SeDiCI
SERVICIO DE DIFUSIÓN
DE LA CREACIÓN INTELECTUAL

Instalación de Dspace

Deploy de un repositorio básico con Dspace 3.0

Bloque 2.2 - Contenido

1. Instalación de requisitos
2. Instalación de Dspace
   - Descarga, Compilación y empaquetado
   - Instalación
   - Activación de xmlworkflow, discovery
   - Deploy
3. Uso desde consola: comando Dspace y Cronjobs
4. Backup
5. Dspace en ejecución: logs, jvm
6. Entorno de debug

Requisitos (1) - PostgreSQL

- Se puede utilizar PostgreSQL 8.4+ y Oracle 10g+

- Instalación:

    debian* --> apt-get install postgresql

Configuración: /etc/postgresql/x.y/main/

- Configuración general: postgresql.conf
- Reglas de autorización de usuarios por origen: pg_hba.conf

Tip: Creación de un superusuario

```
sudo -u postgres createuser --superuser $USER
sudo -u postgres psql
#\password $USER
```

Requisitos (2) - java JDK

- Dspace 3.x: OpenJDK 6, OpenJDK 7, Oracle Java 6 u Oracle Java 7
  - apt-get install **openjdk-6-jdk**
- Dspace <= 1.8.2: Oracle Java 6

La licencia de Oracle no permite la redistribución de los paquetes java*. Para instalar Oracle Java 6 o7, es necesario descargarlos y compilarlos.

- Alternativa 1 (http://www.webupd8.org/2012/11/oracle-sun-java-6-installer-available.html):

  sudo add-apt-repository ppa:webupd8team/java
  sudo apt-get update
  sudo apt-get install oracle-java6-installer

- Alternativa 2 (https://github.com/flexiondotorg/oab-java6)

  *sudo apt-get purge sun-java**
  mkdir ~/src && cd ~/src
  git clone https://github.com/flexiondotorg/oab-java6.git
  cd ~/src/oab-java6 && sudo ./oab-java.sh
  sudo apt-get install sun-java6-plugin sun-java6-jre sun-java6-bin sun-java6-jdk
  #LOG: tail -f ~/src/oab-java6/oab-java.sh.log

Requisitos (3) - WebContainer

Dspace funciona sobre Jetty, Tomcat, Caucho Resin o casi cualquier web container.

- Instalación: apt-get install tomcat
- Configuración
  - /etc/default/tomcat7

    JAVA_HOME="/usr/lib/jvm/java-6-oracle" #Si no se usa la OpenJDK
  - /etc/tomcat7/server.xml

<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000" URIEncoding="UTF-8" redirectPort="8443" />

Requisitos (4) - Otras dependencias

- Maven: compilación, empaquetado,filtrado de archivos y generación del paquete de instalación.

  apt-get install maven

  # maven2 o maven3

- Ant: automatización de procesos de instalación y actualización.

  apt-get install ant ant-contrib

- Git: versionado distribuido de codigo fuente

  apt-get install git

Instalación (1) - Descarga

Versión reducida (dspace-release-3.0-rc3):

- suficiente para personalización de temas
- Recomendado para instalaciones planas con personalizaciones de temas y traducciones

Versión completa (dspace-source-3.0-rc3)

- contiene todos los módulos de dspace para compilar
- permite modificar cualquier módulo
- Recomendado para instalaciones más complejas, con redefinición de módulos o directa del core

Es posible descargar la versión completa del HEAD desde github https://github.com/DSpace/DSpace/archive/master.zip o de un release/rc desde sourceforge http://sourceforge.net/projects/dspace/files/

Instalación (2) - Usuarios

- Crear un usuario para tomcat/dspace
  - adduser dspace
  - es conveniente vincular los usuarios de tomcat y dspace para evitar **conflictos de permisos**:
    - con mismo grupo y umask: (no funciona con ubuntu y tomcat)
    - con mismo username: menos elegante pero sí funciona.

