Technology Transfer Partners

Installing Search Service

version 2.1

(updated Dec 2022 for Solr v9.1.0)
updated Feb 2024 for Solr changes)

Joe Doupnik

jrd@netlab1.net

jdoupnik@microfocus.com

Mindworksuk and Micro Focus

Prof (ret.) Univ of Oxford

MICRO FOCUS

Installing Apache Solr itself

Installation of Solr proper consists of

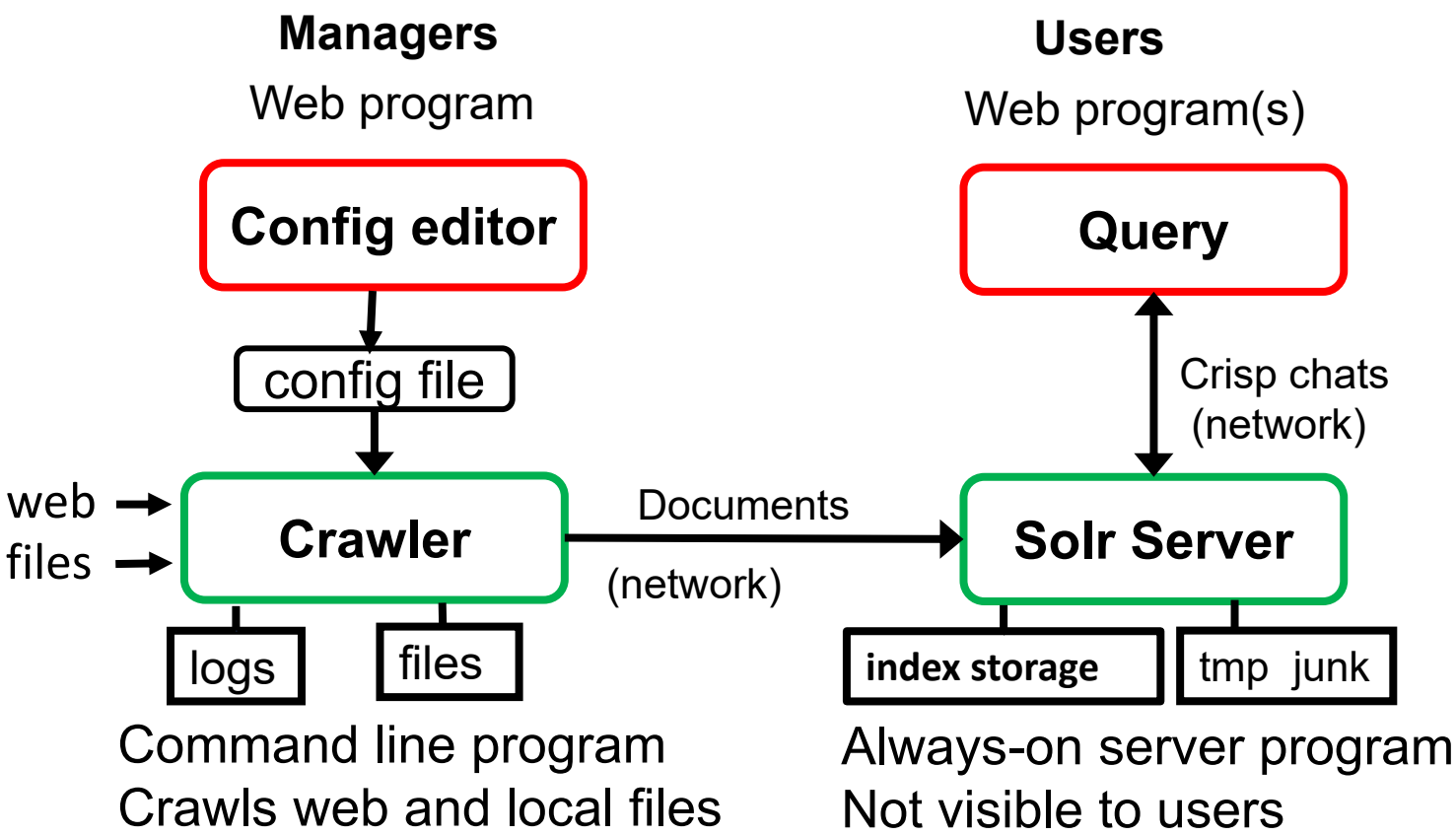
- create a Solr/Java user name, recommend is solr:users
- optionally install required JDK v1.8 RPM from Oracle or use openjdk
- fetch and unpack Solr from Apache.org
- edit three Solr files
- create directory “data” to hold indices
- add my schema bundle to “data/configsets” or make your own version (instructions indicated later in these slides)
- create a “tmp” Java work directory
- make all these be owned by the Solr username

I put everything into area /**home/search** in this discussion

This material covers using PHP v5.3 through v7.x, Apache 2.2 & 2.4 (pre-fork and threaded) and Solr v6 through v8.

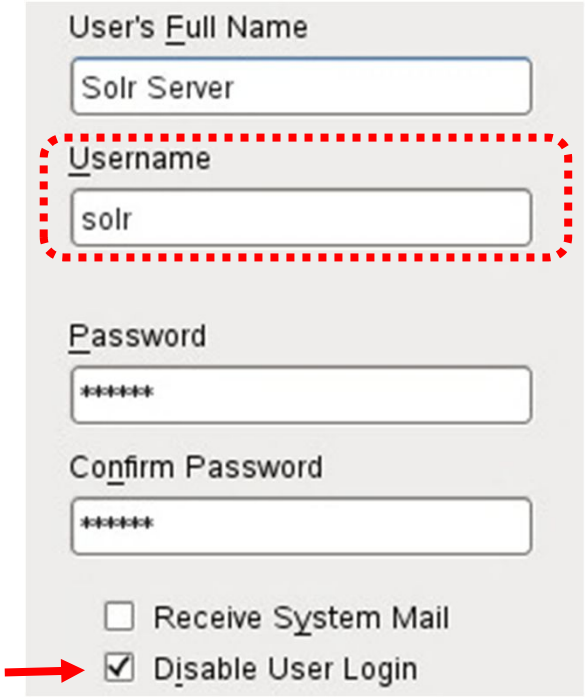
BEWARE: Solr v9 has many major changes which are loosely accommodated here. I recommend using Solr v8.11.1 for now. See notes futher along about adapting to Solr v9.0.0.

