SUPERCOMPUTACIÓN
CASTILLA Y LEÓN

FED4AMIGA federation for GTC, ALMA and SKA data processing

Objetivo del proyecto

El estado del arte de diversos ámbitos de la investigación y desarrollo científico, está marcado por la presencia de instrumentos y metodologías que generan y requieren el procesado de enormes volúmenes de datos. El escenario hermético en el que un centro o un laboratorio es autosuficiente para las necesidades de investigación está comenzando a presentar cierta obsolescencia, manifiesta ya en los grandes experimentos e infraestructuras como LOFAR (***Low Frequency ARray***) o el LHC (***Large Hadron Collider***), que se expanden incluso en ámbitos multinacionales. Existen, por lo tanto, necesidades computacionales para algunos experimentos que requieren de recursos externos para llevarlos a cabo.

En este nuevo escenario, en el que la computación y la explotación de grandes volúmenes de datos alcanzan un gran protagonismo dentro del método científico, se presentan necesidades de nuevas herramientas y paradigmas. Atendiendo a estas necesidades aparece uno de los instrumentos emergentes para el desarrollo, reusabilidad y divulgación científica: los llamados flujos de trabajo o *workflows*. Los flujos de trabajo se presentan como poderosas herramientas para el manejo y análisis de datos. Este proyecto, siguiendo el principio SaaS (*Software como Servicio*) tiene como objetivo dar soporte al diseño y ejecución de *workflows* científicos en infraestructuras de gran capacidad de computación.

Bajo estas premisas, se constituye FED4AMIGA con el acometido de servir de nexo entre la comunidad científica y los distintos recursos computacionales existentes. Así, la meta fundamental de FED4AMIGA se sintetiza como el desarrollo de una "capa integradora" para el desarrollo de *workflows* científicos sobre infraestructuras computacionales de gran capacidad. Concretamente, FED4AMIGA en su estado actual, hace uso de un clúster HPC ubicado en las instalaciones de la propia Supercomputación de Castilla y León (SCAYLE).

Metodología

En el emergente mercado de los flujos de trabajo, nos encontramos con un escenario heterogéneo, donde conviven numerosas tecnologías y paradigmas. En este escenario aparecen tanto motores de workflows -herramientas de diseño y gestión- (Taverna, Galaxy, Triana, Kepler,...) como portales y gateways (WS-PGRADE, gUSE,...) o proyectos de integración de flujos de trabajo (ER-FLOW, SCI-BUS,...).

En este proyecto, por las facilidades ofrecidas en la experiencia de usuario, por su versatilidad y su interfaz intuitiva, se ha optado inicialmente por usar Taverna como herramienta para el diseño y ejecución de los workflows.

El modo más estándar, multiplataforma, versátil y adaptable para proveer servicios computacionales integrables dentro de herramientas de diseño y gestión de flujos de trabajo, son los Servicios Web. Su naturaleza autodescriptiva (a través del Lenguaje de Definición de Servicios Web, WSDL) hace de estos una herramienta ideal para su acceso por parte de usuarios y aplicaciones cliente.

Un reto importante para este proyecto era trazar un camino entre los usuarios y los recursos computacionales. Los usuarios interactuarían con su motor de flujos de trabajo (v.g. Taverna) importando servicios web. Esta interfaz supone un extremo de la cuerda. En el otro extremo se encuentran los elementos de computación.

En el mapa de ruta del proyecto, el primer hito importante ha sido el diseño y construcción de una

arquitectura que permitiera el acceso de los usuarios a los Servicios Web, y la conexión de estos con el sistema de colas del clúster de SCAYLE. Esta arquitectura se describe en secciones posteriores.

Para dar sentido a la arquitectura, se ha fijado, como caso de uso inicial, el objetivo de diseñar los servicios web para llevar a cabo un análisis de cubos de datos de astronomía radiointerferométrica para producir modelos cinemáticos de galaxias aisladas. Este es un caso de uso propuesto dentro del proyecto AMIGA4GAS, en el que se orquestan conjuntamente distintas tareas provistas por un software de análisis y modelado astrofísico desarrollado en el Kapteyn Astronomical Institute en Groningen, Holanda: GIPSY (Groningen Image Processing System).