$TOMCAT_USER, $TOMCAT_GROUP

sudo find -L /var/lib/tomcat7/ -user tomcat7 -exec chown dspace {} \; -print

sudo find -L /var/lib/tomcat7/ -group tomcat7 -exec chgrp dspace {} \; -print

- Creación del user en postgres y base de datos
  - sudo -u postgres createuser -d -R -S -P dspace
  - sudo -u dspace createdb -E UNICODE dspace

Instalación (3) - Creación de directorios

- Descomprimir en ${dspace-src}
  - tar -xzf dspace-release-3.0-rc3.tar.gz

- Directorio de instalación ${dspace.dir}
  - ejemplos: /opt/dspace , /var/dspace , /home/dspace

    mkdir ${dspace.dir}

    sudo chown dspace.dspace ${dspace.dir}
  - por default alojará casi todos los datos de dspace: assetstore, logs de aplicación, datos de solr
  - Los datos de postgres, logs de tomcat y apache2, quedan afuera.

Instalación (4) - build.properties

- A partir de **3.0**
  - simplifica el proceso de desarrollo
  - posibilidad de usar múltiples entornos
    - -Denv=dev --> dev.properties
  - para compilación. Luego no existe más
  - Atención: **no** eliminar properties
- Editar [dspace-source]/build.properties

```
dspace.dir = /var/dspace
dspace.hostname = localhost
dspace.baseUrl = localhost:8080
dspace.name  = Mi Repositorio de pruebas
solr.server = localhost:8080/solr
default.language = es
...
db.url=jdbc:postgresql://localhost:5432/dspace
db.username=dspace
db.password=dspace
```

Instalación (5) - Compilación, empaquetado e instalación

- **Compilación y empaquetado maven**
  - cd [dspace-src]
  - mvn package
- **Instalación vía ant**
  - cd [dspace-source]/dspace/target/dspace-3.0-rc3-build
  - ant fresh_install
- **Revisar permisos de ${dspace.dir}**
  - chmod -R ug+rw,o-w log assetstore upload solr exports reports
  - chown -R dspace ${dspace.dir}
- **Crear usuario administrador**
  - ${dspace.dir}/bin/dspace create-administrator

TIP: para evitar conflictos de permisos, se recomienda utilizar siempre el usuario dspace para todas las tareas.

Activación de XMLWorkflow

- Actualización del Schema de la BBDD
  - *dspace dsrun org.dspace.storage.rdbms.InitializeDatabase etc/postgres/xmlworkflow/xml_workflow.sql*
  - *dspace dsrun org.dspace.storage.rdbms.InitializeDatabase etc/postgres/xmlworkflow/workflow_migration.sql*
- Habilitación
  - workflow: config/modules/workflow.cfg

    - workflow.framework=originalworkflow

    + workflow.framework=xmlworkflow
  - Aspectos: config/xmlui.xmap

    - <aspect name="Original Workflow" path="resource://aspects/Workflow/" />

    + <aspect name="XMLWorkflow" path="resource://aspects/XMLWorkflow/" />
- Configuración
  - config/workflow.xml

https://wiki.duraspace.org/display/DSDOC3x/Configurable+Workflow

Activación de Discovery (1)

- Configuración del módulo
  - modules/discovery.cfg
    - solr.search.server = http://localhost:8080/solr/search
- Habilitar Aspecto XMLUI
  - sitemap.xmap
    - <aspect name="Discovery"
      path="resoaurce://aspects/Discovery/" />
- Agregar Event Listeners
  - dspace.cfg >

    event.dispatcher.default.consumers =

    versioning, search, browse, **discovery**, eperson, harvester

Activación de Discovery (2)

- Deshabilitar últimos submissions de dspace
  - dspace.cfg >
    - recent.submissions.count = 0

