Narval

Contents

1 Introduction **1**

2 Modèle conceptuel des traitements **1**
2.1 In 1
2.2 Out 2
2.3 Data 2

3 Implementation **2**
3.1 Environnement 3
3.2 Prés-requis 3
3.3 Configuration 4
3.4 Compilation 4
3.5 Installation 5
3.6 Installation des acteurs génériques 6

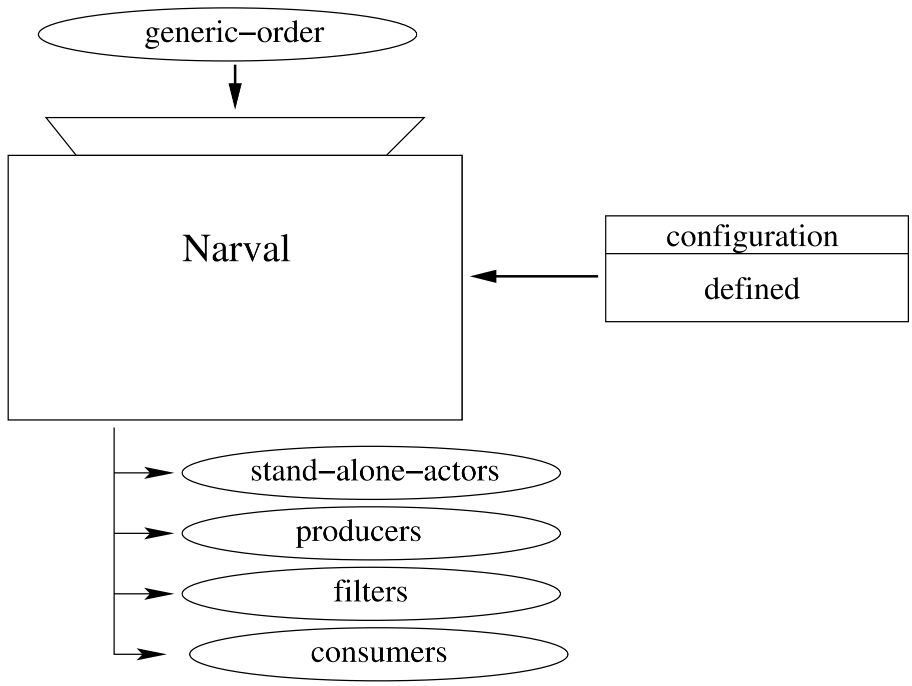
4 Test de Narval **6**
4.1 Serveur 6
4.2 Console Cliente 7
4.3 Client GUI 8

5 Détails d'un scénario **8**
5.1 Définir un sous-système 9
5.2 Configurer une topologie 9
5.3 Charger les bibliothèques 9
5.3.1 Manuellement 10
5.3.2 Automatisation 10
5.4 Lancement 11
5.5 Supprimer un sous-système 11

1 Introduction

Cette page décrit le traitement '**narval**' faisant partie de l'activité '**acquisition**'.
Le but de ce traitement est de recevoir les ordres à destination des différents acteurs.

2 Modèle conceptuel des traitements



2.1 In

Ce traitement est appelé par les évènements énumérés ci-dessous et intégrés à un automate intuitif car linéaire. Ces évènements pouront-être envoyés par :

- le traitement client issu de l'activité RUN-CONTROLLER
- l'interface graphique NARVAL
- la console NARVAL

L'automate ci-dessous ne donne pas explicitement les changements d'états car il sont implicites.

- Etat **Finished**
  – *launch SYSTEM_NAME HOST*
- Etat **Initial** :
  – *set action finish lc*
  – *set configuration_file SCENARIO.XML SYSTEM_NAME*
  – *set action configure SYSTEM_NAME*
- Etat **Configured** :
  – *set action unconfigure SYSTEM_NAME*
  – *set action load SYSTEM_NAME*
- Etat **Loaded** :
  – *set action unload SYSTEM_NAME*
  – *set action start SYSTEM_NAME*

- Etat **Stopped** :
  - *set action reset_com SYSTEM_NAME*
  - *set action start SYSTEM_NAME*
- Etat **Running** :
  - *set action stop SYSTEM_NAME*
  - *set action pause? SYSTEM_NAME*
- Etat **Paused** :
  - *set action stop SYSTEM_NAME*
  - *set action resume? SYSTEM_NAME*

2.2 Out

Le traitement répercute (tel un “broadcast”) les évènements entants à destination de l’ ensemble des acteurs.

