Propuesta de Tesis:
Calendarización de flujos de trabajo en cómputo en la nube

Fernando Aguilar Reyes

Maestría en Ciencias en Computación
Instituto Tecnológico Autónomo de México
Río Hondo #1, Progreso Tizapán, Del. Álvaro Obregón, 01080
México, D.F.

21 de noviembre de 2013

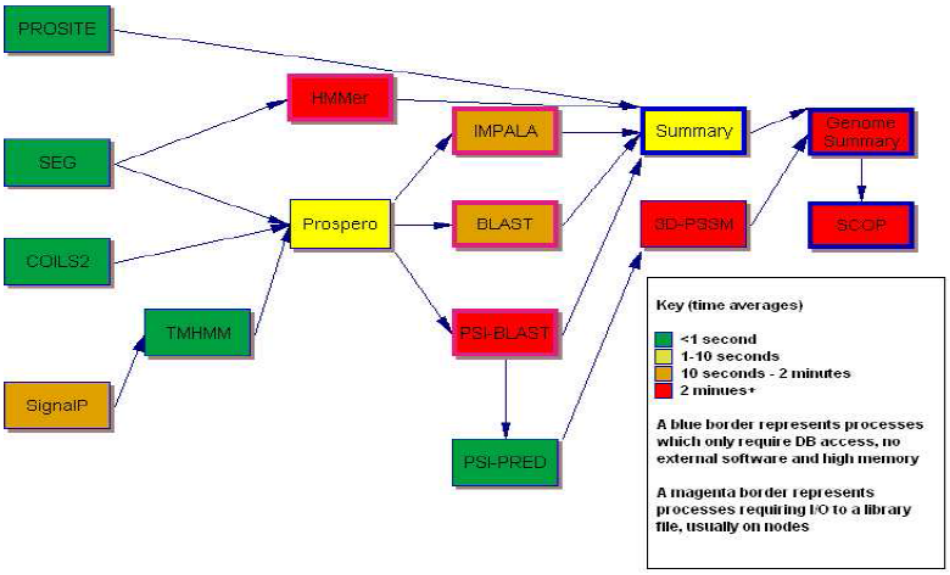

Figura 1: Flujo de trabajo para la anotación de proteínas, diseñado en la Escuela Imperial de Londres para el proyecto *e-Protein*

Resumen

Los flujos de trabajo, o conjunto de tareas de un proceso, son utilizados en aplicaciones como el etiquetado de proteínas o el ajuste de cuentas bancarias al final del día. Como requieren vastos recursos computacionales, se han empleado enfoques de cómputo distribuido (clusters y grids) para ejecutar estos flujos. Sin embargo, el enfoque de cómputo en la nube ha tomado un gran interés tanto por la academia como por la industria, porque el esquema de pagos "pay as you go" y la elasticidad de los recursos de la nube posibilitan la ejecución de estos flujos sin necesidad de contar con infraestructura especial. Una parte importante para la ejecución de los flujos es la calendarización de sus tareas. Con ello, podríamos hacer planeaciones que maximicen el uso del sistema distribuido o que minimicen el tiempo de ejecución total. En esta propuesta de tesis, proponemos crear un calendarizador de flujos de trabajo que tome en cuenta las características únicas del enfoque de cómputo en la nube para que se pueda minimizar el tiempo de ejecución y, también, el dinero utilizado por la renta de los servicios en la nube.

0.1. Análisis de antecedentes: Revisión inicial de literatura

0.1.1. Flujos de trabajo

Entendemos que un flujo de trabajo, o *workflow* en inglés, es un conjunto de pasos que modelan la ejecución de un proceso [4]. Un ejemplo de un flujo de trabajo es el que el se desarrolló en la Escuela Imperial de Londres para el proyecto *e-Protein* [6], cuyo objetivo era la idenficación y anotación de partes de proteínas que expliquen la estructura y la función de dicha proteína. Una proteína está compuesta de varios aminoácidos y se estima que existen al menos 500 aminoácidos diferentes [7], dando lugar a un número muy grande de posibles proteínas. De este modo, los investigadores de la Escuela Imperial diseñaron un flujo de trabajo compuesto de vario s programas especializados para hacer la anotación de proteínas. En la figura 1 podemos ver el flujo de trabajo [6], donde las flechas indican el flujo de datos y los colores indican el tiempo de ejecución aproximado.

Por lo general, los flujos de trabajo tienen una complejidad computacional que hace prohibitiva su ejecucion en una sola computadora. Por ello, diversos enfoques se han aplicado para distribuir la ejecución de un flujo de trabajo entre varias computadoras. De acuerdo a Buyya et al., los enfoques de cómputo más importantes para los flujos de trabajo son los *clusters*, los *grids* y las *nubes* [3]. A continuación, explicaremos cada uno de los enfoques.

