

Proyecto

“Estudio de Modelos Teóricos de tipo NP-completos en el Laboratorio Nacional de GRIDS de Súper Cómputo, Utilizando Algoritmos Evolutivos de Optimización con Técnicas de Procesamiento Distribuido”

REPORTE DE AVANCES DEL PRIMER SEMESTRE

Integrantes del proyecto

Universidad Autónoma del Estado de Morelos	Instituto Tecnológico de Veracruz
Cuerpo Académico: Optimización y Software Integrantes: Investigador Líder Dr. Marco Antonio Cruz Chávez M.C. Martín Heriberto Cruz Rosales Dr. José Crispín Zavala Díaz Alumnos: M.C. Fredy Juárez Pérez L.I. Alina Martínez Oropeza	Cuerpo Académico: Cómputo Intensivo Aplicado a la Ingeniería Integrantes: Investigador Principal Dr. Abelardo Rodríguez León M.C. Rafael Rivera López Alumnos: Tania I. Araus Pérez Marcela González Tinoco

Introducción

A la fecha se ha realizado poco trabajo para tratar problemas de optimización de tipo NP-duro de tamaño grande utilizando grids. Generalmente este tipo de problemas de optimización se tratan en equipos que cuentan con unos cuantos CPU's. Estos problemas por ser de gran complejidad requieren primero del diseño de algoritmos eficientes/eficaces y después de la ejecución de estos algoritmos en equipos de cómputo con gran poder de procesamiento como lo pueden ser las grids. Fujisawa, fue uno de los primeros pioneros en el estudio de problemas de tipo NP-duros en sistemas grid [Fujisawa, et al., 2004]. Los problemas abordados fueron problemas de optimización cuadrática aplicando un método de relajación convexa sucesiva con implementación en paralelo. El sistema grid utilizado fue compuesto de dos clusters, el primero localizado en Tokio, el cual estaba compuesto de cuatro núcleos Pentium III Xeon a 700 Mhz; el otro cluster se encontraba en Kioto, el cual

estaba compuesto de cuatro núcleos Pentium III Xeon a 850 Mhz. Esta pequeña grid tenía una conectividad por Internet de 2 a 4 Mbs por segundo y una LAN de 100Mbs. En esta investigación se sugiere que el tiempo de transferencia de datos debe ser el menor posible (menos de un segundo) entre nodos de clusters, debido a la latencia existente para que se afirme que el algoritmo está bien paralelizado para un ambiente grid. Zunino [Zunino, et al., 2007], presenta un procedimiento para escribir algoritmos exactos distribuidos de optimización orientados a objetos sobre un ambiente grid para modelos con función multiobjetivo. Su procedimiento incluye un subprocedimiento para reducir el espacio de búsqueda del problema. Utilizan un paradigma de esclavo-maestro para la paralelización de las unidades de las tareas y utiliza un algoritmo de ramificación y acotamiento como el subprocedimiento. Codifican los diferentes objetos del algoritmo para generar una comunicación pequeña entre nodos. Trabaja con problemas Np-duros de calendarización como es el flowshop. Para la paralelización del algoritmo utilizan las librerías MPI debido a que éstas pueden ser utilizadas en clusters y en ambientes grid. Nebro propone un algoritmo genético para resolver el problema de ensamble del DNA en una gris [Nebro, et al., 2008]. Este es un problema de optimización combinatoria NP-hard. Ellos trabajan con 150 computadoras en un sistema grid y logran reducir el tiempo de obtención de una solución aceptable obtenida originalmente en 10 días a un tiempo de unas cuantas horas. Propone trabajar sobre nuevas metaheurísticas bajo ambiente grid como una nueva clase de algoritmos para ser aplicados a problemas que requieren de un gran esfuerzo computacional. La presente investigación busca tratar con problemas de tipo NP-hard los cuales no se han probado en ambientes gris y con algoritmos genéticos que aportan una estructura adecuada para su paralelización, ambos conceptos, problema y algoritmo son un área de oportunidad para su estudio en ambientes grid.

