

Introduciendo conceptos de Cloud Computing utilizando el entorno CMRE

Laura De Giusti¹, Franco Chichizola¹, Sebastián Rodríguez Eguren¹, Mariano Sánchez¹, Juan Manuel Paniego¹, Armando De Giusti^{1,2}

¹Instituto de Investigación en Informática LIDI (III-LIDI) – Facultad de Informática –UNLP

² Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

Argentina

{ldgiusti, francoch, seguren, msanchez, jmpaniego, degiusti}@lidi.info.unlp.edu.ar

Abstract. Se analiza el modo de incorporar conceptos básicos de computación en la nube (“cloud computing”) en etapas tempranas del aprendizaje de los alumnos, empleando el entorno multi-robot CMRE que se ha utilizado para introducir conceptos de concurrencia y paralelismo en un curso CS1.

El entorno CMRE (Concurrent Multi Robot Environment) se extiende para poder considerar la nube como un robot externo, con capacidades diferenciales y posibilidad de comunicarse en forma independiente con cada robot del entorno virtual y o real, de modo de explicitar al alumno los conceptos básicos de los servicios en la nube, sin cambiar el entorno simple en el que trabaja.

En el trabajo se analizan algunos casos ejemplo para mostrar el modo de presentar los conceptos para el alumno y relacionar las aplicaciones con actividades colaborativas de los robots para las cuales se requiere interacción con la nube.

Keywords: Concurrencia, Paralelismo, Algoritmos Paralelos, Cloud Computing.

1 Introducción

Las currículas informáticas clásicas [ACM68][ACM78][ACM99] contenían conceptos de concurrencia en diferentes áreas (Lenguajes, Paradigmas, Sistemas Operativos) pero omitían casi totalmente el tratamiento del paralelismo, salvo al plantear los conceptos de sistemas distribuidos.

Las actuales arquitecturas multiprocesador han producido un notorio impacto, obligando a replantear el “modelo base” para la enseñanza conceptual de las arquitecturas y de la programación. Reemplazar la “máquina de Von Neuman” con un solo hilo de control, por un esquema que integra múltiples procesadores cada uno con uno o más hilos de control y varios niveles de memoria accesible en forma diferenciada [GEP06][MCC08] conlleva también el necesario aprendizaje de la *programación paralela* sobre las arquitecturas actuales [ACM13]. Sin embargo, la programación paralela (y los conceptos fundamentales de concurrencia) resulta más compleja para un alumno en las etapas iniciales de su formación, y es necesario contar con nuevas estrategias que permitan abordarla exitosamente [DEG15].

El alumno actual de un curso CS1 está “entrenado” en elementos informáticos de alta interacción tales como consolas y juegos electrónicos, celulares, tablets y

computadoras. Esta familiarización favorece la tendencia al empleo de herramientas interactivas para la enseñanza de conceptos fundamentales de programación. Tendencia que se fortalece con elementos tales como la programación de robots [MIN15] a alumnos desde la escuela media y en los cursos universitarios iniciales [ACM04][ACM08][ACM13][DEG13].

En esta línea, la posibilidad de dar los primeros pasos en el mundo de la programación mediante un entorno gráfico e interactivo, coordinado con demostraciones en el mundo real con robots permite reducir la brecha que tradicionalmente existió entre la abstracción y la posibilidad de “ver” y “verificar” la ejecución de algoritmos. [AMD09][HOO13].

El entorno gráfico CMRE, donde se cuenta con un conjunto de robots que se mueven en una ciudad, ha permitido incorporar la enseñanza de los conceptos básicos de concurrencia y paralelismo en un curso inicial de Informática. En sucesivos trabajos ha sido presentado a la comunidad universitaria y está siendo empleado desde hace 3 años en la UNLP.

Asimismo los avances en las tecnologías de virtualización y cómputo distribuido han dado origen al paradigma de Cloud Computing, que se presenta como una alternativa a los tradicionales sistemas de Clusters y Grids para ambientes de cómputo paralelo [ROD07][DEG07] [BER08][KON09].