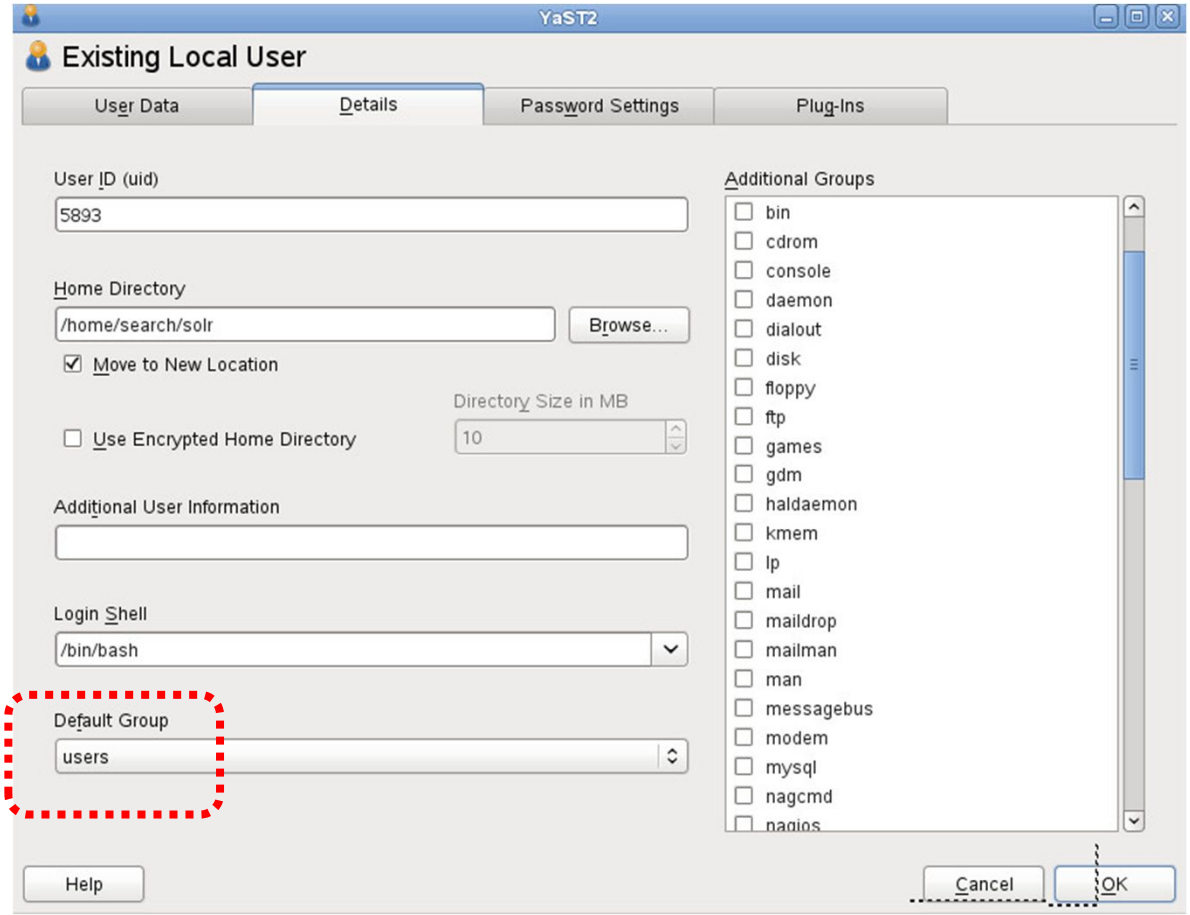
Topology of the search service



The system allows for multiple Solrs, crawlers, query programs, and multiple Solr schema. This is not a monolith.

Create a Solr username with YaST





solr:users

Note: I use /home/search to hold Solr and related components. Change that to suit your requirements. User solr will need to own the area. The result is username **solr** in primary group **users**.

Option: Fetch Oracle Java JDK v1.8 RPM

Java SE Development Kit 8u121

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.86 MB	jdk-8u121-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.83 MB	jdk-8u121-linux-arm64-vfp-hflt.tar.gz
Linux x86	162.41 MB	jdk-8u121-linux-i586.rpm
Linux x86	177.13 MB	jdk-8u121-linux-i586.tar.gz
Linux x64	159.96 MB	jdk-8u121-linux-x64.rpm
Linux x64	174.76 MB	jdk-8u121-linux-x64.tar.gz
Mac OS X	223.21 MB	jdk-8u121-macosx-x64.dmg
Solaris SPARC 64-bit	139.64 MB	jdk-8u121-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.07 MB	jdk-8u121-solaris-sparcv9.tar.gz
Solaris x64	140.42 MB	jdk-8u121-solaris-x64.tar.Z
Solaris x64	96.9 MB	jdk-8u121-solaris-x64.tar.gz
Windows x86	189.36 MB	jdk-8u121-windows-i586.exe
Windows x64	195.51 MB	jdk-8u121-windows-x64.exe

Install via command rpm -i jdk-8u121-linux-x64.rpm
Results go into /usr/java/jdk-8u121 (or similar version)

SLES 12 and later have the jdk and do not need this step.

Fetch Solr from http://lucene.apache.org

Solr 8.5.0

Solr 8.5.0 is the most recent Apache Solr release.

- Source release: solr-8.5.0-src.tgz [PGP] [SHA512]
- Binary releases: solr-8.5.0.tgz [PGP] [SHA512] / solr-8.5.0.zip [PGP] [SHA512]
- Change log

Solr-8.5.0.tgz is the current release in this note

Unpack as cd /home/search

```
tar xzvf  solr-8.5.0.tgz
mv  solr-8.5.0  solr
```

or create a symlink ln -s solr-8.5.0 solr

Moving Solr files about

Three Solr files are involved

cd /home/search/solr/bin

1. cp init.d/*solr* /etc/init.d (system start/stop script)
2. cp *solr.in.sh* /etc/default (sets Java & Solr defaults)

3. vi *solr* near line 1718 find the lsof line saying

running=`lsof -PniTCP:$SOLR_PORT -sTCP:LISTEN`

Change it to comment out the -sTCP:LISTEN portion:

running=`lsof -PniTCP:$SOLR_PORT ###WAS -sTCP:LISTEN`

(retain the back tics in this line)

SLES 12 and later do not need this lsof change

Edit and enable file /etc/init.d/solr

vi /etc/init.d/solr, add two lines near line 51

```
SOLR_ENV="/etc/default/solr.in.sh"
##JRD set unlimited virtual address space for large indices
ulimit -v unlimited
```

Enable the Solr service:

```
cd /etc/init.d
chmod a+x solr
chkconfig -a solr
```

