NutchOSGi

Nutch with osgi

"OSGi technology is the dynamic module system for Java™ OSGi technology provides a service-oriented, component-based environment for developers and offers standardized ways to manage the software lifecycle."

Why

- I like the idea of extensions (components, OSGi bundles) being packaged in single .jar file
- By thinking in components/services/interfaces the design in generall will naturally be more clear and less tight couplings are used

Short term goals

- Evaluate how OSGi (more specifically Apache Felix) could fit in nutch, perhaps the easiest place to start are is the plugin system and plugins
- Build minimal prototype system that can complete full crawling cyckle
- Verify that required actions could be done non intrusively, eg. nutch-osgi must not introduce any changes to nutch or hadoop (but can of course bring up things discovered and "wishlist" to these projects to make life easier for nutch-osgi)
- Add support to use OSGi bundles as nutch plugins (if applicable)

What has been done so far

Build system was mavenized. Required jars were grouped together and packaged as OSGi-bundles, packaking was made with maven2 and with help of maven-osgi-plugin. (maven2 is not required but IMO it makes things easier)

Bundles and their contents

bundle	**jars**
nutch-hadoop	hadoop-0.9.0-SNAPSHOT,hadoop-0.5.1-SNAPSHOT
hadoop-nutch-common-deps	jetty-5.1.4,lucene-misc-1.9.1,lucene.core-1.9.1,commons-cli-2.0-SNAPSHOT,commons-logging-1.0.2,log4j-1.2.13
nutch-deps	concurrent-1.3.4,commons-lang-2.1,oro-2.0.4
protocol-http	lib-http-0.9.0-SNAPSHOT.jar,protocol-http-0.9.0-SNAPSHOT
scoring-opic	scoring-opic-0.9.0-SNAPSHOT
urlfilter-prefix	urlfilter-prefix-0.9.0-SNAPSHOT
nutch-osgi-adapter	Contains custom code required to run nutch inside OSGi container

Identified gluecode (so far)

Configuration] Acts as a Decorator for [http://lucene.apache.org/hadoop/docs/api/org/apache/hadoop/conf/Configuration.html object. Adjusts classpath so that Configuration object can find the configuration files from inside bundle (hadoop-default.xml, hadoop-site.xml, nutch-default.xml, nutch-site.xml)|

PluginHelper Listens for bundle activations and registers plugin-bundles as plugins into nutch plugin system (osgi plugins cannot depend on any non OSGi plugin)

PluginDescriptor] Adapts bundle to [http://lucene.apache.org/nutch/nutch-nightly/docs/api/org/apache/nutch/plugin/PluginDescriptor.html

Extension] Adapts bundle to [http://lucene.apache.org/nutch/nutch-nightly/docs/api/org/apache/nutch/plugin/Extension.html

Plugin] Adapts bundle to [http://lucene.apache.org/nutch/nutch-nightly/docs/api/org/apache/nutch/plugin/Plugin.html

Some observations, ideas during the (still continung) trip

Hadoop Configuration should really be made Interface (HADOOP-24) and some other configuration method (but files inside jars) should be invented so we can separate nutch and hadoop to two bundles. This would allow us to run different configurations (different versions of each package/bundle) more easily.

Current nutch script propably need to be (re)implemented with java as there will propably be one front door to enter (start) OSGi nutch

It would be nice if lucene and hadoop were "natively" build as osgi bundles, all this requires is a custom manifest inside .jar. It would also be nice if lucene family would automatically build and deploy packages to m2 repositories.

Related reading

- Message in hadoop-dev mailing list.

- Cocoon goes OSGi