Las arquitecturas tipo “*Cloud*” se presentan como una evolución natural del concepto de *Clusters* y *Grids*, integrando grandes conjuntos de recursos virtuales (hardware, plataformas de desarrollo y/o servicios), fácilmente accesibles y utilizables por usuarios distribuidos, vía WEB [MIL08][VAQ09][FOS10]. Estos recursos pueden ser dinámicamente reconfigurados para adaptarse a una carga variable, permitiendo optimizar su uso. Al enfocarnos en Cloud Computing aparecen problemas clásicos de la Ciencia Informática, extendidos para este nuevo modelo de arquitectura: planificación, virtualización, asignación dinámica de recursos, migración de datos y procesos [ARD09][VAZ09].

El desafío de iniciar a los alumnos en la conceptualización del modelo de “*Cloud Computing*” a partir del primer año de sus estudios ha sido la motivación de este trabajo, en el que se trata de integrar al entorno CMRE algunas funcionalidades para comprender conceptualmente lo que el alumno puede esperar de la computación en la nube.

El trabajo se ha estructurado lo siguiente: la Sección 2 presenta el entorno CMRE actual. En la Sección 3 se abordan los aspectos centrales de Cloud Computing, mientras la Sección 4 se centra en las extensiones del entorno CMRE para considerar el rol del Cloud y la programación de aplicaciones que se vinculan con la nube. En la Sección 5 se detallan resultados alcanzados, conclusiones y líneas de trabajo futuro.

2 El entorno CMRE actual

Las características principales del entorno CMRE pueden resumirse de la siguiente manera [GEP06][DEG14]:

- Existen múltiples procesadores (robots) que realizan tareas y que pueden cooperar y/o competir. Los mismos representan los “cores” de una arquitectura

multiprocesador real. Estos robots virtuales pueden tener un reloj propio y diferentes tiempos para la ejecución de sus tareas específicas.

- El modelo de ambiente (“ciudad”) en la que desarrollan sus tareas admite áreas privadas (sólo puede ser recorrida por un robot de los declarados en el programa), parcialmente compartidas (un conjunto de robots pueden recorrerlas) y totalmente compartidas (todos los robots declarados pueden recorrerla).
- Si se instancia a un sólo robot en un área que abarque toda la ciudad, se repite el esquema del Visual Da Vinci [DEG13].
- Cuando dos o más robots están en un área compartida (parcial o totalmente), compiten por el acceso a las esquinas del recorrido y a los recursos que allí existan. Para esto deben sincronizar.
- Cuando dos o más robots (en un área común o no) desean intercambiar información (datos o control) deben hacerlo por mensajes explícitos.
- La sincronización se da por un mecanismo equivalente a un semáforo binario.
- La exclusión mutua puede generarse con la declaración de las áreas alcanzadas por cada robot. Acceder a otras áreas de la ciudad no está permitido.
- Todo el modelo de ejecución es sincrónico y permite la existencia de un reloj virtual de ciclos, que a su vez permite asignar tiempos específicos a las operaciones en cada robot, simulando la existencia de una arquitectura heterogénea.
- El entorno permite ejecutar el programa de manera tradicional, o paso a paso por instrucciones, dando al usuario un control detallado sobre la ejecución del programa, de manera de poder controlar situaciones típicas de concurrencia tales como conflictos (colisiones) o deadlocks, por ejemplo, se puede saber si cada robot está ejecutando o ha finalizado.
- En la ejecución paso a paso, el efecto de las operaciones se puede reflejar en los robots físicos, comunicados vía Wi-fi. Los robots físicos poseen un sistema operativo Linux que permite ejecutar un servidor http implementado en NodeJS [NJS15]. De esta manera el entorno se comunica con los robots (cada robot físico se corresponderá con uno virtual en el ambiente). La comunicación entre ellos es punto a punto, y bidireccional, es decir, el entorno envía las instrucciones al robot físico y luego este último envía su respuesta al entorno indicando la finalización de la instrucción indicada.
- Una dificultad que presenta actualmente el entorno es la de no tener una “visión global” del estado de las esquinas de la ciudad y de los robots (posición, cantidad elementos en su bolsa, etc), y por lo tanto no se puede resolver problemas de índole colaborativo o de optimización de trabajo general. Por esto se plantean algunas ideas en este trabajo y también se está experimentando con un “servidor” que contenga esta información (o la adquiera por una cámara general para los robots físicos) y se pueda comunicar con todos los robots para optimizar algoritmos.

3 Cloud Computing

3.1 Conceptos generales de Cloud

Antes de adentrarnos en la utilización de Cloud Computing con el encorno CMRE se deben mencionar algunos conceptos básicos del paradigma de Cloud Computing o Computación en la nube.