2.3 Data

Des scénarios organisent la disposition et la configuration des acteurs ente eux et spécifie aussi le code métier à charger dans chacun des acteurs. Seul le nom du fichier XML est reçu du traitement, le fichier étant lui stocké par l’appliquation NARVAL..

- `in` : Objet **scénarios**
- `in/out` : Objet **data**

3 Implementation

Ce traitement est **déjà** implémenté dans la couche NARVAL.

Il s’agit d’un serveur adressable via un protocol de haut niveau : une chaîne encapsulé dans du XML et acheminée via un service WEB.

3.1 Environnement

De nouvelles sessions SSH sont ouvertes au lancement des sous-systèmes. **Il faut** placer les définitions des variables d’environnement dans le fichier *narval/.bashrc* et non pas dans *narval/.bash_profile* de même qu’**il est nécessaire** de les placer avant la ligne suivante :

```
# If not running interactively, don't do anything
[ -z "$PS1" ] && return
```

Fichier *narval/.bashrc* :

```
...
# pour configurer et compiler Narval
export PATH=~/tools/ada_i686_XG/bin:$PATH
export NARVALTOPDIR=~/svn/narval/Tags/narval-1.6.4
export NARVAL_TOOLS=~/tools

# pour executer Narval
`/home/narval/narval_install/rev-1.6.4/i686/bin/narval-config --environnement`
export AWS_PORT=6080
export GLADE_PORT=5080
```

```
alias serveur='~/narval_install/rev-1.6.4/i686/bin/narval_launch.sh'
alias console='~/narval_install/rev-1.6.4/i686/bin/narval_shell \
               --end_point http://lpnws5210:6080'
alias gui='java -jar /home/narval/tools/dist/NarvalConsole.jar'
...
# If not running interactively, don't do anything
[ -z "$PS1" ] && return
```

3.2 Prés-requis

```
# su - narval

$ mkdir ~/svn && cd !$
$ svn co http://csngwinfo.in2p3.fr:2401/narval/
$ export NARVALTOPDIR=/home/narval/svn/narval/Tags/narval-1.6.4

$ NARVAL_TOOLS=tools && mkdir ~/!$ && cd !$
$ wget http://csngwinfo.in2p3.fr:2401/NARVAL/downloads/gnat/ada_i686_XG.tar.bz2
$ wget http://csngwinfo.in2p3.fr:2401/NARVAL/downloads/tools_narval/xml_ez_out_1.05.tgz
$ wget http://csngwinfo.in2p3.fr:2401/NARVAL/downloads/tools_narval/log4Ada_snapshot_8_03_2007.tgz
$ wget http://narval.in2p3.fr/files/narval/wwdist.tgz

$ tar -jxf ada_i686_XG.tar.bz2
$ find . -name "*.tgz" -exec tar -zxf {} \;
$ ln -s log4Ada_snapshot_8_03_2007 log4ada

$ mkdir ~/.ssh && cd !$
$ ssh-keygen -t rsa
$ cp id_rsa.pub authorized_keys2
$ ssh localhost
> yes
^D
$ ssh lpnws5210
> yes
^D
```

Il faut préalablement initialiser la connexion SSH. C'est à dire qu'il faut se connecter manuellement à l'adresse utilisée pour définir le sous-système et répondre 'yes' afin que cette étape soit validée et ne bloque pas la connexion du client NARVAL.

Note :, le 04/05/09 LOG4ADA était buggé et il a fallu appliquer un patch.

3.3 Configuration

```
$ cd ~/svn/narval/Tags/narval-1.6.4/
$ ./configure
```

- `No ada compiler found.`

  ```
  $ export PATH=~/tools/ada_i686_XG/bin:$PATH
  ```

- `/src/base/narval.ads not found.`

  ```
  $ export NARVALTOPDIR=/home/narval/svn
  ```

- `error: /xml_ez_out/mckae not found.`

```
$ export NARVAL_TOOLS=~/tools
```

- `Miss ssl libraries for aws.`

  ```
  # apt-get install libssl-dev uuid-dev
  ```

