UNIVERSIDAD DE EXTREMADURA

CENTRO UNIVERSITARIO DE MÉRIDA

INGENIERO EN INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

PROYECTO FIN DE GRADO

INTEGRACIÓN DE UNA HERRAMIENTA DE CÓMPUTO EVOLUTIVO CON UNA DE PROCESAMIENTO MASIVO DE INFORMACIÓN

Autor: Daniel Lanza García

Mérida - Junio 2015

UNIVERSIDAD DE EXTREMADURA

CENTRO UNIVERSITARIO DE MÉRIDA

INGENIERO EN INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

PROYECTO FIN DE GRADO

INTEGRACIÓN DE UNA HERRAMIENTA DE CÓMPUTO EVOLUTIVO CON UNA DE PROCESAMIENTO MASIVO DE INFORMACIÓN

Autor: Daniel Lanza García

Director: Francisco Fernández de Vega

Codirector: Francisco Chávez de la O

Mérida - Junio 2015

Prólogo

En las primeras lineas que describen este proyecto fin de carrera, ...

Agradecimientos

Quiero dar mi mas sincero agradecimiento a A todos muchas gracias.

Índice general

Índice de figuras

Índice de tablas

1. INTRODUCCIÓN

1.1 Motivaciones

La computación evolutiva está adquiriendo cada vez mas importancia a lo largo de los años, sus cada vez mas aplicaciones en sistemas de diferente naturaleza la hacen imprescindible para la resolución de problemas, que haciendo uso de otro modelo computacional, no podrían ser resueltos.

Con el paso de los años y la evolución de los sistemas computacionales, la computación evolutiva se ha intentado aplicar para la resolución de problemas cada vez mas complejos, lo cual en la mayoría de los casos, suele conllevar el uso de más recursos como capacidad de cómputo, memoria y tiempo. Esto hace que se busquen soluciones para mejorar el uso de estos recursos.

Numerosas investigaciones se han llevado a cabo con el fin de minimizar el uso de uno de los recursos mas importantes mencionados anteriormente, el tiempo, se han diseñado algoritmos en este modelo cuya ejecución puede tomar años y que por lo tanto su tiempo de ejecución es impracticable. Varias metodologías se han aplicado para reducir el tiempo de ejecución, una de ellas es la paralelización de parte del proceso, esto puede llevarse a cabo con los procesadores de última generación los cuales poseen varios núcleos de procesamiento o también se puede conseguir con el uso de varios computadores conectados en red.

Este trabajo plantea una solución para utilizar los recursos disponibles de forma eficiente y así reducir los tiempos de ejecución, la solución que se describirá hace uso de una herramienta que explota ambas metodologías, ejecución paralela en procesadores multi-núcleo y uso de numerosas computadoras.

1.2 Objetivos

1.3 Recursos empleados

1.4 Organización del documento

2. ANÁLISIS DEL SISTEMA

2.1 Introducción

2.2 Algoritmos evolutivos

f...

2.3 Procesamiento masivo de información

...

3. DESARROLLO DEL PROYECTO

3.1 Estudio de las herramientas a integrar

3.1.1 ECJ como herramienta de cómputo evolutivo

3.1.2 Hadoop como herramienta de procesamiento masivo de información

3.2 Implementación

4. MANUAL DE USUARIO

4.1 Obtención

El primer paso para la utilización de la solución implementada, no puede ser otro que obtenerla. Con el fin de que la comunidad pueda conseguirla sin problema, se ha creado un repositorio publico desde donde se puede descargar.

El repositorio está localizado en la conocida página por los desarrolladores GitHub (`https://github.com`), en ella se encuentran numerosos proyectos de código abierto y en la cual se ha almacenado este.

Todos los proyectos almacenados en esta página hacen uso de una herramienta de control de versionado llamada Git, la cual hemos también utilizado en este proyecto.

¿Qué es Git?

Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux. Algunas de sus características mas destacadas son:

- Fuerte apoyo al desarrollo no lineal, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones.
- Gestión distribuida. Git le da a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales.
- Los almacenes de información pueden publicarse por HTTP, FTP, rsync o mediante un protocolo nativo, ya sea a través de una conexión TCP/IP simple o a través de cifrado SSH.
- Los repositorios Subversion y svk se pueden usar directamente con git-svn.
- Gestión eficiente de proyectos grandes.

El proyecto se puede encontrar a través del buscador del sitio buscando .ecj_hadoop", o accediendo directamente al repositorio siguiendo esta dirección: `https://github.com/dlanza1/ecj_hadoop` . Una vez en el repositorio, el proyecto se puede obtener de dos formas, una es pulsando en el botón "Download ZIP"(situado en la columna de la derecha), y una vez descargado descomprimir el fichero, y la otra es clonar el repositorio con Git (crear un repositorio igual de forma local).

Para clonar el repositorio con Git se debe tener instalada esta herramienta en el sistema [2] o utilizar un entorno de desarrollo como Eclipse, el cual la trae incluida.

