

【XenServer を使うインストール】

CRS では複数のサーバーコンピュータが必要なので、仮想化されたサーバー群へのインストールが、効率性や管理上からも都合がよい。そこで、システムは、サーバー仮想化 OS である XenServer のテンプレートとして配布する。

配布メディアは USB 外付け HDD ドライブ (Min 100GB) + XenServer の DVD 1 枚。

システムのインストールが済んだ状態で配布できるので、利用者は個別のパラメータ設定とデータ登録だけを行えばよい。以下ではその手順の概要を解説する。

□ テンプレートの種類

CRS_member

creator と repository のセット。シラバスの作成 (creator) と蓄積 (repository) を行うサーバーのテンプレート。学部単位に設置する。

CRS_searcher

searcher と Timetable のセット。検索 (searcher) と履修申請および個人用時間割りサービス (Timetable) を行うサーバーのテンプレート。学部単位、または全学に 1 台以上設置する。

□ creator、repository -- テンプレートからの作成

【ひな形】 CRS_member

IP: 192.168.0.32 となっている

DB: データが入っているので、一度消して sql から再構築する

Lucene の index ファイルも削除する (c:¥CRS¥index, c:¥CRS¥metaindex)

ただし、テストをするために残しておいて、最後にデータを再構築してもよい

開発: Java1.6、Eclipse3.7、と CRS ソースコード、HTML 等が設定済み

【導入手順】

1. スナップショット、テンプレートから VM を作成する
 2. 1 コントロールパネルで IP アドレスを変更する
 2. 2 コントロールパネルでコンピュータ名を変更する
 3. サービスで apache と tomcat を停止する
 4. apache の httpd.conf を編集する
 - ・ ServerName
 - ・ include 文
 5. Eclipse 上で、creator、repository の設定ファイルを編集し、Tomcat に送る
 - ・ creator: repository.xml
 - ・ repository: jef.conf(META INFORMATION<メタリポジトリの URL>、ドメイン情報)、
repositoryUser.xml

メタリポジトリの URL は 192.168.0.31 上となっている。

192.168.0.31 が host なので、リポジトリの種類は member となっている

ドメインにリポジトリを一つしかおかない場合は host とするのがよい

運用時には、公開メタリポジトリ (別途) か独自のメタリポジトリ URL を指定する必要がある
 6. 必要があれば metaRepository の設定ファイルの編集を行い、Tomcat に送る
(metaRepository は Tomcat にインストールしていない)
 7. DB 上で直接ユーザーを登録する
 - ・ テストデータに対応するユーザー (1422, 1359) があらかじめ登録されている
 8. サービスを起動する
 9. repository を起動して参照
http://(IP アドレス)/repository/control
- 配当所属項目の選択肢はダミーの所属が記載してあるので、設定しなす。これをしておかないと searcher での選択肢の表示がダミーの所属になる。
- すでに SML は有効化されているが、SML を更新したあともう一度有効化を実行する

10. **repository** で、[メタリポジトリ] (**metarepository** に登録) を行う

必要なければなくてもよい。

data/metaRepository.sm1 に **192.168.0.31** が記録されているので、登録画面にはこれが表示される。これを正しい情報に書き換えて、保存する (**metaRepository.sm1** を書き換える)。さらに登録ボタンで **MetaRepository** に登録する。

11. **creator** を起動して参照

http://192.168.0.32/creator/admin

テストデータが登録されているので正しく表示されることをチェックする

12. シラバスデータの登録

(1) 下記の DB テーブルと **lucene** のインデックスを削除する

creator: course (SQL: **delete from course;**)

repository: course (SQL: **delete from course;**)

c:¥CRS¥index (単にフォルダを削除する)

(2) **csv** データを **creator** に登録する

13. シラバスデータを **repository** へコミットする

creator ボタンで年度分をまとめてコミットできる

この時、**repository** では **host repository** に URL を自動送信する

URL は **host repository** の **bind** テーブルに登録される

□ searcher、Timetable -- テンプレートからの作成

【ひな形】 CRS_searcher

IP: 192.168.0.34 となっている

DB: user テーブルにはテスト用に学生データ (tanaka/tanaka) が 1 件だけ入っている

開発: Java1.6、Eclipse3.7、と CRS ソースコード、HTML 等が設定済み

【導入手順】

1. スナップショット、テンプレートから VM を作成する
2. コントロールパネルで IP アドレスを変更する
3. サービスで apache と tomcat を停止する
4. apache の httpd.conf を編集する
 - ServerName
 - include 文
5. Eclipse 上で、searcher、Timetable の設定ファイルを編集し、Tomcat に送る
6. DB 上で直接ユーザーを登録する