Edit file /etc/default/*solr.in.sh*

Add these lines to the end of the file, as overrides

SOLR_ULIMIT_CHECKS = false — **For Solr 8.5 et seq**

#JRD give more memory

SOLR_HEAP="2500m" — See following slides for more detail (value increased on Feb 2024)

##JRD enlarge this

#SOLR_OPTS="$SOLR_OPTS -Xss512k"

SOLR_OPTS="$SOLR_OPTS -Xss1024k"

SOLR_STOP_WAIT=30

SOLR_JAVA_HOME="/usr/java/jdk-1.8.0-<version>/" — **Where JDK resides** ***(Omit for SLES12)***

SOLR_PID_DIR="/home/search/solr" — **Where Solr lives**

SOLR_HOME="/home/search/solr/data" — **Where our indices will be stored**

SOLR_LOGS_DIR="/home/search/solr/logs" — **Solr as a whole logs**

SOLR_PORT="8983"

SOLR_OPTS="$SOLR_OPTS -Dsolr.autoSoftCommit.maxTime=3000"

SOLR_OPTS="$SOLR_OPTS -Dsolr.autoCommit.maxTime=60000"

Important locations

SOLR_OPTS="$SOLR_OPTS -Djava.io.tmpdir=/home/search/tmp" — **Java work directory**

SOLR_OPTS="$SOLR_OPTS -Dlog4j2.formatMsgNoLookups=true" — **Log4j blunder mitigation**

Add **Java GC_TUNE** material, discussed next

GC_TUNE addition for /etc/default/solr.in.sh

```
## GC_TUNE below is from Solr v8.1 and still works with v8.0
GC_TUNE=" \
-XX:SurvivorRatio=4 \
-XX:TargetSurvivorRatio=90 \
-XX:MaxTenuringThreshold=8 \
-XX:+UseConcMarkSweepGC \
-XX:ConcGCThreads=4 -XX:ParallelGCThreads=4 \
-XX:+CMSScavengeBeforeRemark \
-XX:PretenureSizeThreshold=64m \
-XX:+UseCMSInitiatingOccupancyOnly \
-XX:CMSInitiatingOccupancyFraction=50 \
-XX:CMSMaxAbortablePrecleanTime=6000 \
-XX:+CMSParallelRemarkEnabled \
-XX:+ParallelRefProcEnabled \
-XX:-OmitStackTraceInFastThrow“
```

From Solr v8.2.0 file CHANGES.txt:

SOLR-13394: The default GC has been changed from CMS to G1. To override this (in order to switch to CMS or any other GC), use GC_TUNE section of bin/solr.in.sh or bin/solr.in.cmd

The myconf.tar.gz bundle has file *solr.in.sh.example*. Saves some typing.

Important nuance for Solr versions

At May 2019 the current Solr was v8.1. It has a large memory consumption problem due to new Java garbage collection algorithm G1. Algorithm GC_TUNE avoids the problem. Test is shown below.

Test is index a set of 80 .pdf and .pptx documents, 4GB SOLR_HEAP

memory consumption

Solr v8.0, no GC_TUNE, 1.7GB idle, 1.9GB indexing, 2.92 minutes to finish

Solr v8.0, with GC_TUNE, 1.1GB idle, 1.3GB indexing, 2.97 minutes

Solr v8.1, no GC_TUNE, 4.3GB idle, 4.4GB indexing, 1.67 minutes

Solr v8.1, with GC_TUNE, 1.0GB idle, 1.3GB indexing, 1.53 minutes

Solr v8.1 contains the new garbage collection algorithm G1 which can be overridden, for now, by the GC_TUNE addition just shown.

See also https://openjdk.java.net/jeps/34

http://mail-archives.apache.org/mod_mbox/lucene-solr-user/201905.mbox/browser for 25 May 2019

https://www.azul.com/files/Understanding_Java_Garbage_Collection_v4.pdf

Solr v9 – adapting to it 1/4

The v9.0.0 release of Solr is much changed from v8 and has shown many problems at my place.

However, I have found a working solution for the time being, which involves using *managed-schema* from v8.11.1 together with solrco*nfig.xml* from v9.0.0. The same applies to v9.1.0.

Files were chosen from solr 9.0.0

/server/solr/configsets/conf/sample_techproducts_configs/conf

Tests on modified Solr v9.0.0 with both existing Solr v8.11.1 indices and newly created v9 indices with these changes show success, including spell checking. Backward compatibility is maintained.

The stock v9 *managed-schema* revealed many problems which I have not solved. This includes Tika failing to parse .odp files, a problem absent in my changed material. This is a Solr project management problem.

Adapted Solr v9.0.0 configuration 2/4

Diff between the Solr v8.11.1 stock distribution (<) ***managed-schema*** and my (>) adapted version

```
26c26
<   <fieldType name="location" class="solr.LatLonType" subFieldSuffix="_coordinate"/>
---
>   <!-- JRD OMIT  <fieldType name="location" class="solr.LatLonType" subFieldSuffix="_coordinate"/> -->
448c448,453
<
---
> <!--JRD addition -->
>    <field name="date" type="text_general" indexed="true" stored="true"/>
>    <field name="url" type="string" indexed="true" stored="true"/>
>    <field name="author" type="string" indexed="true" stored="true"/>
>    <field name="subject" type="string" indexed="true" stored="true"/>
> <!-- JRD end addition -->
513c518
<   <dynamicField name="*_p" type="location" indexed="true" stored="true"/>
---
>   <!-- JRD OMIT  <dynamicField name="*_p" type="location" indexed="true" stored="true"/> -->
```

Distribution file is from solr-8.11.1/example/files/conf

Adapted Solr v9.0.0 configuration 3/4

Diff between Solr v9.0.0 stock (<) distribution ***solrconfig.xml*** and my modifed version of it (>)

Handler /select

```
743a744,749
> <!-- JRD ADDITION -->
>         <arr name="last-components">
>         <str>spellcheck</str>
>         </arr>
> <!-- JRD END -->
>
```

Handler /query

```
772,773c778,783
<     <lst name="defaults">
<       <str name="df">text</str>
---
>         <lst name="defaults">
> <!-- JRD WAS
>      <str name="df">text</str>
>  -->
> <!-- JRD REPLACEMENT -->
>      <str name="df">content</str>
```

spellcheck

```
838c852,854
<       <str name="field">text</str>
---
>       <!-- JRD WAS      <str name="field">text</str> -->
>       <str name="field">content</str> <!-- JRD replacement -->
>
```

enable dedup

```
1231c1247
<   <!--
---
>   <!-- JRD ENABLE -->
1242c1258
<     -->
---
>      <!-- END ENABLE    -->
```

enable langid

```
1244c1260
<   <!-- Language identification
---
>      <!-- Language identification
1255c1271
<     <!--
---
>     <!-- JRD ENABLE -->
1265c1281
<     -->
---
>      <!-- END ENABLE    -->
```