La computación en la nube es un modelo o paradigma de cómputo distribuido para la prestación y consumo de servicios de TI [VAZ09]. En los últimos años, ha ganado aceptación considerable como manera eficaz de adquirir y utilizar servicios informáticos con un costo accesible y predecible. Esto permite que el usuario final de estos recursos y servicios pueda acceder a los mismos sin necesidad de invertir grandes sumas de dinero en la adquisición de hardware para el armado y mantenimiento de un centro de cómputos el cual involucraría espacio físico, alimentación, una red para interconectar los equipos, refrigeración, almacenamiento y soporte técnico. Además el consumo sobre el cloud puede ajustarse a la demanda, no exigiendo compra de infraestructura local en función del crecimiento de nuevos servicios que se requieran [VAQ09]. En pocas palabras, Cloud Computing nos provee de recursos TI bajo demanda, a un costo proporcional al consumo de infraestructura y software requeridos.

Cuando hablamos de Cloud Computing debemos tener en cuenta que podemos elegir entre tres modelos de servicio: SaaS, PaaS e IaaS. A continuación, se mencionan las características principales de cada uno de estos modelos:

- SaaS (Software as a Service): el usuario cloud puede hacer uso de las aplicaciones que contrate al correspondiente proveedor. Un ejemplo puede ser una PyME que contrate una aplicación de correo electrónico para sus 30 empleados. La aplicación no podrá ser modificada por la PyME ni sus usuarios a excepción de posibles configuraciones de usuario o personalizaciones que le permita el proveedor. La aplicación se encontrará alojada en las infraestructuras cloud del proveedor y el usuario no tendrá ningún control sobre las mismas. Ejemplos de este tipo de servicios son: Dropbox, Gmail, Google Doc, Office 365, entre otros.
- PaaS (Platform as a Service): Cuando los recursos proporcionados por el Cloud son entornos configurados con todas las herramientas de software necesarias para el desarrollo y/o despliegue de aplicaciones personalizadas, sin que el usuario se ocupe de la instalación y administración de la infraestructura subyacente, entonces se considera un modelo PaaS. Estas plataformas ofrecen todo lo necesario para soportar los ciclos de vida completos de una aplicación y/o servicio WEB disponible en Internet. Ejemplos de este tipo de servicios son: Windows Azure, Salesforce, Dot Cloud, entre otros.
- IaaS (Infrastructures as a Service): Este modelo ofrece, por medio de la tecnología de virtualización, recursos de infraestructura virtual (servidores, equipos, dispositivos de almacenamiento, dispositivos de red y software de base), flexibles y escalables; en donde múltiples usuarios coexisten compartiendo el mismo hardware físico de forma transparente, segura e independiente. Sobre dicha infraestructura virtual alojará sus aplicaciones y plataformas; sobre estas últimas el

usuario tiene el control, pero no sobre la arquitectura física real asignada. Por Ej. Amazon Web Services, Windows Azure, Google Compute Engine, Rackspace, entre otros.

Otra característica a tener en cuenta cuando hablamos de Cloud Computing es el modelo de despliegue, del cual podemos distinguir 3 tipos:

- Cloud público: Es aquel desplegado por un proveedor de IT, que ofrece servicios (IaaS, PaaS y/o SaaS) de acceso público desde Internet. Generalmente, este modelo de despliegue se relaciona con un contexto comercial, donde los usuarios son considerados clientes y pagan por el tiempo de uso de los servicios. La principal ventaja, es la posibilidad de acceder inmediatamente a un conjunto de recursos con una mínima inversión. Las desventajas que presenta son el alto costo acumulado en el tiempo de uso del mismo y las garantías sobre la privacidad y seguridad de la información del usuario.
- Cloud privado: Se considera un Cloud privado, al que se encuentra desplegado en la intranet (en la red de datos detrás del firewall) de una organización, institución o empresa. El despliegue se efectúa sobre la infraestructura de hardware de la organización y los servicios proporcionados son de características similares al Cloud público, con la diferencia que el acceso está limitado a la red privada y no presenta carácter comercial. Las principales ventajas de un Cloud privado es la capacidad de brindar mayor eficiencia en el uso de la infraestructura física, como también un alto nivel de seguridad sobre los datos sensibles de la organización. Con el objetivo de lograr mayor eficiencia en la utilización de los recursos del datacenter, éstos se consolidan a través de la virtualización. Por otro lado, dado que el Cloud privado se encuentra desplegado en la Intranet de la organización, se garantiza la seguridad de los datos por medio de las mismas políticas y medidas de seguridad del datacenter.
- Cloud híbrido: Se denomina Cloud híbrido al despliegue de un Cloud privado que utiliza de forma segura los recursos de un Cloud público. Este modelo es de gran utilidad cuando una organización requiere aumentar rápidamente sus recursos privados para satisfacer los picos de demanda de determinados servicios. Generalmente, la organización contrata servicios de un Cloud público para expandir los recursos de su Cloud privado. A esta modalidad, también se la conoce como “Cloudbursting”. Un Cloud híbrido permite a las organizaciones obtener los beneficios combinados de un Cloud privado y uno público, logrando así el aumento de la eficiencia en el uso de la infraestructura física del datacenter, un mayor nivel de seguridad de los datos y una mayor disponibilidad de recursos para atender grandes demandas de servicio.