Los trabajos que se han realizado para mejorar el servicio en grids con el objetivo de ofrecer un servicio eficiente de cómputo de alto desempeño son investigaciones en mejoras de algoritmos de búsqueda a larga escala en grids. Para la investigación que se realiza en el presente proyecto es importante conocer estos trabajos que se desarrollan para perfeccionar el servicio de cómputo de alto desempeño, con el fin de poder aprovechar al máximo las características de la arquitectura grid en el uso de cómputo de alto desempeño. El middleware tiene que proporcionar un servicio de localización distribuida o un servicio de búsqueda para encontrar servidores disponibles, para tipos de aplicaciones y evaluar el desempeño de aplicaciones en esos servidores [Philippe et al., 2006]. Se han analizado las métricas de desempeño de grids para cómputo de alto desempeño en áreas científicas, considerando a las métricas como tiempo de respuesta de utilización del sistema grid. En grid computacional se ha realizado poca investigación en evaluación de desempeño, existen Benchmarks que se pueden utilizar para realizar estas pruebas de desempeño [Peng, et al., 2004]. Se han realizado estudios de problemas en la ejecución de la grid debido a fallas de la red, condiciones de recursos sobrecargados, o no disponibilidad de componentes de software requeridos. Los sistemas de tolerancia a fallas son capaces de identificar y manipular fallas y soportar una ejecución confiable en la presencia de concurrencia y fallas. En la calendarización de un número grande de trabajos de usuarios para la ejecución en paralela sobre un sistema de grid, los trabajos son sujetos a fallas de sistemas o demoras causadas por hardware infectado, vulnerabilidad de Software y políticas de seguridad no seguras. Priya, propone técnicas de tareas que enmascaran los efectos de la ejecución de falla de tareas. Estas técnicas se controlan por medio de un algoritmo genético para la calendarización de tareas. Por medio de este algoritmo se logra recuperar los sistemas grid

utilizando puntos de control del estado del sistema [Priya et al., 2007]. Las técnicas de calendarización de una grid son más complejas que las técnicas de calendarización aplicadas a sistemas de cómputo de alto desempeño y la calendarización de la grid es uno de los mayores factores que afectan el desempeño de aplicaciones en grid [Huang et al., 2007]. Los métodos de calendarización en grid requieren de considerar las características de los servicios de grid, tales como disponibilidad de servicio, acceso, fiabilidad, rendimiento, tasa e transferencia y otros [Chin et al., 2006].

Objetivo general

Desarrollo de algoritmos evolutivos en plataforma Grid para problemas NP-hard aplicando el estándar MPI

Objetivos particulares

Objetivo 1.- Desarrollo y análisis de una estructura de vecindad con procesamiento distribuido para ser implantada en algoritmos evolutivos.

Objetivo 2.- Desarrollo de un algoritmo evolutivo híbrido con procesamiento distribuido y que aplique búsquedas locales también con procesamiento distribuido para la optimización de problemas de manufactura y que pueda resolver modelos de estudio tomados de la literatura.

Objetivo 3.- Desarrollo de un algoritmo evolutivo híbrido con procesamiento distribuido que aplique también búsquedas locales con procesamiento distribuido para la optimización de una cadena de suministro y que pueda resolver modelos de estudio tomados de la literatura.

Actividades desarrolladas

De acuerdo al plan de trabajo, se realizaron las siguientes actividades:

1. Estado del arte.
 - a. Estudio de los modelos NP-hard, para la asignación de recursos en el transporte con ventanas de tiempo y el problema de manufactura.
 - b. Estudio del estado del arte de estructuras de vecindad para problemas de optimización discreta.
 - c. Estudio del estado del arte de estructuras de vecindad para VRPTW y problemas de manufactura.
 - d. Estudio del estado del arte de Algoritmos evolutivos para VRPTW y problemas de manufactura.
 - e. Análisis de ventajas y desventajas que ofrecen las estructuras de vecindad y algoritmos evolutivos para VRPTW.