- `Install aws tools.`
  Attention, les lignes suivante permet de contourner l'erreur :

  ```
  # apt-get install libaws-dev
  $ ./configure --with-aws=/usr/share/ada/adainclude/aws
  ```

  mais en fait aws est deja fournit avec le compilo installé manuellement.

  ```
  $ which gnatbind
  /usr/bin/gnatbind

  $ export PATH=/home/narval/tools/ada_i686_XG/bin:/usr/local/bin:/usr/bin:/bin
  $ which gnatbind
  /home/narval/tools/ada_i686_XG/bin/gnatbind

  $ ./configure
  ```

3.4 Compilation

```
$ cd ~/svn/narval/Tags/narval-1.6.4/
$ make
```

- `gnatdist: GARLIC library not found`
  Il faut retirer le / à la fin de la variable `$PATH`.

  ```
  $ /home/narval/tools/ada_i686_XG/bin//gnatdist
  gnatdist: GARLIC library not found
  gnatdist: *** can't continue

  $  /home/narval/tools/ada_i686_XG/bin/gnatdist
  Usage: gnatdist [options] name[.cfg] {[partition]} ...
  ...

  $ echo $PATH
  /usr/local/bin:/usr/bin:/bin:/usr/games:/home/narval/tools/ada_i686_XG/bin/

  $ export PATH=/usr/local/bin:/usr/bin:/bin:/usr/games:\
  /home/narval/tools/ada_i686_XG/bin
  $ ./configure
  ```

- `"Event_Receiver_Is_Alive" not declared in "Events_Receivers"`
  Mauvaise version de LOG4ADA.

- `"Full_Clear" is undefined`
  Patch prévu :

  ```
  $ cd ~/tools/xml_ez_out/mckae/
  $ patch -p2 < /home/narval/svn/narval/Tags/narval-1.6.4/patches/patch_xml_out_add_free
  ```

- `file "input_sources.ads" not found`

- `file "sax.ads" not found`
  Attention, la correction du fichier */svn/narval/Tags/narval-1.6.4/gnatdist_config/GNUmakefile* permet de contourner l'erreur :

  ```
  NARVAL_INCLUDES=-I../src/base \
  ...
  -I/home/narval/tools/ada_i686_XG/include/xmlada
  ```

  mais en fait il faut utiliser le compilateur installé manuellement.

  ```
  $ which gnatbind
  /usr/bin/gnatbind

  $ export PATH=...
  $ which gnatbind
  /home/narval/tools/ada_i686_XG/bin/gnatbind
  ```

- `locfatal error: file s-finimp.ali is incorrectly formatted`
  `make sure you are using consistent versions of gcc/gnatbind`
  Il faut utiliser le compilateur ADA installé et non pas le compilateur par défaut (voir l'erreur précédente).

  ```
  $ which gnatbind
  ```

3.5 Installation

```
$ make install
```

Fichier */.bashrc*

```
# pour executer Narval
'/home/narval/narval_install/rev-1.6.4/i686/bin/narval-config --environnement'
```

3.6 Installation des acteurs génériques

- Ajout des acteurs :

  ```
  $ cd ~/svn/narval/Tags/misc_actors/version-1.6.4/
  $ make library
  $ make add_project

  $ cat /home/narval/svn/narval/Tags/narval-1.6.4/autres_projets_cibles.mk
  ```

- Re-compilation prenant en compte les acteurs :

  ```
  $ cd ~/svn/narval/Tags/narval-1.6.4/
  $ make

  $ ls -l ~/svn/narval/Tags/narval-1.6.4/gnatdist_config/misc*
  ~/svn/narval/Tags/narval-1.6.4/gnatdist/misc_consumers.cfg -> misc_consumers.cfg
  ~/svn/narval/Tags/narval-1.6.4/gnatdist_config/misc_filters.cfg -> misc_filters.cfg
  ~/svn/narval/Tags/narval-1.6.4/gnatdist_config/misc_producers.cfg -> misc_producers.cfg

  $ make install
  ```

- Ajouter/modifier du code C et C++ dans les acteurs génériques :

  ```
  $ cd ~/svn/narval/Tags/misc_actors/version-1.6.4/
  $ make library
  ```

4 Test de Narval

Le serveur implémente d'office les 2 accés suivant :

- mode console
- et mode WSDL

Fichier */.bashrc* :

```
export AWS_PORT=6080
export GLADE_PORT=5080
```

Afin de pouvoir lancer la console ou l'interface graphique, il faut préalablement lancer le serveur.