- Los *clusters* son sistemas distribuidos, paralelos, compuestos de varias computadoras que son vistas como un único recurso de cómputo [3]. Un ejemplo de un cluster es la instalación de la Universidad Autónoma Metropolitana, campus Iztapalapa, llamada *Aitzaloa*, compuesta por 270 nodos de cómputo, cada uno equipado con dos procesadores Intel Xeon Quad-Core y 16GB en RAM; los nodos están conectados entre sí por medio de switches Ethernet e Infiniband. El cluster también cuenta con un sistema de almacenamiento distribuido badado en Lustre. La conexión con Internet se administra con el nodo maestro llamado Aitzaloa. La capacidad real de cómputo del cluster Aitzaloa es de 18.4 teraFLOPS [2].

- Los *grids* son sistemas distribuidos, paralelos, compuestos de computadoras autónomas y geográficamente distribuidas que pueden trabajar en conjunto o de manera independiente de acuerdo a los objetivos, políticas y mecanismos de uno o varios administradores del sistema, es decir, un grid puede ser compartido entre varias instituciones [3]. El proyecto *LANCAD*[1] es un buen ejemplo, pues une el cluster *Aitzaloa* de la UAM, el cluster de la UNAM *KamBalam*, y el cluster *Xiuhcoatl* del CINVESTAV por medio de una red de fibra óptica instalada en las estaciones del Sistema de Transporte Colectivo Metro. La suma de la potencias reales de cada *nodo robusto* del grid es de 48.55 teraFLOPS [1].

- Las *nubes* (clouds) son sistemas distribuidos, paralelos, compuestos de computadoras o máquinas virtuales interconectadas que son aprovisionadas para usarse como uno o varios recursos de cómputo, de acuerdo a un contrato de nivel de servicio acordado entre el proveedor de la nube y el cliente [3]. Empresas nuevas y existences proveen servicios de cómputo en la nube, tales como GoGrid, Rackspace, Amazon, Microsoft, IBM, Oracle, entre otras. La forma en que operan es muy sencilla: se paga cierta cantidad por utilizar servicios de cómputo o alacenamiento durante determinado tiempo. Así, los clientes no tienen que invertir grandes cantidades de dinero para contar con una gran infraestructura como en el caso de los los clusters y los grids.

0.1.2. Cómputo en la nube

El *cloud computing* o enofque de cómputo en la nube está tomando mucho interés tanto por la industria como por la comunidad científica, porque hace accesible una gran cantidad de recursos computacionales con cantidades razonables de presupuesto.

Cuando trabajamos con flujos de trabajo en los dos primeros enfoques, se utiliza software para administrar la ejecución del flujo de trabajo. Estos programas planean y organizan la ejecución de un flujo de

[1] Laboratorio Nacional de Cómputo de Alto Rendimiento

trabajo, es decir, *calendarizan*. Por ejemplo, Open Grid Scheduler tiene algoritmos para distribuir tareas paralelas (Parallel Virtual Machines o MPI) en un grid. También tiene políticas y mecanismos para administrar trabajos secuenciales, por medio de un batch-queue system. Pero, ¿cómo administramos flujos de trabajo que se ejecutan en la nube? Intuitivamente, podríamos pensar que el proveedor en la nube tiene software especializado para administrar el uso de las máquinas virtuales y los sistemas de almacenamiento. Lamentablemente, estos programas sólo administran la ejecución de las máquinas virtuales que pida el cliente, mas no administran flujos de trabajo en ejecución.

Podríamos, entonces, crear nuesto grid con máquinas virtuales y utilizar los administradores de flujo de trabajo que se utilizan en grids, como Open Grid Scheduler. Aunque esta solución suena viable, el enfoque de la nube agrega una característica que no está presente en los grids: podemos solicitar nuevas instancias de cómputo sin ninguna restricción física, el único límite es nuestro presupesto. Si tuviéramos un cluster propio para la ejecución de los flujos de trabajo, nuestro límite está marcado por las características del hardware, no podemos aumentarlo de manera "instantánea". De manera análoga, aunque podamos unir varios clusters distribuidos geográficamente, estamos limitados a la capacidad total del grid, sin tomar en cuenta si éste es compartido por varias instituciones. Además, el hecho de que existan varios proveedores de servicios de cómputo en la nube nos da la oportunidad de evaluar diferentes presupuestos que cada empresa ofrece para la ejecución de nuestro flujo de trabajo, dejando abierta la posibilidad de elegir la configuración más conveniente.

