October 30, 2007

AFFAIR- a flexible fabric and application information recorder

T. Anticic

Rudjer Boskovic Institute, Zagreb, Croatia

I. INTRODUCTION

The AFFAIR(1) package monitors the DATE software as well as the system behavior of the computer cluster where DATE is running. It is installed separately from DATE and can be downloaded from the AFFAIR Web site http://www.cern.ch/affair, with the full code and detailed installation instructions. This manual pertains to AFFAIR versions above 3.0, which, among other changes, has MySQL support.

Figure 1 gives an overview of the AFFAIR system. It extracts performance data from the *monitored nodes* and forwards it in fixed time intervals to the *monitoring station*, where the data is stored and analysed. Summary plots accesible via a Web server or the Affair Control are also created on the *monitoring station*. All communications and data transfer use the DIM/SMI client/server software of DATE.

The main AFFAIR processes, running on the *monitored nodes* are:

- *DATE Collector*: it collects the performance data from processes running on the LDCs and GDCs. An instance of this process is started for each LDC and GDC node. The DATE Collector maps to the DATE shared memory, accesses the data (see table XI) and transfers the extracted values to the *Affair Monitor*.
- *System Collector*: it collects the system performance from nodes where DATE processes are running. The *System Collector* parses the files in the /proc directory or runs system calls, and transfers the extracted values to the *AFFAIR Monitor*.

The main AFFAIR processes, running on the *monitoring station* are:

- *Affair Monitor*: it collects monitored data from the Collectors and places them into temporary storage databases as well as into permanent ROOT storage.
- *Data Processor*: it analyses the stored data and produces in fixed time intervals the updated plots for the Web interface.
- *Affair Control*: a graphical user interface which is able to configure, start, stop, control AFFAIR on the entire cluster, and to display performance graphs.
- *Web inteface*:provides links to various performance plots and data.

The performance data is grouped into three types, named *monitoring sets*, depending on which *Collector* is used:

- GDC *monitoring set*: data collected by the *DATE Collectors* on GDC machines
- LDC *monitoring set*: data collected by the *DATE Collectors* on LDC machines
- DATEMON *monitoring set*: data collected by the *System Collectors* on all machines

The DIM package is used for the communication between the nodes:

- the DATE DIM name server controls the communication between the AFFAIR Monitor and nodes for DATE processes (DATE Collector).
- the AFFAIR DIM name server controls the communication between the AFFAIR Monitor and the nodes where the System Collector is running. It can be different from the DATE DIM name server, so that system performance monitoring can be continued, even when all DATE processes are shut down.