Estas tareas, relacionadas entre sí con sus consecuentes dependencias de información, constituyen un workflows científico ideal para validar la efectividad de la plataforma desarrollada en el proyecto. Sin embargo, las tareas de análisis computacional a llevar a cabo en este flujo, son de naturaleza secuencial aunque presentan, en base a la repetición de las mismas, la posibilidad de una ejecución paralela. La invocación múltiple de servicios web individuales supone una solución de granularidad gruesa y sumamente ineficiente. Por eso, se ha hecho uso de COMPSs[2], un entorno de programación desarrollado por el Barcelona Supercomputing Center (BSC), para potenciar el paralelismo inherente en el flujo de análisis astronómico propuesto.

Tras un éxito inicial, se plantea el crecimiento a nuevos ámbitos científicos, en concreto al ámbito de la meteorología, usando WRF (Weather Research and Forecasting), un sistema de nueva generación de predicción meteorológica numérica. Aprovechando las sinergias existentes fruto de la colaboración continuada de SCAYLE con el Grupo de Física de la Atmósfera de la Universidad de León (GFA), se plantea la adaptación de FED4AMIGA para mejorar los mecanismos de interacción de los investigadores del GFA con el clúster de supercomputación de SCAYLE.

Arquitectura

Para recorrer el trayecto entre las interfaces de usuario y los recursos de computación, se ha ideado e implantado una arquitectura de capas (Figura 1).

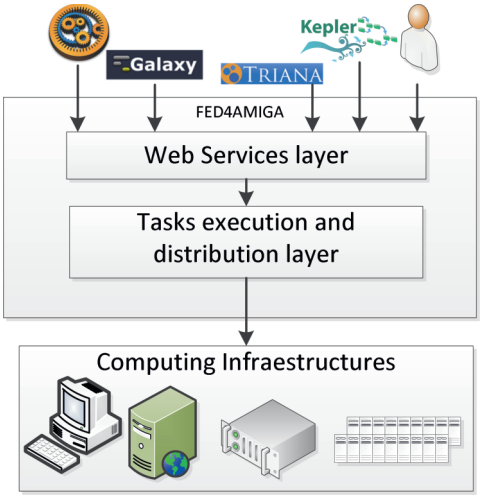

Figura 1. Vista global de la arquitectura de FED4AMIGA. © SCAYLE

En la parte superior del sistema se encuentran las herramientas de edición y gestión de flujos de trabajo o, potencialmente, cualquier sistema capaz de consumir servicios web. FED4AMIGA se constituye propiamente en dos capas: una primera con los servicios web ofrecidos y por debajo, la capa encargada de la ejecución y distribución de tareas sobre las infraestructuras de computación.

Capa de Servicios Web

En la capa de Servicios Web, aparecen numerosas tecnologías (Figura 2). Los servicios web se han implementado en python, por lo que se requieren tecnologías compatibles.

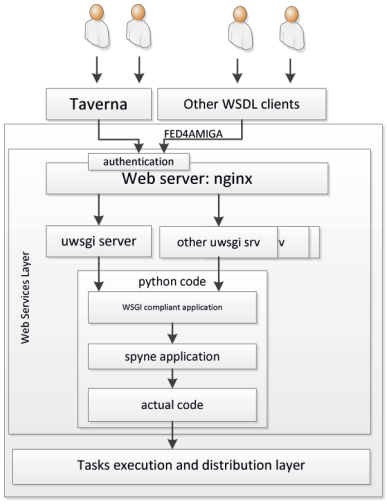

Figura 2. Capa de Servicios Web. © SCAYLE

Los servicios son descubiertos por medio de WSDL y los mensajes con las aplicaciones clientes (ej: Taverna), se transmiten por medio del protocolo SOAP. Para escuchar y dar respuesta a estos mensajes, se requiere un servidor web. Aunque en un inicio, se usó Lighttpd como servidor web, posteriormente se optó por Nginx por sus características: Nginx es rápido, compatible con SSL y LDAP (por medio de un módulo adicional), permite el uso de uwsgi y ofrece la posibilidad de configurar de forma sencilla múltiples servidores para ofrecer balanceo de carga.

