Modelando la Performance de Aplicaciones en un Clusters de PC's

Villalobos M., Flores S., Piccoli F., Printista M. *
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 - San Luis
Argentina
e-mail: {mavi, sflores, mpiccoli, mprinti}@unsl.edu.ar

1 Introducción

Un *cluster* de computadoras es una máquina que consiste de un número de workstations o PC's de bajo costo interconectados por una red para actuar como un único recurso de computación.

Los multiprocesadores simétricos, *SMP*, corrientemente tienen mejor relación precio-performance que una única workstation y por lo tanto son muy atractivos para ser incorporados en un cluster.

Resolver problemas en máquinas paralelas introduce intercambio de datos. El volúmen de datos intercambiados crece con el número de procesadores intervinientes en la aplicación. Por lo cual, para construir clusters escalables, la capacidad de la red debe escalar con el número de nodos en el cluster.

A medida que el *cluster* ha ganado popularidad como plataforma adecuada para computaciones paralelas, ha surgido la necesidad de predecir la performance para el análisis de aplicaciones escalables que corren sobre dicha plataforma. La estimación de la performance facilita actividades tales como mejorar el rendimiento (*tuning*) de las aplicciones, selección de las mejores técnicas de implementación y comparación de la performance entre distintas plataformas de multicomputadoras.

La performance de un multicomputador es una función de la performance de los procesadores o *SMP* utilizados, pero el *tiempo de pasaje de mensajes* es el tiempo más influyente en aplicaciones con pasaje de mensajes de gran volúmen. Estimar este tiempo es un problema complejo. El tiempo depende de las características de los procesadores, de la red de interconexión, de la librería de pasaje de mensajes utilizada, de los patrones de comunicación, de los tamaños de los mensajes, del número de nodos y de la distancia entre los nodos de procesadores.

El objetivo de este proyecto es analizar y desarrollar un modelo que caracterice la performance de aplicaciones paralelas en un cluster de PC's. En particular, nuestros desarrollos se estan realizando en un cluster de 14 *SMP* (duales), conectadas mediante una *Fast Ethernet*. Este cluster pertenece a la Unidad de Servicio Computacional (para aplicaciones científicas) del Instituto de Matemática Aplicada San Luis (IMASL), de la U.N.S.L.

*Grupo subvencionado por la UNSL y ANPCYT (Agencia Nacional para Promover la Ciencia y Tecnología)

2 Modelo de Computación

Para modelar los costos de Computación, nos basamos en medidas que son dependientes de la aplicación. Este modelo propone un límite superior para la performance que puede ser alcanzado por una aplicación particular sobre ciertos procesadores. Dicho límite es encontrado contando las operaciones escenciales en la aplicación. El inconveniente radica en que algunas operaciones tienen diferencias significativas en sus costos. Por ejemplo una iteración que opera sobre un arreglo de números enteros es significativamente más costosa que una iteración que opera sobre un arreglo de números de punto flotante. Basándonos en ésto, trabajamos sobre "bloques conceptuales de computación". Para el análisis de performance nos concentramos en estos bloques asignándoles un factor de costo. Cada bloque conceptual, Γ^j, tiene un conjunto de instrucciones asociadas denominado Υ_{Γ^j}. Notar que en base al número de veces que cada operación de Υ_{Γ^j} se ejecuta se pueden formar k subconjuntos que cumplen:

$$\bigcup_{i=1}^{k} \Upsilon_{\Gamma^j{}_i} \quad = \quad \Upsilon_{\Gamma^j} \quad \text{y} \quad \bigcap_{i=1}^{k} \Upsilon_{\Gamma^j{}_i} \quad = \quad \emptyset \tag{1}$$

Asociando un costo a cada subconjunto, se puede expresar el tiempo de computación insumido en cada bloque conceptual como:

$$T^j{}_{comp} \quad = \quad \sum_{i=1}^{k} C_i * N_{\Gamma^j{}_i} \tag{2}$$

donde C_i es el factor de costo del subconjunto $\Gamma^j{}_i$ y $N_{\Gamma^j{}_i}$ es el número de veces que las operaciones del subconjunto se ejecutan durante el programa. La expresión final para el tiempo de computación insumido durante la ejecución del programa se obtiene por la composición del tiempo de todos los bloques:

$$T_{comp}(p) \quad = \quad \sum_{j=1}^{n_{bcomp}} T^j{}_{comp} \tag{3}$$

3 Modelo de Comunicación

El enfoque presentado aquí se basa en medir y modelar la performance de un conjunto de patrones básicos de comunicación. Estos patrones fueron seleccionados de forma tal que otros patrones de comunicación más complejos puedan ser construidos desde los patrones básicos analizados. Se ha utilizado como librería de comunicación el estándar *MPI* [4], ya que sus primitivos de comunicación cubren la totalidad de primitivos de intercambio de datos globales y de primitivos de sincronización.