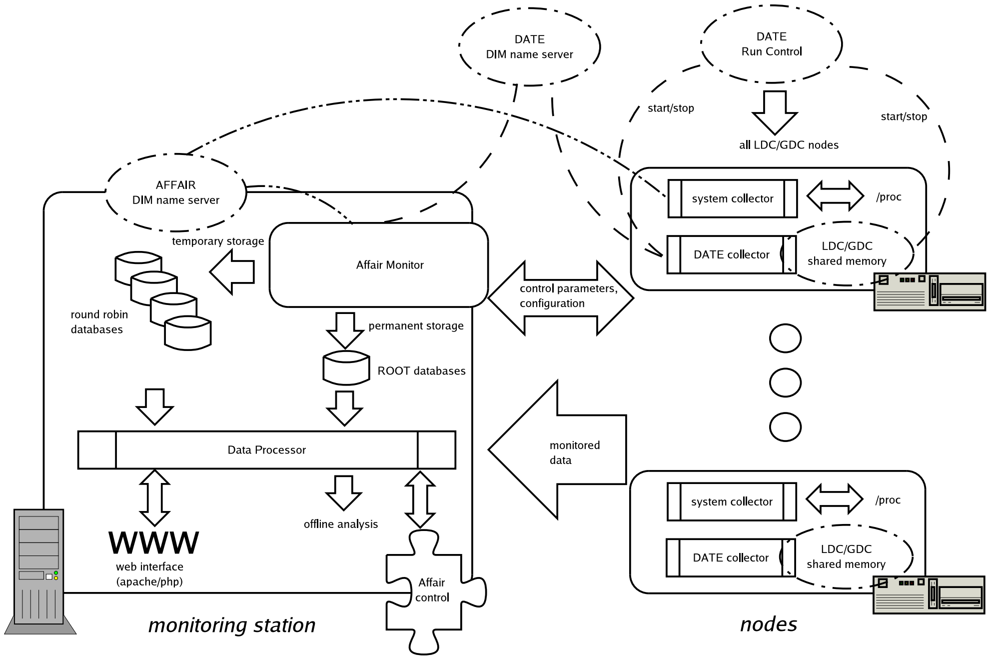

FIG. 1 Overview of the Affair structure

II. INSTALLATION OF THE MONITORING STATION USING RPMS

The *monitoring station* is the host where data collection, storage and analysis of performance data takes place and where the Web interface is located.

A set of RPMs has been created for the monitoring station on systems running Scientific Linux 4 (SLC4), where a user alicedaq, with its home directory at /home/alicedaq, is assumed to exist. The default SLC4 configuration includes the necessary MySQL database and the apache web server. The AFFAIR MySQL account should be created as described in section III.A.

The RPMs are:

- php-5.2.2-3.i386.rpm
- php-cli-5.2.2-3.i386.rpm
- php-devel-5.2.2-3.i386.rpm
- php-common-5.2.2-3.i386.rpm
- php-pdo-5.2.2-3.i386.rpm
- php-mysql-5.2.2-3.i386.rpm
- root-5.17-04.i386.rpm
- affair-monitor-3.2-1.i386.rpm

They are all located at http://www.cern.ch/affair/Download.html. The installation procedure is (as root):

- rpm -ivh php-5.2.2-3.i386.rpm php-cli-5.2.2-3.i386.rpm php-devel-5.2.2-3.i386.rpm php-common-5.2.2-3.i386.rpm php-pdo-5.2.2-3.i386.rpm php-mysql-5.2.2-3.i386.rpm root-5.17-04.i386.rpm affair-monitor-3.2-1.i386.rpm

This installs the ROOT package, adds php support for the web server. configures the web server, installs the actual AFFAIR monitor, and creates the approprite link at the web server to the AFFAIR web interface.

The environment variables are initialised by adding the following line to the */home/alicedaq/.bashrc* file:

- source /home/alicedaq/affair/set-affairenv.sh

The apache daemon is started by:

- /usr/sbin/apachectl start

It is recommended to start apache at boot time, by adding the following line to the file */etc/rc.d/rc.local*:

- [-x /usr/sbin/apachectl] && /usr/sbin/apachectl start

The AFFAIR Web is accessed with:

- http://mymachine.mydomain/affair/index.php (e.g.: http://pcald30.cern.ch/affair/index.php)

III. INSTALLATION OF THE MONITORING STATION WITHOUT RPMS

The *monitoring station* is the host where data collection, storage and analysis of performance data takes place and where the Web interface is located. The following packages must be installed on the *monitoring station*:

- ROOT (version 5.*) (2), with MySQL support.
- DIM and SMI.
- MySQL(4) (version 5.*)
- the Apache Web server (version 2.*), with PHP (version 5.*) support and source code (3),(5).

The steps involved in the installation are the following:

1. creation of the AFFAIR MySQL account
2. creation of the Apache Web server with PHP support
3. declaration of environment variables
4. installation of the actual AFFAIR code
5. linking of the Web server to the AFFAIR web interface

The AFFAIR user should also start the DIM name server on both the DIM_DNS_NODE computer (the standard DATE DIM name server) and the AFFAIR_DNS_NODE computer (the AFFAIR specific DIM name server). Note that for the simplicity these two name servers can be the same.

A. MySQL configuration

AFFAIR uses the MySQL database to store and manipulate the configuration files, and to extract the list of nodes monitored from the DATE MySQL database. While flat files can still be employed, it is strongly recommended to use MySQL. The procedure to create the affair MySQL database and account, with your choice of username and password, is:

- mysql -u root -p
- CREATE DATABASE IF NOT EXISTS AFFAIR;
- grant all privileges on *.* to 'my_username'@'%' identified by 'my_password' with grant option;
- grant all privileges on *.* to 'my_username'@'localhost' identified by 'my_password' with grant option;
- quit;

B. Apache Web server with PHP support installation

The system configuration on the *monitoring station* may already have a combination of apache/PHP running. To enable AFFAIR installation without interferring with existing web server setups, several web server installation scenarios are considered:

- no Apache Web server running.
- Apache Web server running with PHP and no PHP source code.
- Apache Web server running with PHP and PHP source code.

