

Aportacións do proxecto ó software libre

Autor: *Alberto Luengo Cabanillas* (aluengocabanillas@gmail.com)

Lugo, a 30 de Outubro de 2010

Índice

1. Introducción: Obxectivos e motivación	3
2. Tecnoloxías <i>open-source</i> empregadas	3
2.1. Entorno de desenvolvemento e execución.....	3
2.2. Linguaxe de programación.....	4
2.3. Documentación e presentación.....	4
2.4. Implementación, execución e probas	4
3. Actividade pública do proxecto	6

1. Introducción: Obxectivos e motivación

O proxecto describe o desenvolvemento dun framework software extensible o máis xenérico posible que permite a execución de Algoritmos Xenéticos (en adiante, AXs) sobre un entorno distribuído, explotando tódala funcionalidade do modelo de programación distribuída MapReduce. Isto, baixo o noso punto de vista, pode presentar moitas vantaxes, como por exemplo o prototipado rápido de AXs, xa que o soporte á execución distribuída tolerante a erros xa está incluída en Hadoop (a implementación *open-source* de MapReduce escollida para esta proxecto). Por outro lado, proporciónase unha estrutura de clases suxeita a un alto nivel de persoalización e expansión (gran número de parámetros de configuración, inclusión/modificación intuitiva de clases, etc.). Por último, a encapsulación do proxecto como un ficheiro JAR facilita a portabilidade do mesmo entre distintos entornos (que, eso sí, deben dispoñer dunha instalación de Java).

2. Tecnoloxías *open-source* empregadas

Neste apartado desglosaranse pola súa natureza as distintas tecnoloxías libres empregadas no proxecto.

2.1. Entorno de desenvolvemento e execución

Dende o comezo do proxecto planteouse a necesidade de que a aplicación construída fose multiplataforma, debido ó carácter heteroxéneo dos entornos nos que se vai a executar. Sen embargo, o entorno local de desenvolvemento foi Windows XP por comodidade e porque tiña gran parte do entorno de execución previamente configurado. Aínda así, empregouse por motivos de accesibilidade unha máquina virtual mediante a ferramenta **VMWare** (da que só necesitabamos o reproductor gratuito para lanza-la máquina virtual) que montaba unha distribución **Ubuntu Server 8.04**. Mediante esta máquina virtual podíamos simula-las execucións da aplicación de forma local. Nos momentos nos que se realizaban as probas no entorno de produción (clúster 'nm' do Departamento de Electrónica e Sistemas da Facultade de Informática da Coruña - <http://nm.des.udc.es/>, sobre o que está montada unha distribución de Linux),

dispoñiamos dunha conta de usuario e unha shell dende a que podiamos compilar e executar tamén a aplicación.

2.2. Linguaxe de programación

Neste proxecto empregouse **Java** por familiaridade e porque as implementacións escollidas de MapReduce estaban codificadas con esta linguaxe. De esta forma, a documentación de cada unha das clases programadas está feita de acordo ó estándar JavaDoc, para que sexa posible construír a documentación do API de forma cómoda (mediante un “target” no ficheiro de proxecto *build.xml*).

2.3. Documentación e presentación

Para a elaboración da documentación da memoria do proxecto, empregáronse os editores de texto Microsoft Office Word e **OpenOffice.org Writer** (este último, parcialmente). Nesta documentación incluíronse distintos diagramas de clase, secuencia, etc. realizados íntegramente coa aplicación **MagicDraw UML 12.0**.

A presentación do proxecto realizouse principalmente con Microsoft Office PowerPoint e **OpenOffice.org Draw**.

2.4. Implementación, execución e probas

Existen moitas implementacións do modelo de programación MapReduce dispoñibles no mercado, algunhas de elas propietarias e outras libres. Dentro de este último conxunto, escollimos **Hadoop**, un proxecto Apache desenvolvido en Java e sometido a control de versións que cumpría perfectamente os requisitos buscados. Asimesmo, empregamos un subproxecto de Hadoop –tamén desenvolvido en Java– chamado **Pig**, que nos permite tratar de forma cómoda grandes conxuntos de datos.

Como ferramenta de desenvolvemento empregámo-lo IDE **Eclipse** (en concreto, a súa versión 3.5), xa que a outra alternativa que tiñamos –NetBeans– non dispoñía dos *plugins* de Hadoop e Pig necesarios para construí-la aplicación.