- indexar los documentos por primera vez (reindexar)
  - ./bin/dspace update-discovery-index

https://wiki.duraspace.org/display/DSDOC3x/Discovery#Discovery-EnablingDiscovery

Instalación (6) - Alternativas de Deploy

1. Copiar cada aplicación ${dspace.dir}/webapps/* en ${tomcat.dir}/webapps/
2. Cambiar el appbase de tomcat a ${dspace.dir} /webapps/
3. Crear contextos para cada aplicación en server.xml

   <Context path="/xmlui" docBase="${dspace.dir}/webapps/xmlui" debug="0" reloadable="true" cachingAllowed="false" allowLinking="true"/>
4. Crear link simbólicos para cada aplicación (+simple)

   ln -s ${dspace.dir}/webapps/xmlui ${tomcat.dir}/webapps/xmlui

   ln -s ${dspace.dir}/webapps/oai ${tomcat.dir}/webapps/oai

   ln -s ${dspace.dir}/webapps/solr ${tomcat.dir}/webapps/solr

Si se desea acceder a partir de /, debe reemplazarse el directorio ROOT de ${tomcat.dir}/webapps/ROOT

Instalación (7) - Dspace en puerto 80

1. Cambiar el puerto del conector HTTP de tomcat
   a. + simple
   b. - configuración limitada
2. Configurar un proxy reverso desde un servidor web

Ejemplo con apache2, mod_proxy y mod_proxy_ajp:

a. Habilitar ajp en tomcat (server.xml):
   <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
b. Habilitar módulos en apache2: "*a2enmod proxy proxy_ajp*"
c. Deshabilitar forward proxy: "*ProxyRequests Off*"

A) Con Location

```
<Location />
  ProxyPass ajp://localhost:8009/ retry=10
  ProxyPassReverse ajp://localhost:8009/
</Location>
```

B) Con mod_rewrite

```
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ ajp://localhost:8009/$1 [P]
```

Comando *dspace*, *Cronjobs*

Comando *dspace*

- Script shell (*#!/bin/sh*)
  - Inicia una nueva instancia de la JVM
  - Utiliza sus propios parámetros de VM: tamaño de pila, PermGen, etc
  - incluye el directorio **{dspace.dir}/lib** en el classpath
- Uso
  - ./dspace *tarea parametros*
  - ./dspace --help
  - ./dspace *tarea* --help

Comando *dspace*
Orden: create-administrator

- Se usa para crear un usuario Administrador en el sistema
- Debe invocarse luego de la instalación para crear el primer usuario en el sistema

Comando *dspace*
Orden: curate

Ejecuta una "curation task" para realizar algún tipo de análisis o modificación sobre los ítems

Puede aplicarse sobre:

- Repositorio completo
- Una comunidad específica
- Una colección específica
- Un ítem específico

Comando *dspace*
Orden: curate, ejemplos

Ejemplos de curation tasks

- Verificación de links muertos
- Validaciones de integridad de datos
- Análisis de formatos de archivos usados
- Análisis de los archivos en busca de virus
- etc.

Comando dspace
Órdenes para estadísticas

- Se utilizan para recopilar información estadística de acceso, descargas, etc.
- Se realiza un análisis de los logs de DSpace *${dspace.dir}/logs/stats-*.log*
- Existen múltiples comandos asociados: stat-general, stat-initial, stat-monthly, stat-report-general, stats-utils, etc

Comando dspace
Orden: update-discovery-index

- Actualiza el índice de búsquedas de Apache Solr (**/search**)

- Se utiliza cuando es necesario indexar por primera vez el repositorio o cuando se desea reindexarlo todo.