Distirbution file is from solr-9.0.0/server/solr/configsets/sample-techproducts_configs/conf

Solr v9.0.0 modified file solr.in.sh 4/4

```
# JRD values
# These GC settings have shown to work well for a number of common Solr workloads
GC_TUNE=" \
-XX:+ExplicitGCInvokesConcurrent \
-XX:SurvivorRatio=4 \
-XX:TargetSurvivorRatio=90 \
-XX:MaxTenuringThreshold=8 \
-XX:+UseConcMarkSweepGC \
-XX:ConcGCThreads=4 -XX:ParallelGCThreads=4 \
-XX:+CMSScavengeBeforeRemark \
-XX:PretenureSizeThreshold=64m \
-XX:+UseCMSInitiatingOccupancyOnly \
-XX:CMSInitiatingOccupancyFraction=50 \
-XX:CMSMaxAbortablePrecleanTime=6000 \
-XX:+CMSParallelRemarkEnabled"

#JRD give more memory
##SOLR_HEAP="4096m"
SOLR_HEAP="2048m"
##JRD enlarge this
#SOLR_OPTS="$SOLR_OPTS -Xss512k"
SOLR_OPTS="$SOLR_OPTS -Xss1024k"
SOLR_STOP_WAIT=30
####SOLR_JAVA_HOME="/usr/java/latest/"
SOLR_JAVA_HOME="/usr/lib64/jvm/jre-11-openjdk"
SOLR_PID_DIR="/home/search/solr"
SOLR_HOME="/home/search/solr/data"
SOLR_LOGS_DIR="/home/search/solr/logs"
SOLR_PORT="8983"
SOLR_OPTS="$SOLR_OPTS -Dsolr.autoSoftCommit.maxTime=3000"
SOLR_OPTS="$SOLR_OPTS -Dsolr.autoCommit.maxTime=60000"
SOLR_OPTS="$SOLR_OPTS -Djava.io.tmpdir=/home/search/tmp"
SOLR_ULIMIT_CHECKS="false"
SOLR_OPTS="$SOLR_OPTS -Dsolr.sharedLib=server/lib"
SOLR_HOST=
SOLR_SOLRXML_REQUIRED=true
SOLR_OPTS="$SOLR_OPTS -Dsolr.dns.prevent.reverse.lookup=true"
```

SOLR v9 items

GC_TUNE values make a large reduction of memory consumption

This was used on SLES15 which has Java jre-11-openjdk. Install dir is /home/search. solr is a symlink to solr-9.x. Increase autosoftCommit to say 6000ms for slow to load files. This file lives in /etc/default/.

Solr v9 nuances and quirks

Solr v9 Tika declines to properly parse Open Desktop files (LibreOffice material). If we examine such files we find each is a zip archive of a collection of sundry files, thus they are difficult to compose into a single parsed result. We have two options as working solutions.

1. Export .odp & .odt files to .pptx or .pdf format and then index.
2. Use my “myconf.tar.gz” bundle which has Solr v8 managed-schema and which does properly parse the O-D files. This approach does work with Solr v9, but it does not include all of the v9 latest tweak nuances for other file types.

Errors include date+times not ending in Z (Zulu, aka GMT), and certain kinds of paragraphed text. Endless complexity.

Overall, this is a Solr project management problem which needs to be addressed by the product team.

Memory consumption nuance

This leaves us wondering if we use Solr v8.1 with its new Java garbage collector G1 then, because that GC consumes all allocated memory and returns none, we ask how much memory to give it via SOLR_HEAP.

Experiments provide the only safe indication. Change the SOLR_HEAP value, index (the major memory user), check for success, and beware of slow memory leaks for large indices.

Experiments with Solr v8.1 on my systems show SOLR_HEAP = “1024m” works satisfactorily only for awhile with no GC_TUNE settings. Smaller, say 512m, results in Solr failing to start.

For now a safe combination seems to be Solr v8.x (speed), with the GC_TUNE parameters (regular CMS GC), and SOLR_HEAP = “2048m” or larger (but not very large which might invoke algorithm changes yielding less performance).

Touch up /home/search, 12 quick steps

1. cd /home/search
2. mkdir **tmp logs** will hold Java work files & logs
3. chmod a+rwx tmp logs enable access by everyone
4. cd /home/search/solr
5. mkdir **data** will hold indices, can be elsewhere
6. cd data
7. cp /home/search/solr/server/solr/**solr.xml** .
8. mkdir **configsets** will hold schema bundles
9. cd configsets
10. tar xzvf path/**myconf.tar.gz** my Solr schema bundle "**myconf**"
    (modify the contents, as discussed next)
11. cd /home/search
12. chown -R solr:users /home/search

Locating indicies outside of Solr

If the index-holding directory “data” is not within Solr, unlike my examples, then ensure its location is stated correctly in file **/etc/default/*solr.in.sh***, property SOLR_HOME.

User solr:users will need to own that directory.

As that area receives many changes due to index creation it would be best located in a POSIX file system rather than encounter NSS Salvage operations.

Allow for growth. Example: a 660K email archive directory, 24GB source size, resulted in a 9.8GB index.

Layout of index-holding directory “data”

In /etc/default/solr.in.sh: SOLR_HOME="/home/search/solr/**data**“

Directory “**data**”

```
----------------------------------------------------------------
 |          |          | …                    |          |
(created automatically)                    (we set up these three:)
IndexA   IndexB  IndexC …                 configsets   solr.xml
 |          |          |                      |
core.props,                                  myconf ← From supplied bundle
  data                                          |
    |                                         conf
index  tlog                                     |
 |      |                                    files (the operational schema etc)
files   files
```

Dealing with Solr/Lucene changes

Original installation and later major version updates requires that we modify both *solrindex.xml* and *managed-schema*

What follows is how to accomplish that task, updated for Solr v8.x

Update the contents of

/home/search/solr/data/configsets/myconf/conf

with the contents of

/home/search/solr/example/files/conf

by saying:

1. cd /home/search/solr/
2. cp -R example/files/conf/* data/configsets/myconf/conf

Then make the three small changes shown in the next slides.