4.1 Serveur

Lancement du serveur :

```
# apt-get install screen
$ ~/narval_install/rev-1.6.4/i686/bin/narval_launch.sh

$ ps -ef
narval    ?        SCREEN -m -d -c ~/narval_install/.../bin/screen_commands_for_narval
narval    pts/3    narval_naming_service --boot_server tcp://lpnws5210:5080
narval    pts/4    central_log --log_level info --boot_server tcp://lpnws5210:5080
narval    pts/5    aws_shell --aws_server_port 6080 --boot_server tcp://lpnws5210:5080
                   --automatic-script-loading

$ man screen
...
 C-a "        Present a list of all windows for selection.
 C-a n        Switch to the next window.
 C-a p        Switch to the previous window (opposite of C-a n).
 C-a d        Detach screen from this terminal.

$ screen -r
```

4.2 Console Cliente

Le lancement d'un client en mode console peut au choix, ouvrir un shell ou juste envoyer les commandes une à une :

```
$ cd ~/narval_install/rev-1.6.4/i686/bin
$ ./narval_shell --end_point http://lpnws5210:6080 help commands
GET,HELP,LAUNCH,SEND_ORDER,SET,LOAD,SLEEP,WAIT,ECHO,DUMP,GROUP,SPECIAL_DOMI
$

$ ./narval_shell --end_point http://lpnws5210:6080

narval shell>help commands
GET,HELP,LAUNCH,SEND_ORDER,SET,LOAD,SLEEP,WAIT,ECHO,DUMP,GROUP,SPECIAL_DOMI
```

Cette commande permet de voir si le dilogue à bien lieu entre la console et le serveur :

```
narval shell>launch lc lpnws5210
ok

narval shell>get sub_systems
lc
```

- `waited 10 seconds without success`
  50% des problèmes viennent de l'environnement dixit un sage de l'IPN ! (qui ?)
  Les clients narval utilisent des connexion SSH non interractives.
  Essayez `ssh localhost 'echo $PATH'` afin de diagnostiquer le problème. Veuillez vérifier :
  - Les accès SSH (clé + 'yes')
  - La définition de l'environnement (au début du fichier */.bashrc*)
- `waited 10 seconds without success`
  L'instuction suivante permet de verifier si c'est un problème X11. De plus les eventuels messages d'erreur sortirons dans la console aws_shell.

  ```
  launch lc localhost local
  ```

Le script *test_shell* correspond a un scenario complet pour lancer une acquisition :

- Scenario complet :

  ```
  $ ~/narval_install/rev-1.6.4/i686/bin/narval_shell --end_point http://lpnws5210:6080
  > load test_shell localhost
  ```

- Pas à pas : (ie `cat ~/narval_install/rev-1.6.4/narval_scripts/test_shell`)

  ```
  echo on
  # launch a chef_orchestre with sub_system name toto
  launch toto localhost
  set run_number 2 toto
  get sub_systems
  get arguments toto
  set configuration_file multi_data_rate_fifo.xml toto
  set action configure toto
  set action load toto
  get actors toto
  get arguments toto data_receiver
  get arguments toto a_stand_alone_example
  get actors toto
  set action initialise toto
  set action start toto
  sleep 10.0
  set action stop toto
  get run_number toto
  set action reset_com toto
  set action unload toto
  set action unconfigure toto
  set action finish toto
  ```

4.3 Client GUI

```
# apt-get install sun-java6-jre
$ java -jar ~/tools/dist/NarvalConsole.jar
```

Utiliser l'url *http://lpnws5210:6080*.

- `Erreur in URL factory [http://lpnws5210:6080] [null]`
  Ajouter le fichier */werewolf.xml* :

```
<wolf>
    <splashScreen>
        <audio jingle="false">
        </audio>
    </splashScreen>
    <servers>
        <url><![CDATA[ http://lpnws5210:6080]]>
        </url>
        <url><![CDATA[ http://lpnws5210:5080]]>
        </url>
        <timeout><![CDATA[10000]]>
        </timeout>
    </servers>
    <narval>
        <scanning><![CDATA[1500]]>
        </scanning>
    </narval>
</wolf>
```

Lancez le wizard de NarvalConsole qui propose puis sélectionnez le fichier de config *multi_data_rate_fifo.xml* pendant la procedure de lancement du sous systeme.

