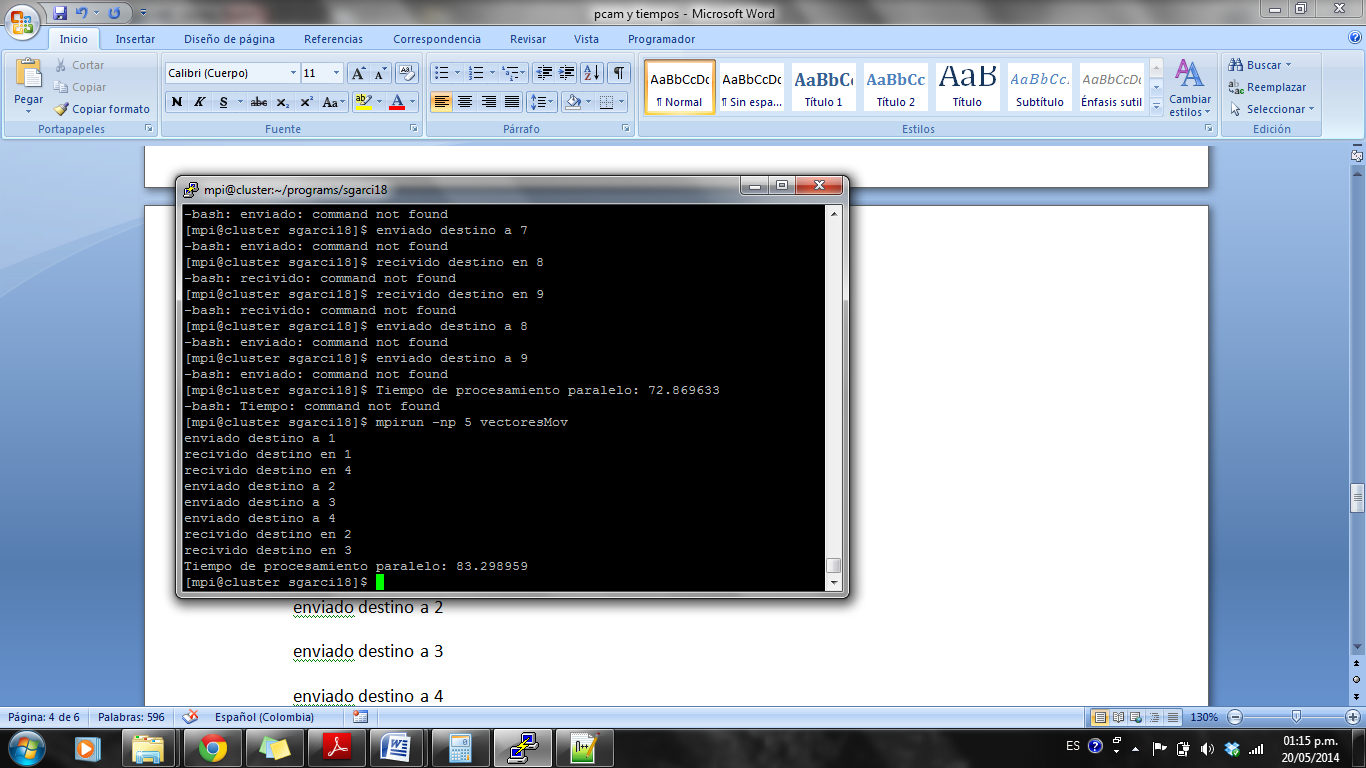
#define TAMMACROBLOQUE 100

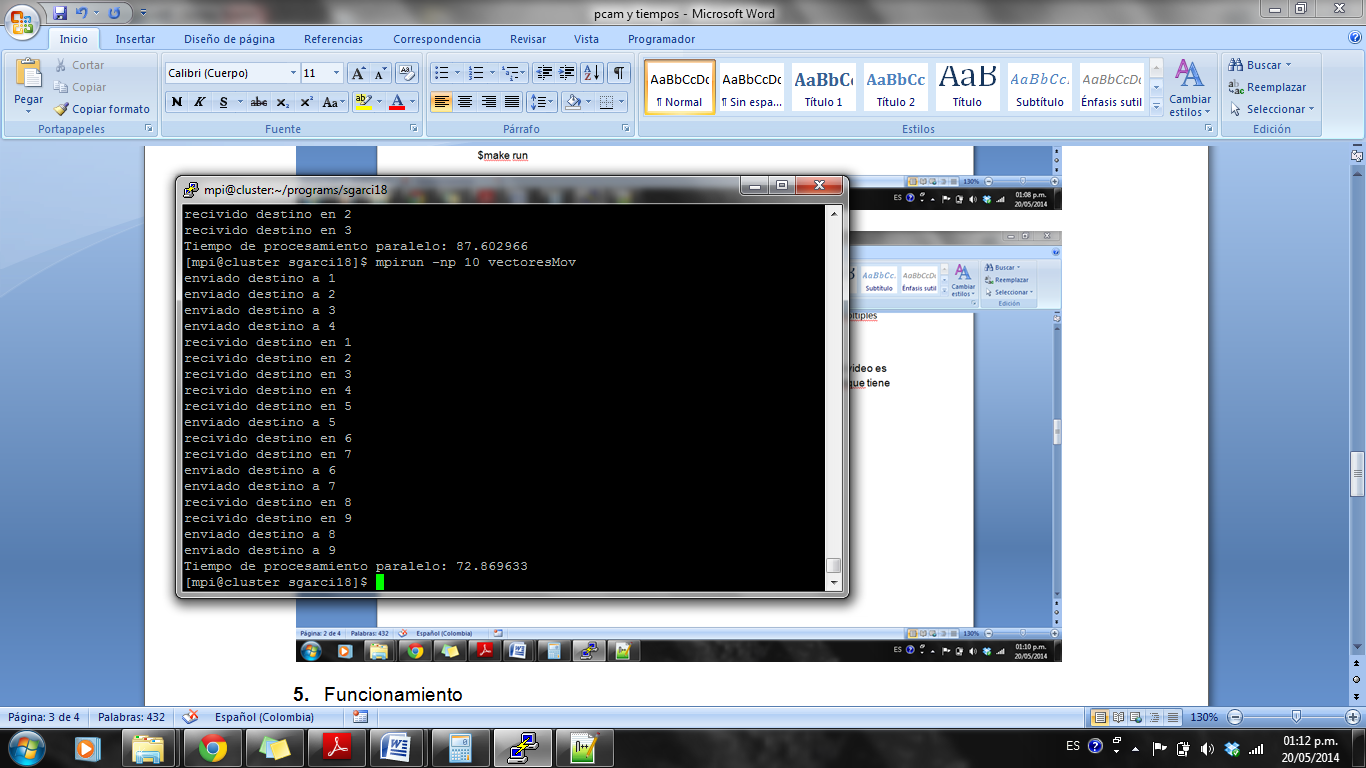
|  |  |
| --- | --- |
| num nodos | t ejecucion |
| 2 | 131,382273 |
| 3 | 98,388611 |
| 4 | 87,602966 |
| 5 | 83,298959 |
| 10 | 72,869633 |
| 100 | 66,886327 |