Para clonarlo desde un entorno de desarrollo como Eclipse [1] el procedimiento suele ser el de importar un proyecto pero con la diferencia de que se debe indicar que la fuente es un repositorio Git, nos solicitará el repositorio que queremos clonar y es ahí donde debemos escribir la URL del repositorio. Algo que debe ser aclarado es que esta operación creará un repositorio local, no genera el proyecto en el entorno de desarrollo, lo cual se explica en la sección siguiente.

Si de otro modo, tenemos la herramienta instalada en el sistema, debemos en primer lugar abrir una consola, posteriormente dirigirnos al directorio donde queremos crear el repositorio local y por último clonar el repositorio con el siguiente comando:

```
[usu@host repo]$ git clone https://github.com/dlanza1/ecj_hadoop
```

Una vez clonado tendremos una copia idéntica del repositorio la cual contiene la última versión del proyecto además de toda la información necesaria para que funcione el sistema de versionado utilizada por Git.

4.2 Importar a entorno de desarrollo

Esta sección tiene como objetivo explicar el procedimiento para importar el proyecto en un entorno de desarrollo, esto no es necesario para su ejecución por lo que no estén interesados en modificar/ampliar el proyecto pueden continuar la lectura en la sección siguiente (Compilación).

En el apartado anterior se ha explicado como obtener el proyecto, pero lo obtenido es básicamente los ficheros de código fuente, no se incluyen proyectos de entornos de desarrollo ni ejecutables.

Si acabamos de descargar/clonar el proyecto, no lo tendremos importado en nuestro entorno, para llevar a cabo esta operación tan solo debemos dirigirnos al menu de importación de proyectos, solucionar que es un proyecto Maven y seleccionar el directorio que contiene el proyecto.

Otra opción, teniendo la herramienta Maven instalada en el sistema [3], es generar primero el proyecto del entorno de desarrollo y posteriormente importarlo como un proyecto normal. Para generar el proyecto del entorno de desarrollo debemos abrir una consola, dirigirnos al directorio donde se sitúa el proyecto y ejecutar el comando Maven correspondiente a nuestro entorno, por ejemplo para Eclipse debemos ejecutar el siguiente comando:

```
[usu@host repo]$ mvn eclipse:eclipse
```

Esto nos generará el proyecto Eclipse y podremos importarlo como cualquier otro proyecto.

4.3 Compilación

Como se mencionaba anteriormente, los ficheros ejecutables no son proporcionados por lo que se deben generar. Con el uso de la herramienta de construcción de proyectos, Maven, explicaremos como construir el proyecto para que pueda ser ejecutado, no obstante, si tenemos el proyecto importado en un IDE, no es necesario utilizar Maven para compilarlo, podemos utilizar los usuales procedimientos para compilarlo.

¿Qué es Maven?

Maven es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Tiene un modelo de configuración de construcción simple, basado en un formato XML. Es un proyecto de nivel superior de la Apache Software Foundation.

Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

El motor incluido en su núcleo puede dinámicamente descargar plugins de un repositorio, el mismo repositorio que provee acceso a muchas versiones de diferentes proyectos. Este repositorio pugnan por ser el mecanismo de facto de distribución de aplicaciones en Java. Una caché local de artefactos actúa como la primera fuente para sincronizar la salida de los proyectos a un sistema local.

Maven se ejecuta en unas circunstancias similares a las que explicábamos con Git. Existen dos opciones, o tener Maven instalado en el sistema [3], o utilizar un entorno de desarrollo como Eclipse [1], el cual lo trae incluido.

Si tenemos importado el proyecto en un IDE, para compilarlo (o en nomenclatura de Maven, construirlo) es necesario hacer clic derecho en el fichero incluido en el proyecto, pom.xml, y seleccionar la opción: Run as (Ejecutar como) y en el submenú, Maven build, si hacemos esto por primera vez nos aparecerá un cuadro de dialogo donde se nos solicita los objetivos (goals), ahí debemos indicar "install"(sin las comillas) y aceptar/ejecutar.

En caso de que no estemos utilizando un IDE, necesitaremos tener instalado Maven. Para compilarlo de esto modo debemos abrir una consola, dirigirnos al directorio donde se encuentra el proyecto y ejecutar el siguiente comando:

```
[usu@host repo]$ mvn install
```

Al compilarlo con Maven, tanto desde el IDE como desde la consola, se generará un fichero .jar el cual contiene todas las clases compiladas.

5. TRABAJOS FUTUROS

6. CONCLUSIONES

7. ANEXO I. CÓDIGO FUENTE

BIBLIOGRAFÍA

[1] Entorno de desarrollo Eclipse [online]. URL: https://eclipse.org/.

[2] Procedimientos para la instalación de Git [online]. URL: http://symfony.es/documentacion/guia-de-instalacion-de-git/.

[3] Procedimientos para la instalación de Maven [online]. URL: https://eljaviador.wordpress.com/2013/05/21/guia-de-instalacion-de-maven/.

[4] Articulo de la Wikipedia sobre Git [online]. URL: http://es.wikipedia.org/wiki/Git.

[5] Articulo de la Wikipedia sobre Maven [online]. URL: http://es.wikipedia.org/wiki/Maven.