

Resumen

La tesis presentada se enmarca dentro del ámbito de la ciencia computacional. Dentro de esta, se centra en el desarrollo de herramientas para la ejecución de experimentación científica computacional, el impacto de la cual es cada vez mayor en todos los ámbitos de la ciencia y la ingeniería. Debido a la creciente complejidad de los cálculos realizados, cada vez es necesario un mayor conocimiento de las técnicas y herramientas disponibles para llevar a cabo este tipo de experimentos, ya que pueden requerir, en general, una gran infraestructura computacional para afrontar los altos costes de cómputo. Más aún, la reciente popularización del cómputo en la Nube ofrece una gran variedad de posibilidades para configurar nuestras propias infraestructuras con requisitos específicos. No obstante, el precio a pagar es la complejidad de configurar dichas infraestructuras en este tipo de entornos. Además, el aumento en la complejidad de configuración de los entornos en la nube no hace más que agravar un problema ya existente en el ámbito científico, y es el de la reproducibilidad de los resultados publicados. La falta de documentación, como las versiones de software que se han usado para llevar a cabo el cómputo, o los datos requeridos, provocan que una parte significativa de los resultados de experimentos computacionales publicados no sean reproducibles por otros investigadores. Como consecuencia, se produce un derroche de recursos destinados a la investigación.

Como respuesta a esta situación, existen, y continúan desarrollándose, diferentes herramientas para facilitar el procesos como el despliegue y configuración de infraestructura, el acceso a los datos, el diseño de flujos de cómputo, etc. con el objetivo de que los investigadores puedan centrarse en el problema a abordar. Precisamente, esta es la base de los trabajos desarrollados en la presente tesis, el desarrollo de herramientas para facilitar que el cómputo científico se beneficie de entornos de computación en la Nube de forma eficiente.

El primer trabajo presentado empieza con un estudio exhaustivo de las prestaciones d'un servicio relativamente nuevo, la ejecución *serverless* de funciones. En este, se determinará la conveniencia de usar este tipo de entornos en el cálculo científico midiendo tanto sus prestaciones de forma aislada, como velocidad de CPU y comunicaciones, como en conjunto mediante el desarrollo de una aplicación de procesamiento *MapReduce* para entornos *serverless*.

En el siguiente trabajo, se abordará una problemática diferente, y es la reproducibilidad de experimentos computacionales. Para conseguirlo, se presentará un entorno, basado en *Jupyter*, donde se encapsule tanto el proceso de despliegue y configuración de infraestructura computacional como el acceso a datos y la documentación de la experimentación. Toda esta información quedará registrada en el *notebook* de *Jupyter* donde se ejecuta el experimento, permitiendo así a otros investigadores reproducir los resultados simplemente compartiendo el *notebook* correspondiente.

Volviendo al estudio de las prestaciones del primer trabajo, teniendo en cuenta las medidas y bien estudiadas fluctuaciones de éstas en entornos compartidos, como el cómputo en la Nube, en el tercer trabajo se desarrollará un sistema de balanceo de carga diseñado expresamente para este tipo de entornos. Como se mostrará, este componente es capaz de gestionar y corregir de forma precisa fluctuaciones impredecibles en las prestaciones del cómputo en entornos compartidos.

Finalmente, y aprovechando el desarrollo anterior, se diseñará una plataforma completamente *serverless* encargada de repartir y balancear tareas ejecutadas en múltiples infraestructuras independientes. La motivación de este último trabajo viene dada por los altos costes computacionales de ciertos experimentos, los cuales fuerzan a los investigadores a usar múltiples infraestructuras que, en general, pertenecen a diferentes organizaciones. Se demostrará como la plataforma desarrollada es capaz de balancear la carga de forma precisa al mismo tiempo que divide las tareas automáticamente para cumplir con una cota en el tiempo de ejecución especificada por el usuario. Además, la plataforma asiste a las infraestructuras de cómputo en el proceso de escalado de recursos, minimizando el exceso de recursos y proporcionando así un uso eficiente de los recursos de cómputo.