

Calendarización de flujos de trabajo en cómputo en la nube

Fernando Aguilar Reyes

12 de noviembre de 2013

1. Análisis de antecedentes: Revisión inicial de literatura

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 sería la anotación de proteínas, el cual consiste en identificar ciertas partes de la proteína y poner una etiqueta que describa la función de dicho componente [5].

Por lo general, los flujos de trabajo tienen una complejidad computacional que hace prohibitivos ejecutarlos 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 *Aitzaloo*, 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 basado en Lustre. La conexión con Internet se administra con el nodo maestro llamado Aitzaloo. La capacidad de cómputo del cluster Aitzaloo 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

¹Laboratorio Nacional de Cómputo de Alto Rendimiento

es un buen ejemplo, pues une el cluster *Aitzaloe* 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 cluster 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 existentes 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 almacenamiento 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 clusters y los grids.

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 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 usuario, mas no administran flujos de trabajo en ejecución.

Podríamos, entonces, crear nuestro 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 presupuesto. 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 dos enfoques de cómputo anteriores. 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 utilizando la menor cantidad de dinero.

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.

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

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?

Referencias

- [1] Cluster híbrido de supercómputo — xihcoatl — 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] 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.