Changes to *solrconfig.xml* 1/2

About line 700 or so in the file –

```
<requestHandler name="/select" class="solr.SearchHandler">
  <!-- default values for query parameters can be specified, these
      will be overridden by parameters in the request
    -->
  <lst name="defaults">
    <str name="echoParams">explicit</str>
    <int name="rows">10</int>
    <!-- Default search field
      <str name="df">text</str>
     -->
    <!-- Change from JSON to XML format (the default prior to Solr 7.0)
      <str name="wt">xml</str>
     -->
</lst>
<-- JRD addition -->
    <arr name="last-components">
     <str>spellcheck</str>
    </arr>
<!-- end of addition -->
```

Boxed item is the change

Changes to *solrconfig.xml* 2/2, optional

About line 850 or so in the file --

```
<searchComponent name="spellcheck" class="solr.SpellCheckComponent">

   <str name="queryAnalyzerFieldType">text_general</str>

   <!-- Multiple "Spell Checkers" can be declared and used by this
       component
     -->

   <!-- a spellchecker built from a field of the main index -->
   <lst name="spellchecker">
     <str name="name">default</str>
<!--WAS      <str name="field">text</str> -->
      <str name="field">content</str> <!-- JRD replacement -->
```

This says where to find text to check spelling: where *content* is the body of a document.

Changes to *managed-schema* 1/1

At the end of the list of *<field names*= (line 417 or so) in file *managed-schema* add these lines:

```
<!-- JRD additions -->
<field name="date" type="pdate" indexed="true" stored="true" />
<field name="url" type="string" indexed="true" stored="true" />
<!-- end of additions -->
```

That's it for *solrconfig.xml* and *managed-schema* for Solr v7.x/v8.x. Changes are easier than might first seem.

Restart Solr, build and test an index.

Major Solr version updates (schema changes) may require our existing indices be rebuilt.

The query screen should offer date and title clickables which work, and spell checking should work.

Optional changes to solrconfig.xml

There are many options in that file. However, a few are obvious targets for experiments. These include

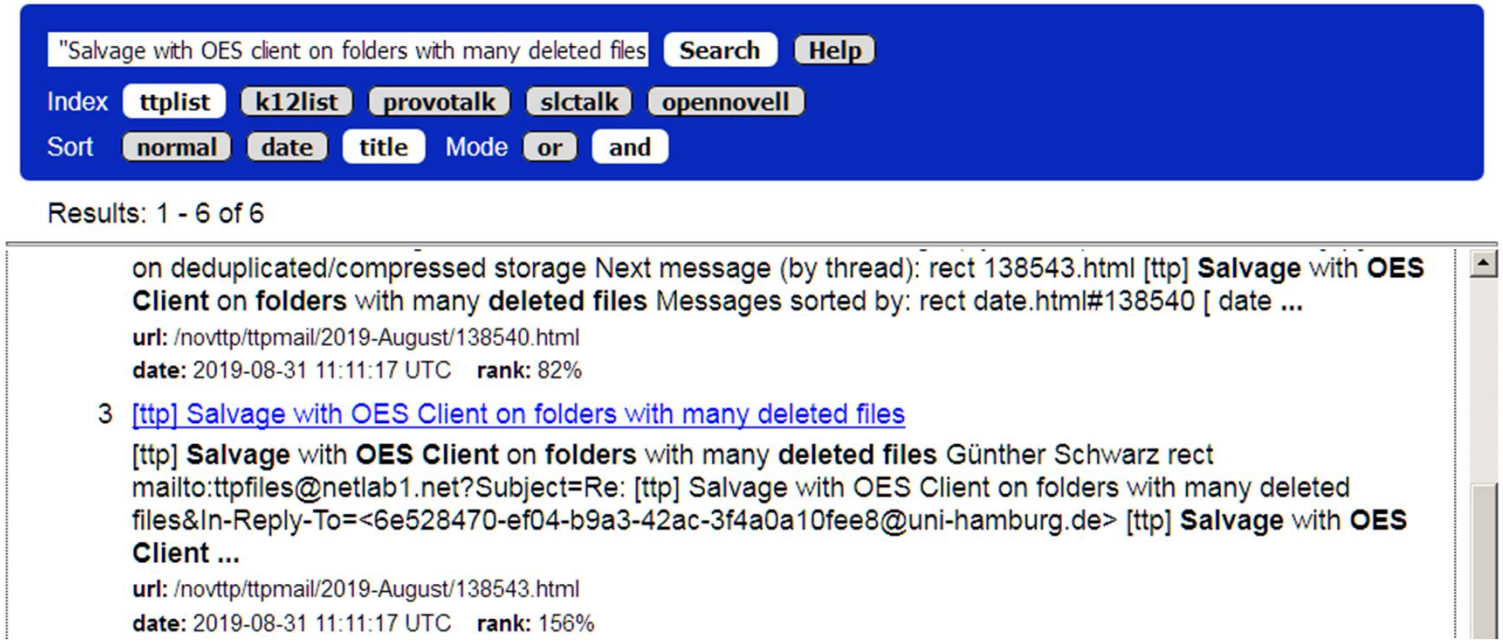
Merge Scheduler, Deduplication, Language identification

Details of such experiments are left to each site to investigate. One ought also consult Apache's Solr web pages to review current activities about them.

Myconf/conf/stopwords.txt

Configuration bundle myconf.tar.gz contains text file *stopwords.txt*, taken from two Internet sources.

Those words will be quietly omitted when searching, as if they were not present at all. Example (note the highlighting):



Note the bold versus normal words above.

Thus you may wish to edit the file to fit local preferences.

Small Solr/Lucene changes

Changing to a new major version number means we likely need to recreate the schema files and reindex everything, as just shown.

Simple Solr log grumbles can be dealt with by editing file /home/search/data/configsets/myconf/conf/*managed-schema.*

Spot the disturbing items, place brackets around each

```
<!--WAS  the original line resulting in grumbles -->
```

Such as these (complaints about obsolete LatLon and location):

```
<!--WAS  <fieldType name="location" class="solr.LatLonType" subFieldSuffix="_coordinate"/> -->
<!--WAS  <fieldType name="location_rpt" class="solr.SpatialRecursivePrefixTreeFieldType" geo="true" maxDistErr="0.001" distErrPct="0.025" distanceUnits="kilometers"/> -->
```

Restart Solr, recreate the test index, observe.

Notes on the changes in this v2.1

Prior to Solr v7 we usually were required to add a spell check tokenizer within *managed-schema*. That worked and still does, but it adds its own dictionary (textSpell) to each document's results and uses considerable disk space.

The current configuration (v2.1) avoids that tokenizer addition and its stored dictionary (textSpell). Instead each document's "content" is used directly for spelling data.

Logging of results can be: no or yes (to a file named index.log). Default is yes.