Comando dspace
Orden: oai

- Controla el módulo OAI 2.0
  - import:
    - Actualiza el índice de Apache Solr (**/oai**)
    - documentos indexados en /oai
    - necesaria periódicamente
    - ejecución frecuente (segúnel movimiento del repositorio)
  - clean-cache
    - vacía la cache
    - necesario ante cambios en mapeos XSL

Comando dspace
Orden: dsrun

- Ejecuta una clase parametrizable
  - ${dspace.dir}/bin/dspace dsrun *fqcn arguments*
    - *fqcn es el nombre de la clase completo (incluido el package) que implementa* un método main()
- Permite definir cualquier tipo de clase para luego ejecutarlas desde la línea de comandos. Ejemplos:
  - import/export de registries
    - ./bin/dspace dsrun org.dspace.administer.MetadataImporter -f config/registries/sedici-metadata.xml
  - scripts de Base de datos
    - ./bin/dspace dsrun org.dspace.storage.rdbms.InitializeDatabase xml_workflow.sql

Comando dspace
Otras órdenes

- **checker**: para verificar la integridad del assetstore
- **import/export:** Importación/ Exportación de ítems o colecciones
- **filter-media**: aplica el proceso de mediafilter sobre un conjunto específico de items
- **harvest**: gestiona las cosechas de las colecciones cuyos datos son recolectados vía OAI-PMH
- **metadata-export/metadata-import**: permite importar/exportar un csv con metadatos
- Muchos más: gestión de handle, índices, discovery, usuarios, importación de usuarios y colecciones, etc

Cronjobs (1) - Acciones sobre bitstreams

- IMPORTANTE: Las tareas deben ejecutarse con el usuario de dspace. No usar root

  crontab -u dspace -e

- Procesamiento con plugins de mediafilter para:
  - generación de thumbnails en base a la primer página de los pdf
  - generación de thumbnails a partir de imágenes JPEG
  - extracción de texto de archivos pdf, word, powerpoint, html

  ${dspace.dir}/bin/dspace filter-media
- Chequeo de checksums y reporte

  ${dspace.dir}/bin/dspace checker -lp

  ${dspace.dir}/bin/dspace checker-emailer -c

Cronjobs (2) - Actualizaciones diarias

- OAI
  - Actualización de datos del core OAI de solr

${dspace.dir}/bin/dspace oai import

- **Embargo-Lifter**
  - Revisa los ítems que tienen fecha de fin de embargo y *levanta el embargo*

${dspace.dir}/bin/dspace embargo-lifter

- Limpieza de la BBDD

vacuumdb --analyze dspace > /dev/null 2>&1

- Sitemaps
  - Actualización de archivos de sitemap (para crawlers)

${dspace.dir}/bin/dspace generate-sitemaps

Cronjobs (3) - Acciones para estadísticas

- Análisis diario de accesos

  ${dspace.dir}/bin/dspace stat-general
  ${dspace.dir}/bin/dspace stat-monthly
  ${dspace.dir}/bin/dspace stat-report-general
  ${dspace.dir}/bin/dspace stat-report-monthly

- Actualización de ips de spiders
  - descarga listados de de ips de internet para poder filtrarlos en los logs

  ${dspace.dir}/bin/dspace stats-util -u

- Marcado de accesos de spiders
  - para poder distinguirlos en las estadísticas

  ${dspace.dir}/bin/dspace stats-util -m

Cronjobs (3) - Acciones para estadísticas

- GEOLITE
  - Actualización de BBDD de geolocalización para estadísticas

ant -f ${dspace-src}/distribution/target/dspace-.../build.xml update_geolite

- Envío de mails de suscripción
  - de suscripciones de usuarios a colecciones

${dspace.dir}/bin/dspace sub-daily

Backups (1) - Alternativa 1: AIP

- AIP (Archival Information Package)

*${dspace.dir}/bin/dspace packager **--disseminate** -a **-t AIP** -e arieljlira@gmail.com -u -i 10915/0 /root/aip-site.zip*