In all cases the path/name of the PHP configuration file has to be declared via the environment variable PHPCONFIG, which is required by the AFFAIR Web interface. In systems where PHP code comes with a preinstalled Apache Web server, this file is usually */usr/local/bin/php-config*. Consider these cases in more detail:

1. *No apache web server*

If the *monitoring station* has no web server, the procedure outlined in the affair homepage at http://www.cern.ch/affair/WebInstallation.html should be followed to install Apache with PHP and MySQL support. This page also contains instruction on how to edit the apache configuration file *httpd.conf*.

2. *Apache with PHP, but no PHP source code*

Often the PHP source code is not included in a distribution with a functioning web server with PHP support. The AFFAIR site has the relevant code which can be downloaded into any directory (e.g. */local/phptmp*), and installed (e.g. in */local/php*) by:

- cd /local/phptmp
- tar xvzf php-5.1.6.tar.gz
- cd php-5.1.6
- ./configure - -prefix=/local/php - -enable-track-vars - -with-apxs2=/usr/local/apache2/bin/apxs - -with-mysql
- make
- make install (*as root*)

The PHPCONFIG environment variable for this example is */local/php/bin/php-config*.

3. *Apache with PHP, and PHP source code*

Nothing needs to be done.

4. *configuration the apache httpd.conf file*

It is necessary to make several changes to the apache *httpd.conf* configuration file (normally in the */usr/local/apache2/conf* directory) to enable the Web interface to use php, MySQL, and password protected pages for offline analysis:

- The following lines, if not present, should be added:
  - LoadModule php5_module modules/libphp5.so
  - AddType application/x-httpd-php .php
- The line containing the text *DirectoryIndex* should be changed to the following:

  - DirectoryIndex index.html index.html.var index.php

- To enable password protected offline analysis of the web interface, the line containing the text *AllowOverride* within the <Directory "/usr/local/apache2/htdocs"> section needs to be changed from *AllowOverride None* to:

  - AllowOverride All

- The following lines need to be added to enable the affair MySQL username and password to be available to the Web interface (the same affair MySQL username and password as set in III.A have to be used):

  - <VirtualHost *:80>
  - php_value mysql.default_host localhost
  - php_value mysql.default_user my_username
  - php_value mysql.default_password my_password
  - </VirtualHost>

5. *starting apache*

The apache daemon is started by:

- /usr/local/apache/bin/apachectl start

It is recommended to start apache at boot time, by adding the following line to the file */etc/rc.d/rc.local*:

- [-x /usr/local/apache/bin/apachectl] && /usr/local/apache/bin/apachectl start

C. environment configuration

The directory where the AFFAIR code on the *monitoring station* will be downloaded and installed needs to be declared as AFFAIR_MONITOR. In addition, all the relevant paths and environment variables for DIM and SMI (both the standard DATE and AFFAIR specific name server), ROOT, and PHP need to be declared. The best method is to add them to the */.bashrc* file. A typical example is:

- # DATE environment variables
- source /date/setup.sh
- # affair specific environment variables
- export AFFAIR_MONITOR=/local/affair
- export AFFAIR_DNS_NODE=lxs5013.cern.ch
- # necessary paths needed
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DIMBIN:$SMIBIN :$AFFAIR_MONITOR/rrdtool/lib: $AFFAIR_MONITOR/src/root:$AFFAIR_MONITOR/src/dim;
- #directory where php configuration is contained
- export PHPCONFIG=/local/php/bin/php-config
- # affair specific environment variables for MySQL. The same affair MySQL username and password as set in III.A have to be used
- export AFFAIR_MYSQL_PWD=my_password
- export AFFAIR_MYSQL_USER=my_usernamex
- export AFFAIR_MYSQL_HOST=localhost

AFFAIR_DNS_NODE	pcdaq01

TABLE I The file *$DATE_SITE_CONFIG/AFFAIR.config*

As ROOT has to be accesible also from the Apache Web server its environment variables have to be in a common location such as */etc/profile*, and not in a user specific one. Thus */etc/profile* should contain:

- #root specific environment variables
- export ROOTSYS=/local/root-for-affair
- export PATH=$PATH:$ROOTSYS/bin
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ROOTSYS/lib

If for any reason this is not desirable to do on your system, then the above lines should be added to the top of the $AFFAIR_MONITOR/web/doOneComputer.sh and $AFFAIR_MONITOR/web/offline/doOffline.sh files.