Removal of deleted files in an index is automatic if their count in an index exceeds MAXDELETEDDOCS (default 100, set in crawler.php). Revise that limit to suit local conditions. Solr v7 has removed the Optimize button from the Solr admin interface; this change reimplements Optimize within crawler.php. Update: Solr v8.1 has removed even this capability: the team has a blind spot about cleaning.

Testing Solr start

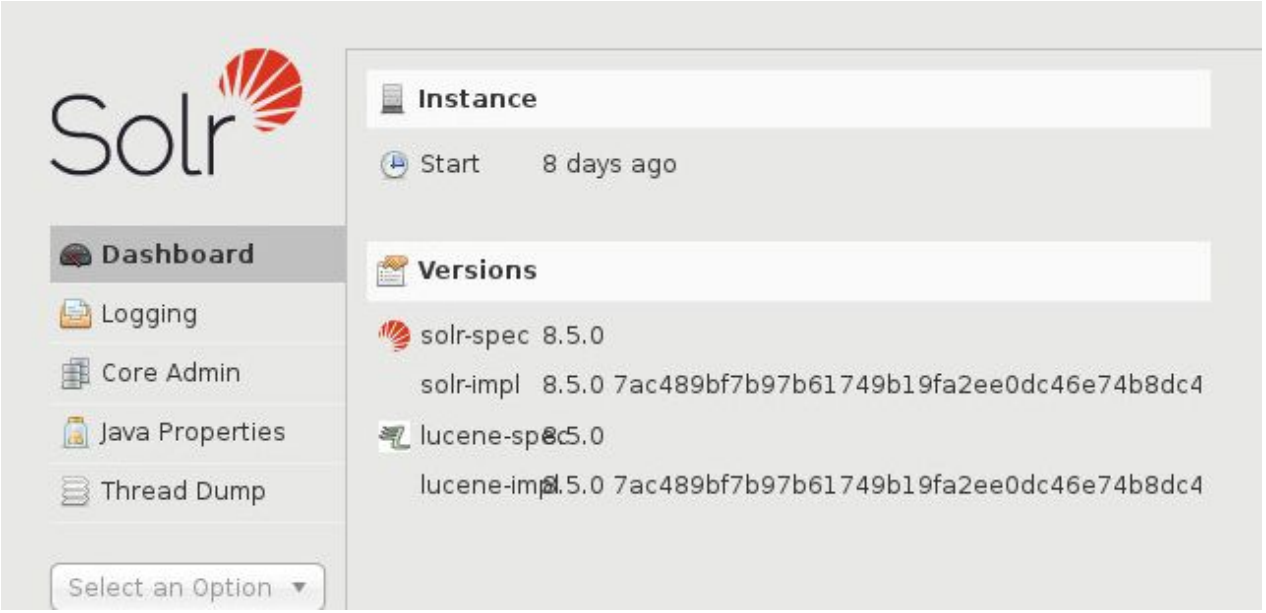
Start service solr

cd /etc/init.d and then say ./solr start

Observe a simple startup response

Double check by visiting Solr's admin web page, at

http://localhost:8983/solr



Failure to start Solr

Six common causes of failure to start:

1. Permissions not correctly set on Solr (/home/search/solr and /home/search/tmp), all of the contents thereof
2. Ownership of solr and tmp not assigned to the Solr user

   Suggest the name be solr in primary group users
3. Solr version has changed its schema requirements (yet again, sigh, adapting is discussed in previous slides)
4. Java JDK not pointed to correctly (if installing it as an extra from say Oracle, else we use system default settings)
5. Typos, of course
6. You are not root. Please find a grownup to help.

Shielding Solr via a web proxy server

We can use a web server (say Apache) to shield Solr from unauthorized contacts. An example Apache v2.4 configuration:

```
<Location "/solr">                                  # Solr URIs start with /solr
        <RequireAll>                                # For multiple who are you criteria
        Require IP 1.2.3.4/24 127.0.0.1             # First criteria: be in the IP list
        AuthType Basic                              # Second criteria: be in LDAP
        AuthBasicProvider ldap
        AuthLDAPUrl ldap://example.com/o=top?uid?sub?(objectClass=*)
        AuthLDAPBindAuthoritative on
        Require ldap-user solrproxy                 # Credentials will be required
        Require ldap-group cn=usergroup, o=users, o=top
        </RequireAll>                               # end of who are you checking
        ProxyPass http://solrhost.net:8983/solr                # proxy to Solr itself
        ProxyPassReverse http://solrhost.net:8983/solr  # rewrite responses
</Location>
```

Important: in Config and Query use this proxy's address plus credentials : edit *$solraddress* to be https:/proxybox.net/solr with credentials for LDAP username solrproxy: *$solrusername* and *$solrpassword* (Config & Query will use the proxy).

Sample Apache conf files for general access to Config and Query programs

```
Alias /config  /home/search/config.php
Alias /query  /home/search/query.php
```

Apache v2.2 version:

```
<Directory /home/search>
Order allow,deny
Allow from all                <naturally, be more restrictive>
</Directory>
```

Apache v2.4 version:

```
<Directory /home/search>
    Require ip 11.22.33.0/24
    Require all granted
</Directory>
```

Add your site's better access restrictions

Now for the fun part, web query & crawler

This is PHP v5.3 through v7.x material.

Thus Apache web server needs to support the full PHP set of modules (except perhaps the database items). See following slides about accommodating PHP within Apache.

Following are two screen captures showing PHP modules and Apache's mod_php in SUSE SLES11

Note that web PHP may show a white screen when a module is missing. View the Apache error log for the cause (typically /var/log/apache2/error_log). Add the missing PHP module.

The query and crawler's config programs are written in PHP and run via a web connection. Crawler (PHP) and Solr (Java) are stand alone programs.

Crawler, Config and Query

The ***crawler***, its configuration editor ***config***, and the general ***query*** program, can be located in multiple places anywhere which supports network contact with Solr.

Config and ***query*** are web based PHP programs. ***Crawler*** is a PHP command line program which needs access to its configuration file created by ***config*** and thus is typically colocated with ***config***.

Config writes to automatically created subdirectory *index*, and can invoke ***crawler*** for a test run. ***Crawler*** reads its configuration file from its command line. Such files are found typically in *index* and ***crawler*** writes to automatically created subdirectories *file* and *logs*.

Config and ***Crawler*** have presetting of variables at the top of their code files. Best to review these settings.

I put all in **/home/search** for convenience. Do as you wish. ***Config*** and ***crawler*** will need write access to their areas, and ***config*** is run by Apache (wwwrun:www for SUSE SLES).