Como nexo de unión entre las aplicaciones en python y el servidor web, se usa WSGI (Web Server Gateway Interface), que define una interfaz simple e universal entre servidores web y aplicaciones web o frameworks para el lenguaje de programación python.

El control de acceso se lleva a cabo a través del servidor web Nginx. En SCAYLE, un servidor LDAP se usa para autorizar y autenticar usuarios. Los credenciales de los usuarios se reciben usando Basic HTTP authentication, por lo que la transmisión de datos se ha protegido con encriptación SSL. La aplicación de escritorio Taverna Workbench ofrece la posibilidad de importar servicios web definidos mediante WSDL y acceder a ellos usando Basic HTTP authentication. También es capaz de gestionar certificados para trabajar con encriptación SSL.

Por medio de un socket, Nginx se comunica con un servidor uWSGI usando el protocolo uwsgi (uWSGI es un protocolo y un servidor para la construcción de aplicaciones de web distribuidas -de hecho, el servidor podría hacer las veces de servidor web-).

El servidor uWSGI permite servir aplicaciones que cumplan con WSGI. Algunos tests[1]] afirman que uWSGI ofrece un gran rendimiento y escalabilidad con un gran desempeño incluso para números elevados de conexiones simultáneas.

Para ofrecer una aplicación compatible con la interfaz WSGI, los servicios web han sido implementados usando Spyne y Ladon. Ambos son toolkits para ofrecer servicios RPC (Remote Procedure Call) accesibles por medio de diferentes protocolos (entre ellos SOAP). Con estas dos herramientas se consigue ofrecer como servicios web algunos métodos, haciendo uso de decoradores Python. Esta sintaxis permite conservar un código limpio; y la variedad de protocolos da la posibilidad de reusar el código de los servicios web para la reutilización en otros escenarios.

La dualidad de toolkits obedece a cuestiones cronológicas: siendo Spyne la herramienta seleccionada en el inicio de las implementaciones. Spyne es un desarrollo más robusto y versátil, pero la integración con Taverna resulta más verbosa, dado que -por su forma de construir los servicios y su descripción WSDL- exige la inclusión de elementos adicionales en los flujos de trabajo para la asignación de entradas y salidas de los servicios web. Ladon, por su parte, permite una interacción más limpia, pero presentaba algunos problemas que han exigido correcciones en sus fuentes. A medida que se han ido corrigiendo los defectos de Ladon, se ha ido afianzando como el toolkit usado para ofrecer una aplicación compatible con WSGI.

Al final, la implementación de los servicios web hace uso de elementos la capa inferior de Ejecución y Distribución de Tareas para delegar la ejecución en los elementos de cálculo.

Paralelismo y distribución de tareas. COMPSs

Para entender la capa de Ejecución y Distribución de Tareas, conviene describir primero una de las herramientas empleadas en la misma: COMPSs.

COMP superscalar (COMPSs) [2], precedido por GRID superscalar (GRIDSs) [3], es un modelo de programación que pretende facilitar el desarrollo de aplicaciones para infraestructuras distribuidas, como Clústers, Grids y Clouds. Es un desarrollo del BSC-CNS 4] que permanece en continua evolución. Constituye una capa que se ubica sobre JavaGAT, e incluye un entorno en tiempo de ejecución que explota el paralelismo inherente en aplicaciones secuenciales.

Este modelo de programación provee los medios para la paralelización automática de aplicaciones ahorrando al programador la codificación de los mecanismos de paralelización y distribución (creación de hilos, sincronización, paso de mensajes, tolerancia a fallos,...). Permite el uso de Java y C/C++ como lenguaje de programación de las aplicaciones a paralelizar (soportar Python está dentro del roadmap previsto en el desarrollo). No obstante, cabe la posibilidad de desarrollar pequeñas

aplicaciones Java, que sirvan como envoltorio de aplicaciones secuenciales codificadas en otro lenguaje.