Un usuario puede adoptar uno o más de estos modelos según sus necesidades. La decisión vendrá condicionada por dónde desea centrar sus esfuerzos y conocimientos: en las aplicaciones, en las plataformas y/o en las infraestructuras tecnológicas. Qué elementos le aportan valor a su negocio y por lo tanto quiere seguir implicado más de cerca en su evolución y cuales no le suponen un valor diferencial y prefiere contratar a un proveedor especializado.

Si hacemos hincapié en los beneficios que nos brinda el paradigma de Cloud Computing a la hora de utilizar sus servicios, nos encontramos con la posibilidad de tener recursos disponibles bajo demanda, interfaces de administración basadas en servicios web, escalabilidad y elasticidad de recursos, aprovisionamiento automático de recursos, y autoservicio. Sin embargo, también existen riesgos, que es necesario conocer y tener en cuenta a la hora de utilizar estos servicios: seguridad física de los datos, de las conexiones a los servicios, en la autenticación y el control de acceso; dependencia absoluta de la red y del ancho de banda que se tenga, necesidad de estar siempre conectado; dependencia del proveedor, posibles fallas del cloud o las comunicaciones, aspectos legales que pueden variar de acuerdo a la ubicación geográfica de los datacenters del proveedor; integridad de los datos, posibles pérdidas, privacidad y confidencialidad y potencial acceso a datos privados sin aviso por aplicación de leyes o mandatos judiciales.

3.2 Objetivos conceptuales para el aprendizaje en un alumno de un curso CS1

La idea principal es que el alumno adquiera de manera sencilla los conceptos relacionados a Cloud Computing. En una primera instancia el énfasis estará en ver el Cloud como SaaS.

Desde este punto de vista, en el Cloud se tiene un componente externo al sistema, que en CMRE será un “super robot” que conoce el estado global de la ciudad, y brinda un conjunto de servicios para optimizar algoritmos orientados a resolver de manera colaborativa problemas que involucren más de 1 robot en la ciudad.

Teniendo en cuenta que el “super robot” se encuentra en un hardware distinto al resto de los robots, no poseen un área compartida, por lo que la comunicación es exclusivamente por medio de mensajes (lo cual responde al modelo del Cloud, con el cual nos comunicamos vía red local o internet).

4 Extensiones conceptuales del entorno CMRE para Cloud Computing

4.1 La nube como un “super robot” externo al entorno.

Conceptualmente para el alumno (que en nuestro caso cursa Taller de Programación en el segundo semestre de primer año, con 3 paradigmas: imperativo, orientado a objetos y concurrente) es relativamente simple admitir que el entorno CMRE tiene posibilidad de vincularse con un “robot externo” con otras capacidades de resolución de problemas y de comunicación de primitivas a ejecutar por múltiples robots en el entorno.

Análogamente este modelo se aplica a los robots físicos empleados en demostración, los cuales efectivamente se pueden conectar vía Internet a la nube (“super robot” del modelo) y recibir comandos y control.

Es de hacer notar que el “super robot” ve todo el escenario de la ciudad, la ubicación de objetos en las esquinas, eventuales obstáculos en los recorridos y también conoce todos los datos de los robots (por ej. velocidad de movimiento de cada uno).