- Archivos de datos
  - ${dspace.dir}/var

- Configuración y logs
  - ${dspace.dir}/config y ${dspace.dir}/logs

Backups (2) - Alternativa 2, Dump+rsync

- Archivos de datos
  - ${dspace.dir}/assetstore
  - ${dspace.dir}/var
- Configuración y logs
  - ${dspace.dir}/config y ${dspace.dir}/logs !!!
- Postgres
  - Dump

pg_dump -h localhost -U {dbuser} -a -b -x -O -f {output_file.tar} -F t {dbname}

#Pasar el dump a SQL, para que sirva en cualquier versión de PostgreSQL

pg_restore -Ft -O -f {output_file.sql} {output_file.tar}

- Restore

dropdb {dbname}

createdb {dbname}

psql -h localhost -U {dbuser} -d {dbname} -f {output_file.sql}

Backups (3) - Caminos posibles

- Backups periódicos
  - AIP
  - archivos
  - configuraciones y logs
  - solr
- Herramientas complementarias
  - Bacula / backuppc / backupninja / etc
  - Backups externos: AmazonS3, Duracloud, etc
- Ejemplo de Plan de backup
  - backup de AIP semanal
  - backup rsync + dump **onserver** c/12 hs
  - backup rsync + dump **offserver** c/24hs
  - backup offsite semanal aip+rsync+dump

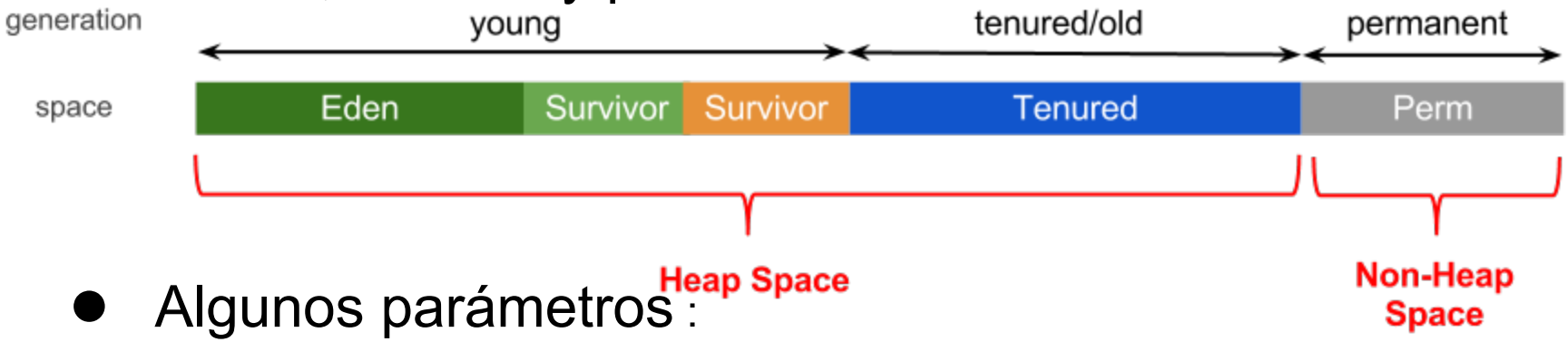
Dspace en Ejecución

Herramientas para mantenimiento

- Postgres
  - pgadmin3
  - phppgadmin
- Detección de excepciones
  - monitoreo de dspace.log y catalina.*
  - reporte de excepciones
- Monitoreo del servicio
  - watchdog (interno,estado del servidor)
  - monit (interno, estado de servicios)
  - pingdom, nagios, etc (externo)

Uso de Memoria en la JVM

- La jvm aloca memoria del SO y gestiona internamente la liberación y alocación de memoria para objetos, clases, threads y para sí misma.



- Algunos parámetros :
  - Tamaño de heap: Máximo (-**Xmx**1548m) , mínimo inicial(-**Xms**512m)
  - Tamaño de Perm space: Máximo (**-XX:MaxPermSize**), mínimo inicial **-XX:PermSize**=128m
- Selección del GC:
  - -XX:+UseSerialGC --> aplicaciones muy chicas
  - -XX:+UseParallelGC
  - -XX:+**UseConcMarkSweepGC**

Uso de Memoria en Dspace

- En el webcontainer
  - /etc/defaults/tomcat7
    - JAVA_OPTS=... -**Xmx**1548m **-XX:PermSize**=128m ..."