COMPSs ofrece una sintaxis para definir las tareas que han de ser distribuidas y ejecutadas en los distintos recursos computacionales. Esta metodología permite mantener el código fuente original intacto. Cuando una aplicación se lanza usando el entorno de COMPSs, éste detecta la invocación de las tareas seleccionadas y construye un grafo de dependencias entre las tareas invocadas, teniendo en cuenta los parámetros que de entrada y salida de las mismas. Si un parámetro se refiere a un archivo, COMPSs transmitirá los archivos al emplazamiento donde la ejecución tendrá lugar (al recurso computacional encargado de llevarla a cabo).

Según se menciona anteriormente, COMPSs usa JavaGAT para el envío de tareas a los recursos así como para la gestión de archivos. De este modo, las aplicaciones construidas por medio de COMPSs no presentan cohesión con ninguna plataforma en concreto (no especifican cómo se hace la transmisión de datos ni reserva de recursos). Esto abre la posibilidad a portar aplicaciones entre infraestructuras de forma transparente.

Las posibilidades que ofrece para paralelizar software secuencial hacen de COMPSs una herramienta ideal para los objetivos de este proyecto, en concreto para el caso de uso de análisis astrofísico; en el que se ejecutan aplicaciones secuenciales barriendo un rango de parámetros, que conlleva la repetición de tareas. En este caso de uso, COMPSs permite recoger toda esa funcionalidad en una aplicación Java donde cada set de parámetros determina la ejecución de una tarea que será distribuida entre los recursos computacionales escogidos.

Capa de Ejecución y Distribución de Tareas

La capa de Ejecución y Distribución de Tareas es la responsable de distribuir los trabajos entre los elementos de computación (Figura 3).

En el estado actual del proyecto, la distribución de Tareas se centra únicamente en distribución sobre un clúster de supercomputación (en concreto el sistema de colas SGE que gobierna el clúster de

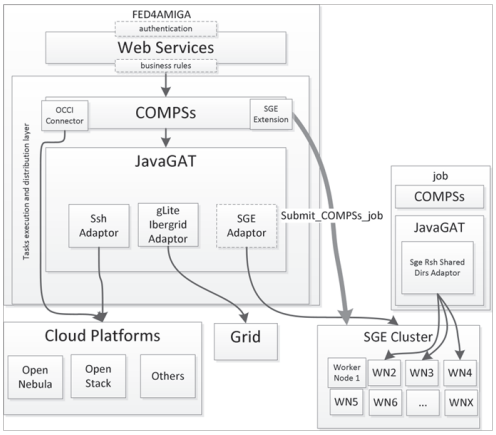

Figura 3. Capa de Ejecución y Distribución de Tareas. © SCAYLE

SCAYLE). No obstante, por completitud, en esta figura se muestra un escenario en el que se vislumbran diferentes infraestructuras y entornos de computación.

La Capa de Servicios Web delega la ejecución en el entorno de ejecución de COMPSs. Normalmente, COMPSs crearía grafos de tareas y distribuiría las tareas a los entornos de computación por medio de diferentes adaptadores de JavaGAT. Aparte de la versatilidad que ofrecen esos adaptadores de JavaGAT (usados para el envío de trabajos y transmisión de archivos), COMPSs incluye un conector OCCI que permite el instanciado de máquinas virtuales que servirían para recibir y ejecutar las tareas.

Para el clúster SGE, se observan dos posibles esquemas de uso:

- Directamente a través de un adaptador JavaGAT. Este escenario se ha desestimado por cuestiones de rendimiento: cada tarea supondría la instanciación de un trabajo en el clúster, lo cual es sumamente ineficiente por el tiempo de reserva y liberación de recursos y una causa potencial de saturación del sistema de colas.
- Mediante el envío al clúster de un trabajo que inicie una instancia completa del runtime de COMPSs.