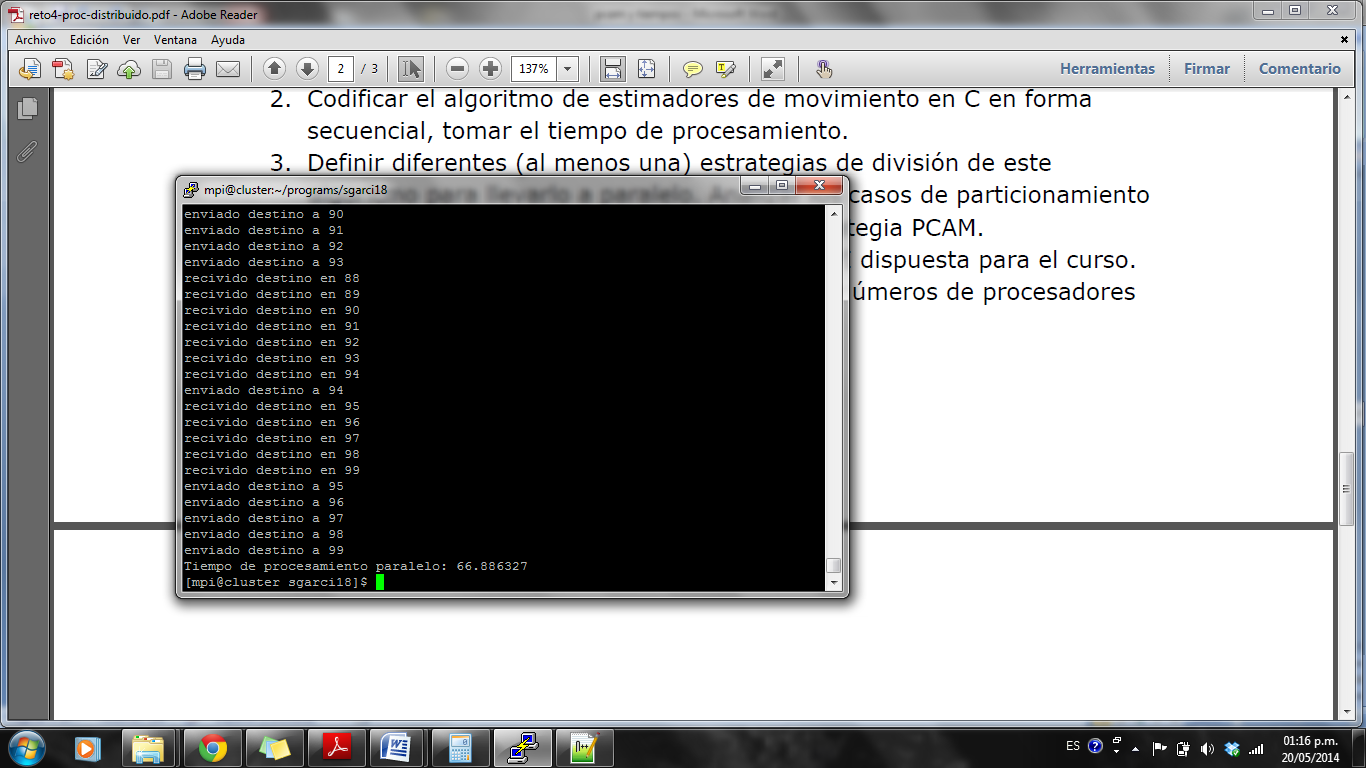












1. **Funcionamiento**

Para facilidad de la ejecución se tiene un makefile

Para compilar, una ves ubicado en la carpeta del proyecto se debe usar el comando

$make

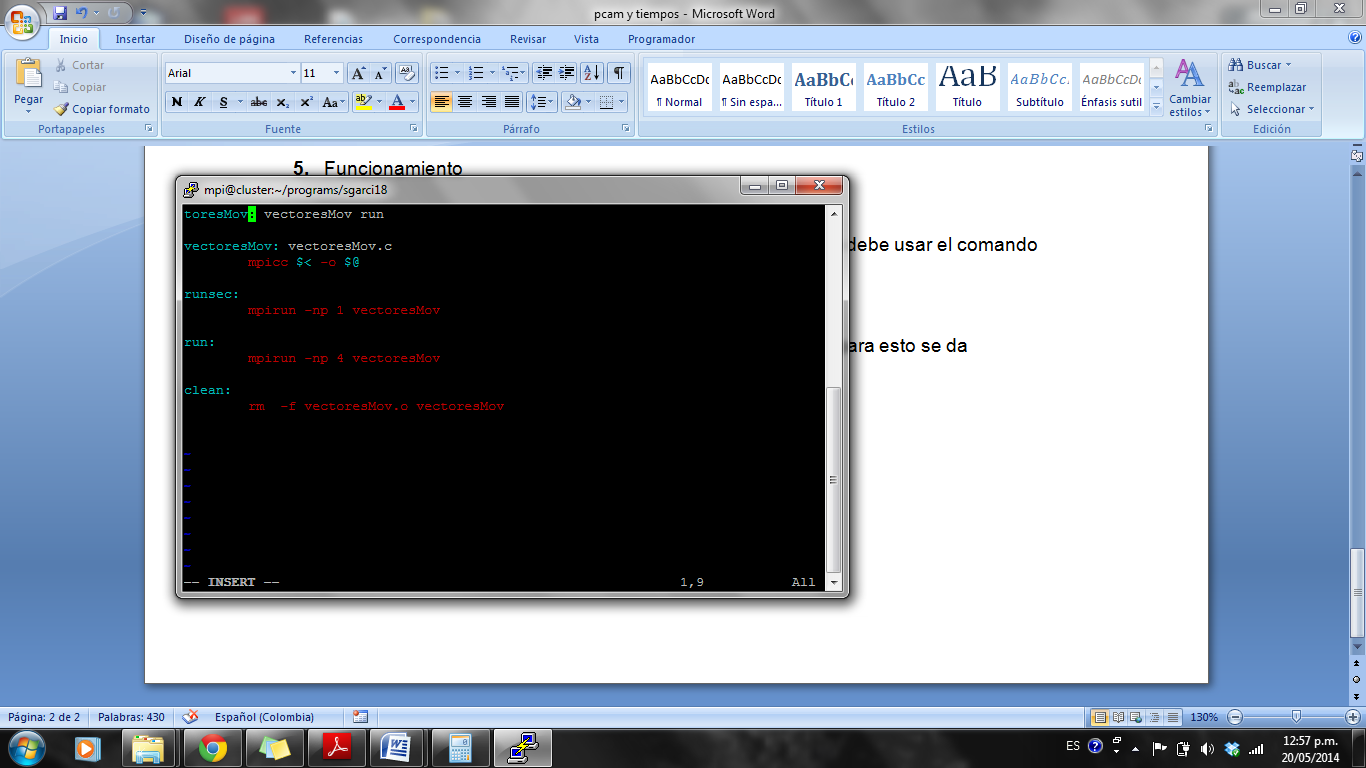
El comando run viene por defecto para ejecutar en 4 nodos, para esto se da