- En las tareas ejecutadas desde consola
  - Handle Server:
    - bin/start-handle-server
  - Comando dspace y Cronjobs
    - bin/dspace

- Es posible predefinir JAVA_OPTS antes de cada tarea

Causas típicas de OutOfMemory

- OutOfMemoryError: PermGen space
  - PermSize muy chico
- OutOfMemoryError: Java heap size
  - Xmx muy chico
  - Loops en código
  - Recursiones muy largas con creación de muchos objetos
  - Uso en exceso de variables estáticas
- Otros posibles, menos frecuentes
  - Demasiados threads
  - Arrays gigantes
- Herramientas de análisis
  - livianas y free: jvmstat+visualgc, jmap+jhat
  - completas y pagas: yourkit y muchisimas más.

http://blog.codecentric.de/en/2010/01/the-java-memory-architecture-1-act/
http://javarevisited.blogspot.com.ar/2011/05/java-heap-space-memory-size-jvm.html
http://java-source.net/open-source/profilers

Logging - Log de aplicación vía log4j

- Configuración
  - config/log4j.properties
  - log4j-handle-plugin.properties

- Appenders preconfigurados
  - $dspace.dir/log/dspace.log
  - $dspace.dir/log/checker.log
  - $dspace.dir/log/cocoon.log
  - $dspace.dir/log/handle-plugin.log

Logging - Log de aplicación vía log4j

- Log personalizado, ejemplo

log4j.logger.*ar.edu.unlp.sedici=INFO, A4*
log4j.logger.*ar.edu.unlp.sedici.xyz=WARN*
log4j.additivity*.ar.edu.unlp.sedici=false*

log4j.appender*.A4=org.dspace.app.util.DailyFileAppender*
log4j.appender*.A4.File=${dspace.dir}/log/sedici.log*
 # yyyy-MM-DD for daily log files, or yyyy-MM for monthly files
log4j.appender*.A4.DatePattern=yyyy-MM*
log4j.appender*.A4.MaxLogs=3*
log4j.appender*.A4.layout=org.apache.log4j.PatternLayout*
log4j.appender*.A4.layout.ConversionPattern=%d %-5p %c @ %m%n*

Logging - Logs de tomcat y otros

Tomcat

- Logs predefinidos: /var/log/tomcat7
  - catalina.out
  - catalina.yyyy-mm-dd.log
  - localhost.yyyy-mm-dd.log

- Configuración
  - logging.properties
  - server.xml (Access log, opcional)

```
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
    prefix="localhost_access_log." suffix=".txt" pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" />
```

  - Para mejorar el logging, es posible habilitar **log4j**

Entorno de desarrollo

- Eclipse 3.5 (Galileo) o superior.
- Plugins:
  - Egit
  - m2e Maven integration for Eclipse
  - m2e-Egit Maven SCM handler for EGit (opcional)
- Clone del repositorio Dspace:
  a. EGit>clone Git repository
     i. git://github.com/DSpace/DSpace.git
     ii. seleccionar branch master
  b. importar proyectos maven:
     i. con Maven SCM connector for EGit: EGit>import maven proyects
     ii. sin Maven SCM connector: Java> import existing maven proyects

Lectura de stacktraces

- Elementos fundamentales
  - Cause
  - Clase que genera el error y número de línea
- Información extra, XMLUI
  - URL causante
  - parámetros de request
  - usuario causante e info de sesión
  - ip de origen

- Ver ejemplo

...

Muchas gracias

Dudas y comentarios?

alira@sedici.unlp.edu.ar
nestor@sedici.unlp.edu.ar
marisa.degiusti@sedici.unlp.edu.ar