Query program internal adjustables

$solrhost = "http://localhost:8983/solr"; How to reach Solr/Lucene

$solrusername = “myself”; **Use if Solr access is protected**

$solrpassword = “secret”;

$pagesize = 20; Number of responses per query (results per web page)

$indexlist = array(); List of which indices (empty means all), and their order, which this particular program copy will touch. Example:

$indexlist = array(“sales”, “marketing”, “pricing”);

This is an important localization and security feature.

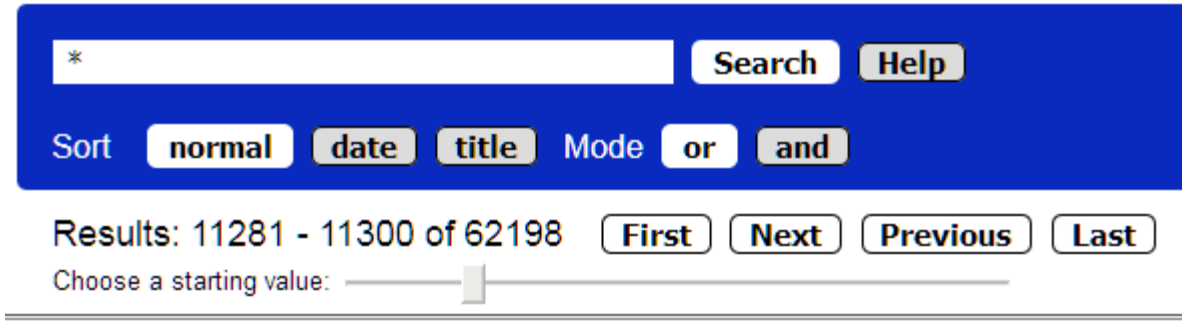
$trans[“index”] = array(“from”, “to”); Modify the URL prefix (“from”) of each query result to be URL prefix (“to”) for index “index”.

Indices are not changed, just the program’s returned results. Thus a query program may do post-indexing adjustment to the displayed URLs.

$spellcheck.count = 5; Number of “did you mean” results

A nuance about Query's slider control

Some browsers by default may not support the slider



One reason, in Firefox's *about:config* screen, is that

javascript.enabled=true

security.csp.enable=false

need to be set as above.

Query uses several lines of simple internal (no external) javascript to provide the slider.

Directories and left-over files therein

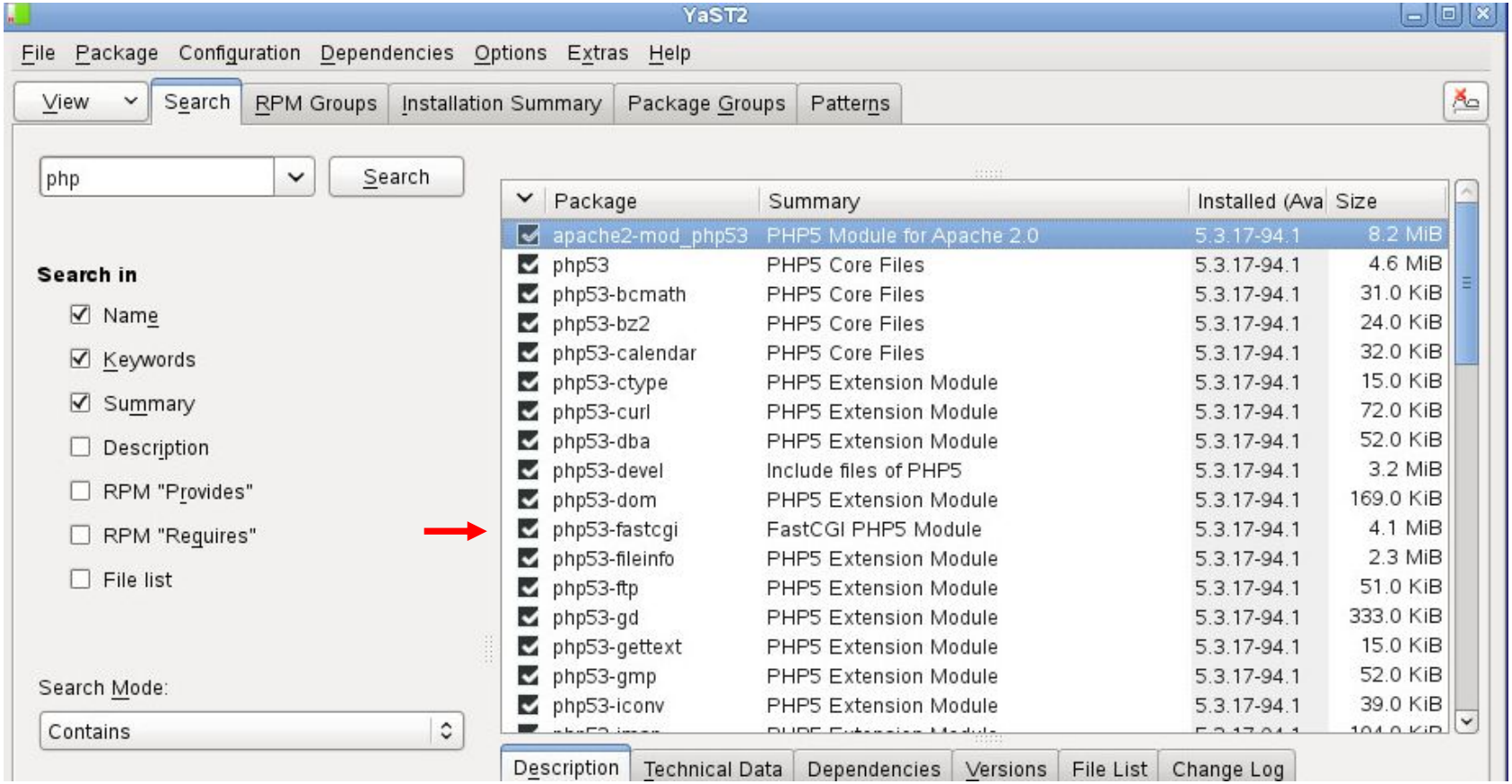
Program Config creates and writes crawler configuration files to subdirectory “**index**” which the crawler reads.

Program Crawler creates and writes to subdirectories “**logs**” and “**files**”. “Files” is a temporary work area.

Java uses subdirectory “**tmp**” for Jetty and Solr spill files. Best to clean it now and then, especically left-over files *upload-*.tmp* (a known parser bug, sigh).

Solr itself creates transaction logs within directory “**tlog**” in each index’s “data” directory. Best to clean these periodically. See /home/search/data/<index name>/data.