Timetable を利用する学生ユーザー (例: tanaka/tanaka)

管理者用ユーザーはスーパーパスワードだけでログインでき、それは searcher の jef.conf に記載する
7. サービスを起動する

【構成的なインストール】(参考)

J2SDK、postgresql、tomcat はすでにインストールされているものとする

1. GRS システムのコピー

C:¥Tomcat¥webapps にソフトウェアをコピーする
(ソフトウェア : webapps 配下のフォルダイメージ)

2. データベースの作成

Postgresql の pgAdmin を開いて以下の 3 つのデータベースを作成する (user は postgres、文字コードは UTF-8)。さらに、各データベースで SQL 画面を開いて SQL を実行しテーブルを作成する

DB 名	owner	encoding	テーブル作成用 SQL
creator	postgres	UTF-8	[¥c:¥tomcat¥webapps¥]creator¥system¥sql¥creator.sql
repository	postgres	UTF-8	[¥c:¥tomcat¥webapps¥]repository¥system¥sql¥repository.sql
searcher	postgres	UTF-8	[¥c:¥tomcat¥webapps¥]searcher¥system¥sql¥searcher.sql

3. 設定ファイルの編集

プログラム名	
creator	<p>[¥c:¥tomcat¥webapps¥]system¥config¥repository.xml]</p> <p>従属するリポジトリ情報を記入する。ID、パスワードは repositoryUser.xml と同じ</p> <p>creator と repository が同一コンピュータ内で稼働している時、url には IP アドレスではなく localhost と記述してもよい</p> <pre><repository> <repositoryUrl> http:// 192.168.0.131/repository/service </repositoryUrl> <repositoryUid>localSOC_1</repositoryUid> <repositoryPasswd>localSOC_1</repositoryPasswd> <userDomain>music.kwassui.ac.jp</userDomain> <userLabel>活水女子大学音楽学部</userLabel> </repository></pre>

プログラム名	
repository	<pre>[¥c:¥tomcat¥webapps¥]system¥config¥jef.conf] 例のようにドメイン情報とインデックスの格納場所を指定する 役割で、ドメインを代表するリポジトリは host、それ以外は member を指定する host は一つだけ。member の場合、host の URL が必要。認証キーは host 側で決め たものを記載する。スレッドとタイムアウトは管理者が記入 # ドメイン情報 ドメイン名 = music.kwassui.ac.jp ドメイン表示名 = 活水女子大学音楽学部 検索サービス = http://192.168.0.131/repository/service 役割 = member ホスト URL = http://192.168.0.213/repository/service 認証キー = abcdef12345 # # インデックスを保存する場所 INDEX_DIR = C:¥CRS¥index¥ # # 検索に許容する最大スレッド数とタイムアウト時間 # MAX_THREADS = 40; TIMEOUT_MILLIS = 3000; [¥c:¥tomcat¥webapps¥]system¥config¥repositoryUser.xml] アクセスを許可する Creator の情報を記入する。ID、パスワードを決定する <user> <repositoryUid>localSOC_1</repositoryUid> <repositoryPasswd>localSOC_1</repositoryPasswd> <requestUrl> http://192.168.0.131/creator/service </requestUrl> <userDomain>music.kwassui.ac.jp</userDomain> <userLabel>活水女子大学音楽学部</userLabel> <function>soc</function> </user></pre>

プログラム名	
searcher	<pre>[¥c:¥tomcat¥webapps¥]system¥config¥jef.conf] 例のように検索設定を記入する。通常は (1) のみ記入すればよい # 検索設定 # # (1)既定のリポジトリ URL DEFAULT_REPOSITORY=http://192.168.0.131/repository/service # # (2)テンプレートファイル名 TEMPLATE = entryTemplate.txt # # (3)フォーム番号 (フォームとして使用するリポジトリの bind.xml での順序番号) STANDARD_REPOSITORY = 0 # # (4)検索画面に使用する searchIndex の順序番号 SEARCHINDEX = 0 # # (5)ダウンロードするデータの項目名 (年度、学期、所属、時間割コード、科目名、 曜日、時限、担当者) CSV_ITEMLIST=A102-0 A103-0 A106-1 A203-0 A201-1 A501-0 A502-0 A601-1</pre>