Á hora de integrar tanto Hadoop como Pig dentro de Eclipse, atopámonos con varias dificultades (bugs en determinados ficheiros, incompatibilidade de versións, etc), polo que tomáronse dúas decisións:

1. Descarga-las ramas “trunk” dispoñibles nese momento dos proxectos Hadoop e Pig (0.20.1 e 0.4, respectivamente). Desta forma, tiñámo-lo código fonte á nosa disposición e poderíamos traballar con el de maneira máis axeitada. Para xestionar-las distintas revisións de ambos proxectos empregámo-lo cliente SVN **TortoiseSVN**.
2. Como o obxectivo era conseguir dous plugins estables para a nosa versión de Eclipse, empregámo-la ferramenta de xestión de proxectos **Ant** para construí-los distintos ficheiros JAR desexados.

Por mor de desenvolver-la aplicación baixo Windows -cando Hadoop non tiña soporte nativo para esta plataforma- tivemos que lanzar-lo IDE que empregamos dende dentro de **Cygwin**, unha ferramenta libre que nos permite lanzar un entorno *sandbox* do estilo Unix para aproveitar-las características do devandito entorno nas nosas aplicacións. No noso caso concreto, o plugin de Hadoop para Eclipse fai uso do comando “whoami”, nativo en Unix pero non en Windows para conectarse ó sistema de ficheiros distribuído da máquina remota na que pretendemos lanzar-lo sistema.

Finalmente, para conectarnos ó cluster dende o entorno de traballo local, empregamos dúas ferramentas:

1. **Putty**, un cliente SSH para Windows co que nos conectabamos á nosa conta do usuario no clúster e arrancabamo-los demonios de Hadoop necesarios.
2. **WinSCP**, con esta ferramenta de transferencia de arquivos (o equivalente en Windows ó comando Unix “scp”), pasabamos rapidamente o ficheiro JAR da aplicación dende o noso espazo de traballo local á nosa conta no clúster

3. Actividade pública do proxecto

Dende as primeiras etapas do proxecto, decidiuse versiona-lo código mediante a ferramenta “**Google Code**”, xa que o autor xa traballara con esta ferramenta con resultados satisfactorios, e ademais porque proporciona unhas funcionalidades moi interesantes para proxectos de marcado carácter investigador. Así, o código fonte, as publicacións de referencia e os distintos instaladores dos programas empregados atópanse na URL: <http://code.google.com/p/mrpga/>. Pódese apreciar que tódolos recursos creados polo autor aí depositados están baixo a versión 3 da licenza **GNU General Public License**.

Para as distintas anotacións realizadas durante o periodo de desenvolvemento, empregouse a ferramenta **Google NotePad**, xa que permite compartir estas notas exportándoas como documento “Google Docs” ou como HTML (por exemplo, na URL <http://www.google.com/notebook/html?nbid=BDVwg3goQ89Xl8qkk&hl=es> pódense consulta-las distintas notas do diario de desenvolvemento).

Como notas de carácter personal, comentar que moitos dos avances, problemas e curiosidades que foron xurdindo durante a etapa de codificación, fixéronse públicas na conta de **Twitter** do autor (<http://twitter.com/albertoluengo>), así como na de **Facebook**. (usuario “Alberto Caba”). Por outra banda, na canle de **Youtube** creada especificamente para este proxecto, pódense consulta-los tres vídeos creados con motivo da demo que había que executar no momento da defensa do proxecto. As URLs dos vídeos son as seguintes:

- <http://www.youtube.com/watch?v=wVu0aMq1-Ck>. Neste vídeo amósase o fluxo completo de execución do sistema con un dos problemas de Algoritmos Xenéticos implementados, dende a fase de compilación ata a de execución na conta de usuario do autor no clúster obxectivo.
- <http://www.youtube.com/watch?v=xlCF5yIu5f8>. Neste segundo vídeo apréciase a execución do sistema nun entorno pseudo-distribuído (é dicir, lanzando a aplicación contra unha máquina virtual que facía as veces de entorno distribuído).

- <http://www.youtube.com/watch?v=0Hj0QO4hog8>. Por último, amósase a execución da versión secuencial no entorno *sandbox* de Java presente no IDE Eclipse.

Finalmente, participouse de maneira activa nas distintas **listas de correo** de carácter xeral de Hadoop (general@hadoop.apache.org) e Pig (user@pig.apache.org), tanto consultando as dúbidas que ían xurdindo como axudando ó resto dos usuarios, sempre como usuario “cabiwan”.