- f. Propuesta de una hipótesis que mejore la funcionalidad de un algoritmo evolutivo en base a una hibridación con una estructura de vecindad propuesta en base a un conocimiento previo, tomando como base para esto una estructura del algoritmo que trabaje con procesos distribuidos. De acuerdo a una revisión de trabajos previos para evitar que la latencia de la grid sea un factor muy grande de peso en la ejecución de los algoritmos evolutivos, se decidió que la estructura de procesos a generar en el algoritmo en paralelo con MPI, es la de trabajar con conjuntos de poblaciones, donde subconjuntos de estas poblaciones se puedan ejecutar de forma independiente en cada nodo de la grid que se tenga disponible. La comunicación la tendrá un nodo maestro para evaluar por selección, que poblaciones son las que tienen una mejor aptitud en cada nodo de la grid y trabajar con estas para que en un tiempo definido en base a las pruebas, se repliquen estos resultados en cada nodo para efectuar las operaciones principales del híbrido genético que son cruzamiento y búsqueda local. Estas operaciones se harán de forma independiente en cada nodo de la grid.
2. Desarrollo del Algoritmo evolutivo secuencial para una cadena de suministros con ventanas de tiempo. El modelo teórico del problema VRPTW se puede representar como un modelo de programación entera con una función multiobjetivo y un conjunto de restricciones:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} X_{ijk} \quad (1)$$

Sujeto a:

$$a_i \sum_{j \in \Delta^+(i)} X_{ijk} \leq w_{ik} \leq b_i \sum_{j \in \Delta^+(i)} X_{ijk} \quad \forall k \in K, i \in N \quad (2)$$

El algoritmo evolutivo evalúa la aptitud de cada individuo con la función multiobjetivo presentada en la ecuación (1), la función multiobjetivo representa la demanda y el tiempo de atención a cada cliente tratando de minimizar el costo implicado por estos dos objetivos. En la ecuación (1), c representa el costo de transportar del origen i al destino j , X representa el viaje del origen i al destino j en un vehículo k . Lo que se busca es el mínimo número de vehículos asignados a cada viaje mientras las restricciones de capacidad de cada vehículo y las ventanas de tiempo de atención a clientes se respetan. Se busca la distancia total más corta recorrida por el total de vehículos participantes. En la ecuación (2), w_{ik} representa la ventana de tiempo para el nodo i atendido por el vehículo k . La ventana es el tiempo que se da al vehículo para atender al cliente dentro de un intervalo de tiempo $[a_i, b_i]$, donde a_i representa el tiempo mínimo para llegar al nodo i y b_i representa el tiempo de salida del vehículo del nodo i el cual no debe de excederse.

3. Desarrollo del Algoritmo evolutivo secuencial para un taller de manufactura. El modelo teórico del problema de calendarización en manufactura es un modelo de programación entera mixta con una función objetivo y un conjunto de restricciones:

$$\text{Min} \left[\max_{j \in O} (s_j + p_j) \right] \quad (3)$$

$$\forall j \in O \quad s_j \geq 0 \quad (5)$$

$$\forall i, j \in O, (i \prec j) \in J_k \quad s_i + p_i \leq s_j \quad (6)$$

$$\forall i, j \in O, (i, j \in M_k) \quad s_i + p_i \leq s_j \vee s_j + p_j \leq s_i \quad (7)$$

El algoritmo evolutivo evalúa la aptitud de cada individuo que representa una solución al problema, con la función objetivo presentada en la ecuación (3). La función objetivo se relaciona con el máximo tiempo de término de ejecución de la última operación que se realiza en el taller de manufactura. Para este problema se requiere minimizar ese tiempo. El conjunto de restricciones en (5) indica que el tiempo en el que inicia cada operación j , debe ser mayor o igual a cero. El conjunto de restricciones en (6) define las restricciones de precedencia que existe entre pares de operaciones que pertenece a un mismo trabajo. El conjunto de restricciones en (7) define las restricciones de capacidad de recursos que existe entre pares de operaciones que ejecuta la misma máquina.