5 Détails d'un scénario

Pour appeler la console :

```
$ ~/narval_install/rev-1.6.4/i686/bin/narval_shell --end_point http://lpnws5210:6080
```

Voici le scénario décliné au long de cette partie :

```
launch lc localhost
 set configuration_file test_full_generic.xml lc
 set action configure lc
  set action load lc
   set action initialise lc
    set action start lc
    sleep 2
    set action stop lc
   set action reset_com lc
  set action unload lc
 set action unconfigure lc
set action finish lc
```

5.1 Définir un sous-système

Définir, énumérer et supprimer les sous-sytèmes :

```
launch lc localhost
get sub_systems
...
console set action finish lc
```

5.2 Configurer une topologie

Juste apres le launch il faut specifier un fichier de topologie pour le sous systeme. C'est dans ce fichier que l'on trouve la description d'ou se trouve les acteurs.

```
set configuration_file test_full_generic.xml lc
set action configure lc
...
set action unconfigure lc
```

Pour info :

```
generic_producer : produit des données en utilisant une bibliothèque partagée

data_rate_producer : produit des buffers de données le plus vite possible,
utilisé par data_rate_consumer pour mesurer la vitesse des différents liens
supportés par Narval

standalone_actor : acteur qui ne produit ni ne consume aucune données,
suit tout de même les changements d'états
```

5.3 Charger les bibliothèques

Cette étape est nécessaire afion d'utiliser le wizard de NarvalConsole qui propose de selectionner le fichier de config *test_full_generic.xml* pendant la procedure de lancement du sous systeme.

```
> Peux-tu me répéter comment créer le script associé au fichier
> 'test_full_generic.xml' afin de charger un premier sous-système.

le fichier doit s'appeler
test_full_generic
et se trouver dans le repertoire pointe par $NARVAL_SCRIPTS

dedans tu dois faire des commandes du type :
set preload_library bibliotheque_1.so $1 nom_acteur
set preload_library bibliotheque_2.so $1 nom_acteur
.
.
.
set preload_library bibliotheque_n.so $1 nom_acteur
set library /home/narval/producteur/libexample_c.so $1 nom_acteur

Pour chaque acteur qui a besoin d'une bibliotheque partagee
/home/narval/producteur/libexample_c.so
ou
/home/narval/filtre/libexample_c.so
ou
/home/narval/consommateur/libexample_c.so
elle meme dependant de bibliotheque_1.so, ... bibliotheque_n.so

Generalement pour savoir ce dont a besoin la bibliotheque a charger il
faut faire (dans narval_shell)
set library /home/narval/producteur/libexample_c.so nom_sous_system
nom_acteur
si ca passe sans erreur tu peux mettre la ligne dans le script en
remplaçant nom_sous_system par $1
sinon le message d'erreur te donne les symboles manquant qui te
permettent de trouver la bibliotheque a charger via un set preload_library
```

5.3.1 Manuellement

En console (s'assurer que le fichier *$NARVAL_SCRIPTS/test_full_generic* est vide) :

```
set action load debug
set preload_library ~/svn/.../misc_actors/.../producers/libc_test_lib.so lc data_transmitter
set library ~/svn/.../misc_actors/.../producers/libexample_c.so lc data_transmitter
set library ~/svn/.../misc_actors/.../filters/libexample_c.so lc filter
set library ~/svn/.../misc_actors/.../consumers/libexample_c.so lc data_receiver
...
set action unload lc
```

Utiliser `nm` pour trouver les librairies à pré-charger en fonction des symboles manquants donnés par les log.

`set library` va appeller les fonctions `process_config` et `process_register`.

5.3.2 Automatisation

Une fois les bibliothèques trouvées et l'ordre défini, on renseigne le fichier *$NARVAL_SCRIPTS/test_full_generic* en remplaçant le nom du sous-système par `$1`.

```
set preload_library ~/svn/.../misc_actors/.../producers/libc_test_lib.so $1 data_transmitter
set library ~/svn/.../misc_actors/.../producers/libexample_c.so $1 data_transmitter
set library ~/svn/.../misc_actors/.../filters/libexample_c.so $1 filter
set library ~/svn/.../misc_actors/.../consumers/libexample_c.so $1 data_receiver
```

En console, `set action load lc` appelera ce script, déduisant son nom du la précédente directive `set configuration_file` (lui retirant l'extension `.xml`).

```
set action load lc
get actors lc
...
set action unload lc
```

5.4 Lancement

```
set action initialise lc
set action start lc
sleep 2
set action stop lc
...
set action reset_com lc
```

5.5 Supprimer un sous-système

Résumé des commande de destruction :

```
set action reset_com lc
set action unload lc
set action unconfigure lc
set action finish lc
get sub_systems
```