Claramente sobre este “super robot” pueden correr algoritmos más complejos que los admitidos por los comandos de los robots del entorno CMRE actual.

4.2 Posibilidades de optimización de algoritmos multirrobot

Al tener una visión global del entorno (“ciudad”) se pueden plantear problemas de optimización simple, en los que vía comunicación con el “super robot” externo, cada uno de los robots del entorno realicen una determinada tarea en función de una función objetivo a maximizar.

Es interesante comprender que esto responde a uno de los paradigmas más empleados en programación concurrente/paralela que es el de “master-slave”, con la particularidad que el “master” tiene un repertorio de instrucciones extendido y la posibilidad de resolver algoritmos más complejos que cada uno de los robots “esclavos”. Además respetando la posibilidad de heterogeneidad en los robots “esclavos” para enriquecer la resolución de problemas.

4.3 Alcance de la programación por un alumno de un curso CS1

El alcance planteado para el curso 2017 es tener problemas donde la ciudad es compartida por múltiples robots y se plantea un objetivo que requiere colaboración, en el cual el modo más natural de “optimizarlo” es conocer dinámicamente la situación de toda la ciudad y todos los robots.

Esto se podría resolver con la arquitectura de CMRE actual, pero requiere una fase de intercambio de mensajes entre los robots para decidir la próxima acción de cada uno (lo cual hace más lenta la respuesta al crecer el número de robots y/o la dimensión de la ciudad). Por otro lado las decisiones seguirán siendo “sub óptimas” ya que el conocimiento disponible se limitará a la suma del alcance del conocimiento de cada robot del entorno.

El alumno de primer año *no* programará la aplicación de optimización global sobre el “super robot”, simplemente podrá pedir un servicio de software desde cada uno de los robots del entorno. Naturalmente sí se le explica la potencialidad de ese “super robot” en la nube para brindar servicios de software más complejos que los propios del CMRE básico.

4.4 Alcance de la programación por alumnos avanzados

Con el mismo planteo los alumnos avanzados (3er. año Programación Concurrente, 4to. año Sistemas Paralelos y otras asignaturas afines) pueden trabajar con la programación de los algoritmos de optimización en el “super robot” y además comprender conceptos como el control dinámico del programa que ejecuta cada robot del entorno, en función de optimizar una función objetivo.

4.5 Extensiones sobre el entorno de robots físicos

Naturalmente la vinculación de los robots físicos con la nube permite introducir los mismos conceptos, con una demostración real de la implementación de algoritmos colaborativos que involucran el conjunto de robots físicos (incluso siendo heterogéneos y respondiendo a diferentes comandos). Es interesante analizar que nuevos componentes físicos del entorno (como una o más cámaras que analizan la

escena) pueden agregarse y sus datos ser capturados y procesados en la nube, comunicando luego los comandos orientados a la optimización de la tarea global a cada uno de los robots físicos.

Esta etapa crece en potencialidad para los alumnos avanzados, que se introducen a los temas de Cloud Robotics [KEH15] que marcan una de las líneas actuales de gran interés en las aplicaciones informáticas.

4.6 Ejemplo de ejercitación experimental

A modo de ejemplo se ha trabajado con un caso muy simple en el que N robots tienen que alcanzar N objetivos en la ciudad (ej. 8 robots y 8 objetos a recoger) de modo que el número de pasos totales a realizar por el conjunto de robots sea mínimo.

Notar que este problema permite extensiones (N diferente para robots y objetivos, diferente velocidad de movimiento de los N robots, cambio de la función a optimizar, etc.).

Resulta claro que en la nube (nuestro “super robot”) con conocimiento de las N ubicaciones de los robots y los objetos, planificar un movimiento óptimo, en número de pasos por ejemplo, es relativamente sencillo y permite definir una “trayectoria” a realizar por cada robot que optimice el resultado final.

Además se pueden definir otros objetivos de interés como minimizar las instancias de sincronización entre robots (por llegar al mismo tiempo a la misma esquina por ejemplo) y permite fácilmente comparar la eficiencia del resultado con algoritmos de “fuerza bruta” tales como asignar aleatoriamente un objetivo a cada robot.

5 Resultados actuales, Conclusiones y Líneas de Trabajo Futuro

Para lograr el objetivo planteado en el trabajo inicialmente se han estudiado los conceptos de Cloud Computing y la forma de incorporarlos gradualmente utilizando el entorno CMRE.