4. Instalación, configuración y documentación de un cluster de alto rendimiento para pruebas de laboratorio. Se capacitó a los estudiantes que participan en el proyecto, en instalación, configuración y administración de un cluster. Se instaló y configuró en el CIICAp un cluster de alto rendimiento, realizándose con esto las siguientes actividades. Instalación y sintonización de RedHat Enterprise Linux 4, instalación de NIS y NFS, instalación de Open MPI, instalación de Ganglia para el monitoreo de los nodos del Cluster e instalación de Torque para la administración de colas por lotes. Este cluster junto con el cluster del ITVer, servirán de infraestructura para pruebas preliminares realizadas en los algoritmos desarrollados con MPI.

Ajuste de actividades, justificación

Se realizaron algunos ajustes a las actividades, principalmente por la necesidad no prevista de tener un laboratorio de pruebas para evaluar la ejecución de los algoritmos paralelizados con librerías MPI. Esto hará que antes de entrar a la grid se cuente con los algoritmos evolutivos con la mejor puesta a punto, lo cual evite en la ejecución de pruebas la mayoría de las caídas que se puedan dar de algunos nodos por perdidas en el control de los procesos del genético o bien perdidas de los procesos mismos del genético que ocasionen un mal funcionamiento de estas pruebas. Actualmente se tiene terminado este laboratorio en el CIICAp, con la capacitación de los estudiantes en el manejo de este recurso. En

cooperación con el Cluster del ITVer, se generarán las primeras pruebas de ejecución de los algoritmos utilizando ambos clusters. De acuerdo a esta actividad adicional, el calendario y estatus queda de la siguiente manera.

Calendario de actividades, reprogramación y estatus

[illegible]

Referencias

[Nebro et al., 2008] Nebro A. J., Luque G., Luna F., Alba E., DNA Fragment Assembly Using a Grid-Based Genetic Algorithm, Computers & Operations Research ISSN 0305-0548, vol. 35, n°9, pp. 2776-2790, Elsevier Science. 2008.

[Fujisawa et al., 2004] Fujisawa K., Kojima M., Takeda A., Yamashita M., Solving Large Scale Optimization Problems via Grid and Cluster Computing, Journal of the Operations Research Society of Japan, Vol. 47, No 4, 265-274, 2004.

[Zunino, et al., 2007] Zunino I., Melab N., Talbi E-G., A Gris-enabled Framework for Exact Optimization Algorithms, Proceedings 21st European Conference on Modelling and Simulation ECMS 2007, ISBN 978-0-09553018-2-7, 2007.

[Philippe et al., 2006] Philippe L., Damy S., Hermann B., Djama I., Evaluation of a Large Scale Lookup Algorithm in ASP Based Grid, Proceedings of the Fifth International Symposium on Parallel and Distributed Computing (ISPDC 06), ISBN 0-7695-2638-1/06, IEEE Computer Society, 2006.

[Peng, et al., 2004] Peng L., See S., Jiang Y., Song J., Stoelwinder A., Neo H., Performance Evaluation in Computational Grid Environments, Proceedings of the Seventh International Conference on High (HPCAsia 04), ISBN 0-7695-2138-X/04, IEEE Computer Society, 2004.

[Priya et al., 2007] Priya S. B., Prakash M., Dhawan K. K., Fault Tolerance-Genetic Algorithm for Grid Task Scheduling using Check Point, The Sixth International Conference on Grid and Cooperative Computing (GCC 2007), ISBN 0-7695-2871-6/07, IEEE Computer Society, 2007.

[Huang et al., 2007] Huang J., Jin H., Xie X., Zhang Q., An Approach to Grid Scheduling Optimization Based on Fuzzy Association Rule Mining, Proceedings of the First International Conference on e-Science and Grid Computing (e-Science 05), ISBN 0-7695-2448-6/05, IEEE Computer Society, 2005.

[Chin et al., 2006] Chin S., Lee J., Yoon T., Yu H., List Scheduling Method for Service Oriented Grid Applications, Proceedings of the Second International Conference on Semantics Knowledge and Grid (SKG 06), ISBN 0-7695-2673-X/06, IEEE Computer Society, 2006.