De esta forma, podríamos pensar en un calendarizador de flujos de trabajo que utilice el enfoque de la nube, tomando en cuenta las características únicas que hacen diferente este enfoque de los clusters y los grids. Con ello, podríamos elegir si queremos que el flujo sea ejecutado lo más rápido posible o que se optimice una cantidad fija de recursos; o mejor aún, ejecutar el flujo en el menor tiempo utilizando la menor cantidad de dinero.

0.1.3. Problemas similares

Para indagar si es posible diseñar y construir el calendarizador con cómputo en la nube, podríamos investigar problemas similares, como la calendarización de procesos en un sistema operativo o la calendarización de flujos de trabajo en clusters y grids.

Los sistemas operativos multitareas utilizan un administrador de tareas para repartir los recursos de cómputo (tiempo de CPU y memoria). Esta organización se ejecuta varias veces entre varios procesos, dando la ilusión de que todos los procesos se ejecutan al mismo tiempo. Con las limitaciones en disipación en calor, los fabricantes de microprocesadores optaron por diseñar CPU's con varios núcleos de procesamiento en un sólo chip. Por ello, los diseñadores de sistemas operativos también tuvieron que cambiar los algoritmos de calendarización de tareas. ¿Cómo es que estos calendarizadores están relacionados con los calendarizadores de flujos de trabajo con cómputo distribuido? La respuesta esta en que a partir de las ideas con las que se diseñan estos calendarizadores, podemos proponer un calendarizador para flujos de trabajo.

En el trabajo de Xiu et al. [5] hablan de un algoritmo de calendarización de ejecución de múltiples flujos de trabajo que son simples de calcular, pero que son numerosos. Éstos son ejecutados sobre servicios de cómputo en la nube. Los autores presentan una solución en donde se da prioridad a la optimización del costo necesario para ejecutar el flujo de trabjo, sacrificando un poco el tiempo de ejecucion total. A grandes rasgos, el algoritmo de calendarización relaja los tiempos límite de ejecución de las tareas de los flujos, con el motivo de no forzar competencia sobre los recursos más baratos como lo hace Deadline-MDP. El algoritmo de calendarización desarrollado, CTC, utiliza en promedio 15 % menos tiempo de ejecución que el algoritmo Deadline-MDP y también reduce en promedio 20 % el costo de ejecución reportado por el algoritmo de comparación.

Ahora, el problema de la calendarización de flujos de trabajo (workflow scheduling) con restricciones es, en general, un problema NP-completo [8].

0.2. Preguntas de investigación

1. ¿Cuál es la mejor forma de modelar un flujo de trabajo de manera tal que pueda ser ejecutado en un entorno de cómputo en la nube y que pueda ser evaluado para optimizar el consumo de tiempo o presupuesto del cliente?

2. ¿Qué hace especial el problema de la calendarización de flujos de trabajo en cómputo en la nube de otros problemas de calendarización, como un administrador de procesos de un sistema operativo? ¿Es necesario inventar nuevos esquemas de calendarización?

3. ¿Existe la posibilidad de poder modelar todas las restricciones de la calendarización del flujo de trabajo con las variables costo y tiempo?

Bibliografía

[1] Cluster híbrido de supercómputo — xiuhcoatl — cinvestav. `http://clusterhibrido.cinvestav.mx/`. Consultado el 10 de noviembre de 2013.

[2] Laboratorio de supercómputo y visualización en paralelo. `http://supercomputo.izt.uam.mx/infraestructura/aitzaloa.php`. Consultado el 10 de noviembre de 2013.

[3] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.

[4] J Octavio Gutierrez-Garcia and Kwang Mong Sim. Agent-based cloud workflow execution. *Integrated Computer-Aided Engineering*, 19(1):39–56, 2012.

[5] Ke Liu, Hai Jin, Jinjun Chen, Xiao Liu, Dong Yuan, and Yun Yang. A compromised-time-cost scheduling algorithm in swindew-c for instance-intensive cost-constrained workflows on a cloud computing platform. *International Journal of High Performance Computing Applications*, 24(4):445–456, 2010.

[6] Angela O'Brien, Steven Newhouse, and John Darlington. Mapping of scientific workflow within the e-protein project to distributed resources. In *UK e-Science All Hands Meeting*, pages 404–409, 2004.

[7] Ingrid Wagner and Hans Musso. New naturally occurring amino acids. *Angewandte Chemie International Edition in English*, 22(11):816–828, 1983.

[8] Marek Wieczorek, Andreas Hoheisel, and Radu Prodan. Towards a general model of the multi-criteria workflow scheduling on the grid. *Future Generation Computer Systems*, 25(3):237–256, 2009.