En este caso, el envío del trabajo conllevaría la reserva de un número fijo de nodos de trabajo (working nodes) dentro del clúster SGE. El propio trabajo se encargaría de descubrir de modo

automático los nodos reservados para distribuirles las tareas de COMPSs. En este esquema, se usa un adaptador personalizado de JavaGAT para transmitir las tareas y transferir los archivos entre los nodos de trabajo del clúster SGE.

Casos de uso. Explotación del sistema

En un primer caso de uso que ha servido como hilo conductor en el diseño e implementación de FED4AMIGA, se ha planteado llevar a cabo un flujo de trabajo para el análisis de cubos de datos de astronomía radiointerferométrica para producir modelos cinemáticos de galaxias aisladas. Este es un caso de uso propuesto dentro del proyecto AMIGA4GAS, en el que se orquestan conjuntamente distintas tareas provistas por el software de análisis y modelado astrofísico GIPSY (Figura 4)

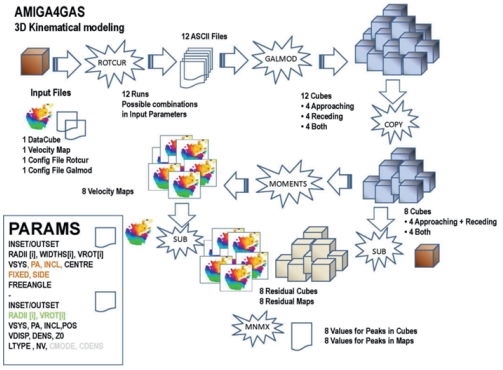

Figura 4. Diagrama de flujo del modelado de cubos de datos para astronomía radiointerferométrica. Imagen cedida por el Instituto de Astrofísica de Andalucía (IIA)

En esta figura se observan distintas tareas (ROTCUR, GALMOD, ...). Algunas de estas tareas se han ofrecido como servicios web, que permiten su ejecución en el clúster de SCAYLE. El usuario de FED4AMIGA podrá importar estos servicios web para confeccionar el flujo de trabajo deseado. Se requerirán las credenciales de usuario que serán transmitidas de forma segura, de modo que el acceso a los recursos de supercomputación permanece regido por las políticas de seguridad y acceso de SCAYLE.

Algunas de estas tareas se ejecutan barriendo sobre un rango de parámetros, por lo que existe un paralelismo presente en la repetición de invocación de tareas. Alternativamente, se han ofrecido algunos servicios web que comprenden la ejecución de varias tareas de forma conjunta (incluyendo el consecuente postprocesado de datos entre ellas), de modo que el paralelismo presente en la repetición de tareas pueda ser gestionado y explotado directamente dentro de los servicios web, permitiendo un paralelismo de granularidad más fina potenciado por el uso de COMPSs.

Tras el éxito del flujo realizado en primera instancia, se ha planteado abarcar otras ramas de la ciencia. En concreto en el ámbito meteorológico. El uso de un sistema de predicción como WRF plantea la ejecución de un flujo de trabajo.

FED4AMIGA plantea plasmar este diagrama dentro de su arquitectura, de modo que las tareas de WRF que requieren de capacidad de cómputo participantes en este flujo, sean ejecutadas en el clúster de computación de manera transparente para el investigador.

Participación de SCAYLE

Dentro de este proyecto, la función principal de SCAYLE será la de desarrollar un sistema federado compuesto por los nodos Grid de IAA-CSIC e IT. Los workflows científicos serán lanzados sobre el sistema federado que decidirá, en base al estado de la infraestructura (eficiencia energética, probabilidad de finalización con éxito, tiempo de latencia de los datos, etc.), donde es más eficiente ejecutar el workflowy dirigir allí al mismo, de una forma totalmente transparente para el usuario.

Además, gracias a Caléndula se puede ir un poco más allá de la ejecución de workflows en diferentes infraestructuras por separado, creándose así un sistema federado mediante técnicas de cloud computing que integrará en una sola las infraestructuras de Grid y Supercomputación. De esta forma esta solución no solo pone un amplio abanico de infraestructuras al servicio de los workflows científicos, si no que las hace más accesibles para los científicos.