SUSE YaST, Software, PHP



Tick all the PHP boxes, except perhaps database items. PHP v5.3 is shown here. PHP v7.x also works.

Install php's fastcgi module if using threaded Apache.

YaST, Software, Apache mod_PHP

Package	Summary	Installed (Available)	Size
apache2	The Apache Web ...	2.2.31-TTP (2.2.12-64.1)	2.3 MiB
apache2-devel	Apache 2.0 Heade...	2.2.31-TTP (2.2.12-64.1)	628.0 KiB
apache2-doc	Additional Packag...	2.2.31-TTP (2.2.12-64.1)	10.3 MiB
apache2-event	Apache 2 event M...	2.2.31-TTP	2.5 MiB
apache2-example-pages	Example Pages fo...	2.2.31-TTP (2.2.12-64.1)	11.0 KiB
apache2-mod_jk	Connectors betwe...	1.2.40-0.2.1	406.0 KiB
apache2-mod_perl	Embedded Perl for...	2.0.4-40.24.1	7.0 MiB
apache2-mod_php53	PHP5 Module for ...	5.3.17-94.1	8.2 MiB
apache2-mod_python	A Python Module f...	3.3.1-147.24.104	1.5 MiB
apache2-prefork	Apache 2 "prefork...	2.2.31-TTP (2.2.12-64.1)	673.0 KiB
apache2-utils	Apache 2 utilities	2.2.31-TTP (2.2.12-64.1)	192.0 KiB
apache2-worker	Apache 2 worker ...	2.2.31-TTP (2.2.12-64.1)	689.0 KiB
libapr-util1	Apache Portable ...	1.3.4-12.22.23.1	208.0 KiB
libapr-util1-devel	Apache Portable ...	1.3.4-12.22.23.1	2.3 MiB
libapr1	Apache Portable ...	1.3.3-11.18.19.8	297.0 KiB
libapr1-devel	Apache Portable ...	1.3.3-11.18.19.8	4.5 MiB
libtcnative-1-0	JNI wrappers for A...	1.3.3-12.4.1	165.0 KiB

Later editions of the module, say for PHP v7, also work

Worker and Event are the threaded versions of Apache

PHP via Apache MPM *worker/event*

If using Apache MPM *worker* or *event*, not default *pre-fork,* then employ module **mod_fcgid** or standalone program **php_fpm**, and do not load mod_php.

The mod_fcgid approach (preferred) -

1. Obtain the module from https://httpd.apache.org/mod_fcgid

2 . File /etc/sysconfig/apache2, add fcgid (omit php, put ssl near start):

```
APACHE_MODULES="actions alias ssl auth_basic authn_core authn_file authz_host authz_groupfile authz_core authz_
user autoindex cgi dir env expires include rewrite header ldap authnz_ldap log_config mime negotiation setenvif
 socache_shmcb userdir reqtimeout proxy proxy_html proxy_ajp proxy_connect xml2enc status version fcgid http2"
```

3. /etc/apache2/conf.d/search.conf (an example config file):

```
Alias /config /home/search/config.php
<Directory /home/search>
        Options +ExecCGI
        Require ip 82.70.37.210/24 10.0.0.1/24 127.0.0.1
</Directory>
```

With similar Alias clauses then other resulting .php URLs will be recognized automatically and run through fcgid.

<Directory …> grants Apache access to a place in the file system.

mod_fcgid approach, cont'd

4. File /etc/apache2/conf.d/mod_fcgid.conf:

```
## PHP via FastCGI
##
## uncomment the following line if you want to handle php via mod_fcgid
##
<FilesMatch "\.php$">
   AddHandler fcgid-script .php
##JRD WAS   FCGIWrapper /srv/www/cgi-bin/php5 .php
   FCGIWrapper /srv/www/cgi-bin/php .php
   Options +ExecCGI
</FilesMatch>
##JRD addition, do not buffer output, allow long running sessions
   FcgidOutputBufferSize 0
   FcgidMaxRequestLen   40000000
   FcgidMaxRequestsPerProcess -1
   FcgidBusyTimeout 86400
</IfModule>
# End of <IfModule fcgid_module>
```

marks my changes

This configures a CGI wrapper for .php files with settings to permit smooth operation of the config.php program and others. Apache v2.2 omits the prefix "Fcgid" on parameter names (eg: BusyTimeout)

PHP with standalone service php-fpm

1. File /etc/sysconfig/apache2 -- add proxy and proxy_fcgi, no php5/7:

```
APACHE_MODULES="actions alias ssl auth_basic authn_core authn_file authz_host authz_groupfile
 authz_core authz_user autoindex cgi dir env expires include rewrite header ldap authnz_ldap
log_config mime negotiation setenvif socache_shmcb userdir reqtimeout proxy proxy_html proxy_
ajp proxy_connect xml2enc status version proxy_fcgi http2"
```

2. /etc/apache2/conf.d/foobar.conf – must use proxying with php_fpm
There are <u>many</u> variations on the proxy theme, see Apache docs for details

```
Alias /config /home/search/config.php
<Directory /home/search>
        Options +ExecCGI
        Require ip 82.70.37.210/24 10.0.0.1/24 127.0.0.1
</Directory>
<Location "/config">
        ProxyPass "fcgi://127.0.0.1:9000/home/search/config.php" flushpackets=on keepalive=on
        ProxyPassReverse "fcgi://127.0.0.1:9000/home/search/config.php"
</Location>
```

3. /etc/php<digit>/fpm -- copy *.default files to non-.default file names, modify file …/php-fpm.d/www.conf to specify UID/GID of web server:

```
; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the default user's group
;       will be used.
;;WAS user = nobody
;;WAS group = nobody
user = wwwrun
group = www
```

Enable systemd service php-fpm

Upgrading Solr over time

Unpack the new Solr parallel to the current Solr

Shutdown Solr: cd /etc/init.d then ./solr stop

Rename the current Solr to be say solr.old

Rename the new Solr to be solr (or use a symlink)

Edit …/solr/bin/solr to fix the lsof LISTEN item as shown earlier (this step is not needed for SLES 12 and later).

chown -R solr:users /home/search/solr

If the index-holding directory “data” were located within old Solr then move (mv) it to the new Solr.

Follow the previously shown steps about updating files *managed-schema* and *solrconfig.xml.*

The new Solr is ready to start and be tested:

cd /etc/init.d then ./solr start

MindWorks Inc. Ltd
210 Burnley Road
Weir
Bacup
OL13 8QE UK

Telephone: +44 (0) 170 687 1900
Fax: +44 (0) 170 687 8203
Web: www.mindworksuk.com
Email: training@mindworksuk.com