A partir de los conocimientos adquiridos, se ha adaptado el entorno CRME para que pueda ser utilizado desde un Cloud. Esta adaptación incluye el tener que comunicar el entorno (que corre en el Cloud) con los robots físicos. Para esto se ha estudiado el tema y se ha implementado un algoritmo de comunicación con sockets en lenguaje Python.

Asimismo se ha mantenido CMRE sobre una máquina local y se ha habilitado la comunicación virtual de cada robot con el cloud, pasando por la máquina local como un intermediario transparente al alumno.

Además en 2016 se avanzó en el trabajo presentado el año anterior [DEG15] para enseñar la importancia del concepto de consumo energético y exceso de temperatura en el área de concurrencia y paralelismo. En este sentido, actualmente se está desarrollando una tesina de grado y se ha modificado en forma experimental el entorno CMRE, incluyendo las instrucciones para medir y graficar el consumo energético y la temperatura de cada robot en una aplicación. A partir de los valores obtenidos se pueden tomar decisiones como reducir la velocidad de un robot, o hasta detener momentáneamente uno de ellos.

- [BER08] Bertogna, M., Grosclaude, E., Naiouf, M., De Giusti, A., Luque, E.: "Dynamic on Demand Virtual Clusters in Grids". In: 3rd Workshop on Virtualization in High-Performance Cluster and Grid Computing (VHPC 08). España. (2008).
- [DEG07] De Giusti, A., Naiouf, M., Pettoruti, J. E., Pousa, A., Rodríguez, I. P., De Giusti, L., Chichizola, F., Ardenghi, J., Bertogna, L., Printista, M.: Parallel Algorithms on Multi-Cluster Architectures using GRID Middleware. Experiences in Argentine Universities". In: Garcia Tobio, J., Doallo Biempica, R., López Cabido, I., Martín Santamaría, M. J., Gómez Tato, A., Touriño Dominguez, J. (eds.), 1st Iberian Grid Infrastructure Conference (IBERGRID) Proceedings, pp. 322—332. Spain. (2007)
- [DEG13] De Giusti, L., Leibovich, F., Sanchez, M., Chichizola, F., Naiouf, M., De Giusti, A. "Desafíos y herramientas para la enseñanza temprana de Concurrencia y Paralelismo". Congreso Argentino de Ciencias de la Computación (CACIC), 2013.
- [DEG14] De Giusti, A., De Giusti L., Leibovich, F., Sanchez, M., Rodriguez Eguren, S. "Entorno interactivo multirrobot para el aprendizaje de conceptos de Concurrencia y Paralelismo". Congreso Tecnología en Educación, Educación en Tecnología. 2014.
- [DEG15] Laura De Giusti, Fabiana Leibovich, Franco Chichizola, Marcelo Naiouf, Armando De Giusti. "Incorporando conceptos en la enseñanza de Concurrencia y Paralelismo utilizando el entorno CMRE" Congreso Argentino de Ciencias de la Computación 2015 (CACIC 2015).
- [FOS10] Foster I. "There's Grid in them thar Clouds". 2 de Enero, 2008.<http://ianfoster.typepad.com/blog/2008/01/theres-grid-in.html>. Noviembre, 2010.
- [GEP06] Gepner P., Kowalik M.F. "Multi-Core Processors: New Way to Achieve High System Performance". In: Proceeding of International Symposium on Parallel Computing in Electrical Engineering 2006 (PAR ELEC 2006). Pags. 9-13. 2006.
- [HOO13] Hoonlor A., Szymanski B. K., Zaki M. J., Thompson J. "An Evolution of Computer Science Research". Communications of the ACM. 2013.
- [KEH15] Kehoe B., Patil S., Abbeel P., Goldberg K. "A Survey of Research on Cloud Robotics and Automation". IEEE Transactions on Automation Science and Engineering (T-ASE): Special Issue on Cloud Robotics and Automation. Vol. 12, no. 2. Apr. 2015.
- [KON09] Kondo, D., Javadi, B., Malecot, P., Cappello, F., Anderson, D. P.: "Cost-benefit analysis of Cloud Computing versus desktop grids". In: IPDPS '09 Proceedings. IEEE International Symposium on Parallel and Distributed Processing. Washington, USA (2009).
- [MCC08] McCool M. "Scalable Programming Models for Massively Parallel Multicores". Proceedings of the IEEE, 96(5): 816–831, 2008.
- [MIL08] Miller M. "Cloud computing: web-based applications that change the way you work and collaborate online". Que Publishing. USA 2008.
- [MIN15] Proyecto PROGRAM.AR del Ministerio de Ciencia y Tecnología e Innovación Productiva. Web: <http://www.mincyt.gob.ar/divulgacion/programar-9920>.
- [NJS15] <https://nodejs.org/api/http.html>
- [ROD07] Rodríguez, I. P., Pousa, A., Pettoruti, J.E., Chichizola, F., Naiouf, M., De Giusti, L., De Giusti, A.: "Estudio del overhead en la migración de algoritmos paralelos de cluster y multicluster a GRID". In: Proceedings del XIII Congreso Argentino de Ciencias de la Computación. Argentina (2007).
- [VAQ09] Vaquero, L. M., Rodero-Merino, L., Caceres, J., Lindner, M.: "A Break in the Clouds: Towards a Cloud Definition". In: ACM SIGCOMM Computer Communication Review, Volume 39, Issue 1, pp. 50--55. USA. (2009).
- [VAZ09] Vázquez Blanco, B., Huedo, E., Montero, R. S., Llorente, I. M.: "Elastic Management of Cluster-based Services in the Cloud". In: Proceedings pp.19-24, ACM Digital Library. ISBN 978-1-60558-564-2. 2009.