$make run

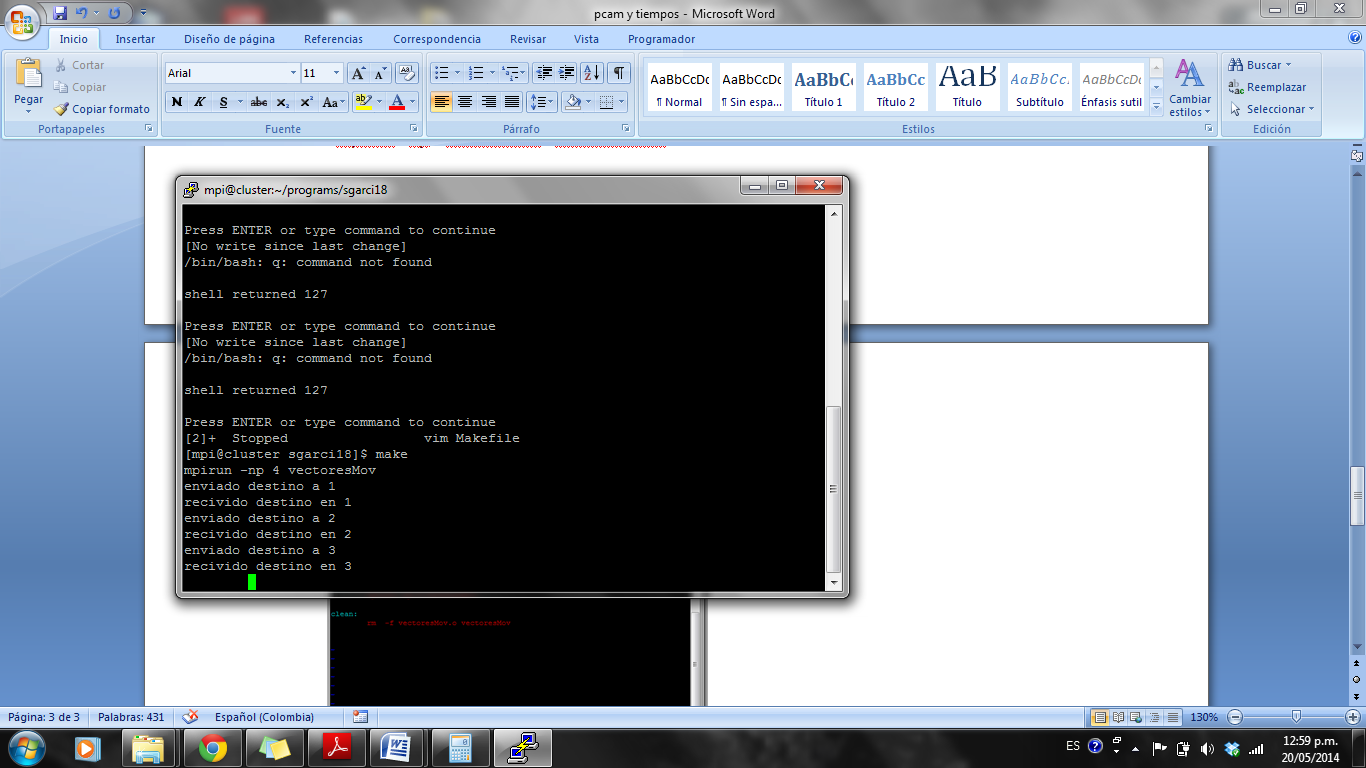
Para ejecutar en otro numero de nodos se usa

$mpirun –np <numNodos> vectoresMov

makefile



ejecucion



1. **Posibles mejoras**

* En el momento el master no hace ejecución, dado que este es un nodo completa podría aprovecharse también

1. **Enlaces**

Repositorio GIT: <https://bitbucket.org/sebasgverde/reto4toptelematica>

1. **Bibliografía**

<http://en.wikipedia.org/wiki/Motion_vector>

<http://hpcc.usc.edu/support/documentation/examples-of-mpi-programs/>