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD
Código AYA2011-30491-C02-02

Conclusiones y Líneas Trabajo Futuras

En el desarrollo de este proyecto se ha construido una plataforma cuyo núcleo fundamental lo constituyen una serie de servicios web que facilitan la ejecución de determinados algoritmos de análisis científico en la infraestructura de HPC de SCAYLE de forma transparente para los usuarios.

El acceso a los servicios web se ha demostrado válido dentro de flujos de datos construidos a través de un motor de workflows como Taverna. Una línea de trabajo futuro viable es la posibilidad de ofrecer el acceso a estos servicios web desde otras herramientas: aplicaciones de Escritorio ad-hoc para los casos de uso propuestos, portales web, otros gestores de flujos, ...

Herramientas como las desarrolladas dentro de FED4AMIGA, permiten mantener una independencia de las partes involucradas en el proceso científico actual, eliminando las barreras tecnológicas que existen en el acceso a grandes herramientas de supercomputación. FED4AMIGA contribuye a favorecer una especialización de las personas involucradas en la experimentación científica con dependencia en el proceso de datos de altas necesidades computacionales; permitiendo así la focalización del trabajo de los especialistas en su ámbito de competencia. Con este proyecto se pretende favorecer el uso intensivo de los recursos computacionales durante su tiempo de vida productivo (por lo tanto, el aprovechamiento de los fondos destinados desde su adquisición y puesta en marcha hasta su obsolescencia); y también quiere resultar de utilidad al proceso científico y evitar prácticas que se producen en la experimentación numérica como la reducción de resolución y calidad de algunas simulaciones por el esfuerzo que supone el acceso a ciertos recursos computacionales.

FED4AMIGA se idea no sólo con el objetivo de dar soporte para el desarrollo de flujos científicos, sino también con la pretensión de constituirse como una capa capaz de federar el acceso a distintas infraestructuras de forma transparente. Esta capa integradora tiene como misión constituirse como una herramienta de acceso a recursos computacionales para poner éstos en valor y reducir las barreras tecnológicas que dificultan la explotación de los recursos por parte de la comunidad científica. FED4AMIGA no sólo ha de servir de enlace, sino que ha de ser capaz de proveer, de forma transparente, el acceso a las infraestructuras de computación con mayor idoneidad para llevar a cabo las tareas seleccionadas, cotejando una serie de reglas de negocio que pueden incluir aspectos como la latencia, la eficiencia energética, el menor tiempo de proceso, la probabilidad de éxito o incluso cuestiones legales y económicas. Tanto la federación como la implementación de estas reglas de negocio constituyen un reto futuro para FED4AMIGA.

Resultados

La arquitectura construida supone una alternativa para la propia SCAYLE a la hora de ofrecer sus infraestructuras para el aprovechamiento científico y por lo tanto el enriquecimiento cultural regional.

La infraestructura desarrollada ha visto refrendada su efectividad y viabilidad como herramienta para el desarrollo de flujos científicos al sustentar algunas de las tareas de un workflow científico presentado por el Instituto de Astrofísica de Andalucía en el marco del proyecto Wf4Ever.

El workflow usa los servicios web ofrecidos por FED4AMIGA como parte del proceso del modelado cinemático de cubos de datos de galaxias. En síntesis, el flujo completo tiene como objeto generar y parametrizar distintas curvas de rotación en aras de seleccionar la más apropiada para el posterior modelado cinemático.

Este flujo representa un caso de éxito de la infraestructura construida en FED4AMIGA y valida a la plataforma como una herramienta para el acceso a servicios de supercomputación de forma gráfica, sencilla e integrable dentro de una herramienta de diseño y gestión de flujos de trabajo como Taverna.

Participantes del proyecto

En este proyecto, se han promovido las alianzas entre instituciones tales como el Instituto de Astrofísica de Andalucía (IAA-CSIC) y Supercomputación de Castilla y León (SCAYLE), ambas en estrecha colaboración con el Barcelona Supercomputing Center – Centro Nacional de Supercomputación (BSC-CNS).