Similar al esquema de bloques conceptuales utilizado en las computaciones, para el análisis de las comunicaciones nos concentramos en cada comunicación en particular (la que será considerada un bloque conceptual de comunicación).

Para el cálculo de los tiempos de cada patrón de comunicación se realizaron repetidamente diferentes experimentos de los algoritmos que los implementan. En cada experimento se varía la longitud de los mensajes y el número de procesadores intervinientes. Se utilizó como medida estándar el *RoundTripTime* de la red [1] y los patrones de comunicación básicos considerados fueron: *Point-to-Point*, *Exchange*, *One-to-Many*, *Many-to-One*. Para desarrollar las fórmulas que den el tiempo como una funcion del número de procesadores (p), los tiempos muestrales obtenidos serán analizados mediante la Técnicas de Ajuste de Multivariables [3], Para modelar el tiempo de comunicación de cada boque conceptual nos basamos en dos componentes: *tiempo de setup* de la red y *tiempo de transferencia*. En tiempo de comunicación para un bloque j, se puede expresar con la siguiente fórmula:

$$T^j{}_{comm}(p) \quad = \quad t_{comm}(p) + tt_{comm}(p) * longmens \tag{4}$$

Donde t_{comm} y tt_{comm} corresponden al tiempo de setup y de tranferencia respectivamente. Estos tiempos deberán ser instanciados al patrón de comunicación particular involucrado en cada bloque conceptual de comunicación.

De lo anterior resulta que el tiempo total insumido en comunicaciones durante la ejecución de una aplicación paralela se obtiene por la siguiente fórmula:

$$T_{comm}(p) = \sum_{j=1}^{n_{bcomm}} T^j{}_{comm}(p) \quad (5)$$

4 Integración del Modelo

En nuestro enfoque para modelar la performance del cluster se han ejecutado dos aplicaciones intensivas de pasaje de mensajes: un algoritmo pipeline que resuelve el problema de los N-cuerpos en el espacio [2] [5] y el algoritmo de Fox [4] para resolver la multiplicación de matrices. El tiempo de ejecución de estas aplicación es la suma de sus tiempos de computación y de comunicación. El límite superior para el tiempo de computación se encuentra cronometrando los "bloques conceptuales de computación"existente en el código fuente de las aplicaciones secuencializado (con cantidad de procesadores $NP = 1$), como especifica el modelo en (2) y (3). El tiempo de comunicación se calcula identificando las llamadas explícitas a las rutinas de pasaje de mensajes de MPI. Luego, se debe relacionar estas llamadas a un patrón básico de comunicación y finalmente de aplica el modelo de comunicación dado en (4) y (5).

5 Conclusiones y Próximos Pasos

El resultado del trabajo preliminar realizado, prueba la utilidad del modelo para ajustar aplicaciones de pasaje de mensajes, permitiendo seleccionar las mejores técnicas de implementación.

Es importante mencionar que el proyecto se dividió en las 3 etapas siguientes:

- Definición del Modelo de Performance
- Análisis de Performance de los patrones básicos de Comunicación
- Análisis y Predicción de la Performance

Este reporte de investigación incluye los resultados de la primera etapa del proyecto. La segunda etapa está en desarrollo. Una vez finalizada, quedar a definido el conjunto de parámetros claves que determinan la performance de comunicación del cluster especificado. Con esta información disponible, será posible realizar predicciones basadas en el modelo analítico presentado en este reporte.

El objetivo último, es que los parámetros obtenidos del modelo resulten lo suficientemente robustos para realizar la comparación de la performance entre distintas plataformas de multicomputadoras.

Referencias

[1] Abandah G. and Davinson E.S. *Modeling the Communication Performance of the IBM SP"*, (10th. International Parallel Processing Symposium, 1996).
[2] Marciniak, A. *Numerical solutions of the N-body Problem* (D. Reidel Publishing Co., Dordrecht, 1985).
[3] Mendenhall W., Wackerly D. D. and Scheaffer R.L. *Mathematical Statistic with Applications*, (Duxbury Press, 1989).
[4] Snir, M., Otto, S., Huss-Lederman, S., Walker, D., Dongarra, J. *MPI: The complete Reference* (Cambridge, MIT Press, 1996).

[5] Wilkinson B. and Allen M. *Parallel Programming: Techniques and Application using Networked Workstations and Parallel Computers* (Prentice-Hall, 1999).