Asimismo el trabajo de los robots físicos contra el Cloud forman parte de un trabajo de Beca Doctoral y de 2 cursos de posgrado enfocados en Cloud Robotics.

Como Conclusiones podemos sintetizar:

- El entorno CMRE permite trabajar con algunos aspectos de Cloud Computing, en particular el enfoque de SAAS, sin modificaciones sustanciales.
- La potencialidad de vincular robots trabajando en tiempo real con la nube excede la simple enseñanza inicial de conceptos de programación paralela, pero es muy importante para los alumnos aprender algunos conceptos básicos desde un curso CS1.
- La programación de aplicaciones colaborativas sobre el entorno CMRE requiere mecanismos que permitan conocer el contexto global de la ciudad y características de cada robot. Este “conocimiento” y “volumen de datos” es natural que residan en la nube, lo cual se refuerza si pensamos en tratamiento de video o imágenes en tiempo real para la toma de decisiones por los robots.
- Es necesario diferenciar el alcance de la programación que puede realizar un alumno de un curso CS1 y aquel alumno de cursos avanzados.

Las principales líneas de trabajo futuro incluyen:

- Transmitir las nociones de IAAS permitiendo que el “super robot” pueda instanciar robots o datos en el entorno a demanda.
- Incorporar efectivamente al menos una cámara global para procesar el entorno para los robots físicos.
- Analizar casos con robots heterogéneos donde la función a optimizar sea el consumo de energía total.

6 Bibliografía

- [ACM04] ACM/IEEE-CS Joint Task Force on Computing Curricula. “Computer Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering”. Report in the Computing Curricula Series. 2004.
- [ACM08] ACM/IEEE-CS Joint Interim Review Task Force. “Computer Science Curriculum 2008: An Interim Revision of CS 2001”. Report from the Interim Review Task Force. 2008.
- [ACM13] ACM/IEEE-CS Joint Task Force on Computing Curricula. “Computer Science Curricula 2013”. Report from the Task Force. 2013.
- [ACM68] ACM Curriculum Committee on Computer Science. “Curriculum ‘68: Recommendations for the undergraduate program in computer science”. Communications of the ACM, 11(3):151-197. 1968.
- [ACM78] ACM Curriculum Committee on Computer Science. “Curriculum ‘78: Recommendations for the undergraduate program in computer science”. Communications of the ACM, 22(3):147-166. 1979.
- [ACM99] ACM Two-Year College Education Committee. “Guidelines for associate-degree and certificate programs to support computing in a networked environment”. New York: The Association for Computing Machinery. 1999.
- [AMD09] AMD. “Evolución de la tecnología de múltiple núcleo”. <http://multicore.amd.com/es-ES/AMD-Multi-Core/resources/Technology-Evolution>. 2009.
- [ARD09] Ardissono L., Goy A., Petrone G., Segnan M. "From Service Clouds to User-centric Personal Clouds". 2009 IEEE Second International Conference on Cloud Computing.