4. ユーザー登録

creator と repository は利用できるユーザーを登録する必要がある。register ディレクトリにある property.txt ファイルがこれらのための設定ファイルである。DB 設定を読み取る conf ファイル名と登録用 csv ファイルの名前を書く。

creator には addusers でユーザーを多数登録するが、repository は管理者だけを adduser で登録する。

登録プログラム	機能	場 所
adduser.exe	1 人ずつ手作業での登録用	[インストールディレクトリ] ¥system¥register¥adduser.exe
addusers.exe	CSV ファイルから一括登録する	[インストールディレクトリ] ¥system¥register¥addusers.exe

CSV ファイルの構造

このようなデータをエクセルで作成し、user.csv という名前でカレントディレクトリに置いておき、addusers.exe を実行する

職員番号	教員名漢字	メールアドレス	ID	パスワード	役割
1240	井谷 俊二		itani	1240	教員
1251	小林 絵里子		kobayashi	1251	教員
1359	野中 和孝		nonaka	1359	教員
1402	加藤 豊		katou	1402	教員
1422	川場 隆	t.kawaba@gmail.com	kawaba	1422	管理者

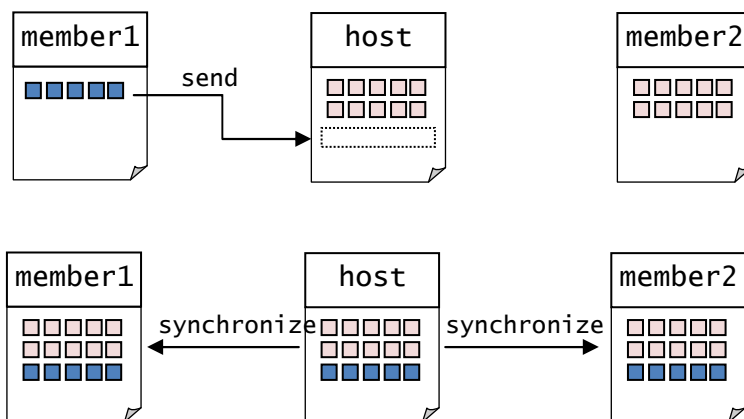
5. apache の設定ファイル (httpd.conf) の編集

- 1) 既定のファイルを `c:¥apache¥conf¥httpd.conf` に上書きコピーする
- 2) エディタでファイルを開きサーバーの IP アドレスだけを書き換えて上書き保存する
(Tomcat が `c:¥tomcat` にインストールされていることが前提)

6. リポジトリを作成する

最初に **host** リポジトリ (以下 **host**) を作成する。リポジトリが一つしかない時はそれを **host** とする。**host** とはドメイン内にある複数のリポジトリを代表するリポジトリである。**host** 以外は **member** リポジトリという。**member** は起動時に **host** に向けて **Bindlist** を送信する。**host** はそれを内部に記録している **Bindlist** に加える。また、統合した **Bindlist** をドメイン内のすべての **member** に再配布して **Bindlist** の同期を取る。

host 作成後、**member** を作成する。**host** の **SML** をエクスポートしてローカルに保存しておけば、それをインポートすることで **member** を素早く作成できる。**creator** や **searcher** は **repository** から必要なデータをもらうことにより起動するため、**repository** が稼働していないと起動できない。



`http://[IPアドレスまたはホスト名]/repository/control` に、アクセスする

- 1) **repository** で **SML** を新規作成する。
- 2) 作成した **SML** を「有効化」する
「有効化」によりその **SML** が従属するすべての **creator** に送信される。
- 3) **member** リポジトリ作成のため、**SML** をエクスポートしておく

7. Creator にシラバスデータを登録する

[http://\[IPアドレスまたはホスト名\]/creator/admin](http://[IPアドレスまたはホスト名]/creator/admin) に、管理者 ID でアクセスする。
role が管理者でないユーザーID ではアクセスできない。

1) CSV 形式にしたシラバスデータを登録する

- CSV は SML と同じ項目名を含み、一意の「識別コード」項目が必要
- 教員が編集するには、教員コードが指定されている必要がある
- 複数の教員が編集する科目は複数の教員コードを半角空白で区切って指定する（後からでも可）