D. AFFAIR instalation

The affairMonitorVERSION.tar.gz file (e.g. affairMonitor3.20.tar.gz) needs to be downloaded to the $AFFAIR_MONITOR directory from the AFFAIR site *http://www.cern.ch/affair*. At this point AFFAIR can be installed as follows:

- cd $AFFAIR_MONITOR
- tar xvzf affairMonitorVERSION.tar.gz
- make

This compiles the full AFFAIR code for the *monitoring station*. At this stage the Web interface can be enabled. The Web server should be linked (needed to be root) to the appropritate AFFAIR code:

- ln -s $AFFAIR_MONITOR/web /usr/local/apache/htdocs/affair (if /usr/local/apache is the directory where Apache is installed)

The AFFAIR Web is accessed with:

- http://mymachine.mydomain/affair/index.php (e.g.: http://pcald30.cern.ch/affair/index.php)

IV. INSTALLATION OF THE NODES

The affairDateVERSION.tar.gz file available on the AFFAIR Web page needs to be downloaded on every *node* where DATE is running, and placed in the $AFFAIR directory (a usual DATE setup already defines AFFAIR as /date/affairRC). If the $AFFAIR directory is on a shared file system, this needs to be done only once.

The AFFAIR_DNS_NODE environment variable has to be declared in addition to the standard DATE environment variables.

The file *$DATE_SITE_CONFIG/AFFAIR.config* should be changed or created to have the content shown in table I, adjusted to the actual setup:

After this, the code can be compiled:

- cd $AFFAIR
- tar xvzf affairDate3.10.tar.gz
- make

(in this example the AFFAIR version is 3.10).

The AFFAIR *Collectors* running on the LDCs and GDCs are started by the run control via the script */date/runControl/AFDC.sh*

By default, this script starts only a dummy process, so it must be changed to actually start the AFFAIR Collectors. The script shown below is suggested, but it is up to the user to write his/her own.

- #!/bin/csh -f
- if ("${?DATE_ROLE}" == "0") setenv DATE_ROLE "UNKNOWN"
- setenv L "/tmp/AFFAIR.${DATE_ROLE}.log"
- chmod 0777 "$L" "$L.*"
- mv -f "$L.4" "$L.5"
- mv -f "$L.3" "$L.4"
- mv -f "$L.2" "$L.3"
- mv -f "$L.1" "$L.2"
- mv -f "$L" "$L.1"
- if (${DATE_ROLE} == "EDM" || ${DATE_ROLE} == "DDG") then
- ${DATE_RC_BIN}/dummy_collector $1
- else
- ($AFFAIR/startAffair.sh $1) >& "$L"
- endif
- chmod 0777 "$L" "$L.*"

V. INSTALLATION OF THE NODES USING RPMS

The compilation of the AFFAIR *Collectors* on the nodes, as described in the previous section, may also be done by downloading from http://www.cern.ch/affair/Download.html the *affair-collector-3.2-1.i386.rpm* RPM, and installing it on the nodes:

- rpm -ivh affair-collector-3.2-1.i386.rpm

VI. CONFIGURATION AND OPERATION

AFFAIR processes need to be started on all the *monitored nodes* as well as the *monitoring station*. The *monitored nodes* are enabled by the DATE *runControl*, where the setting of the checkbutton marked AFFAIR starts the monitoring on all the LDC and GDC nodes.

The method to configure and operate AFFAIR is to use on the *monitoring station* the *affairControl* script:

- cd $AFFAIR_MONITOR
- ./affairControl.sh

This script starts a GUI which allows easy setup and configuration of the *monitored nodes*, as well as viewing of all performance plots accesible via the Web interface.

A. Initial AFFAIR MySQL database setup

The first step is to create the affair MySQL tables, and fill them with default values. This is achieved by pressing the <u>Create and fill default tables</u> button in the *initial setup* tab (see Figure 2).

B. Controlling list of monitored nodes

Figure 3 shows the tab *set list of computers* used to modify the list of monitored *nodes*. Clicking the button Import GDC/LDC nodes from DATE Roles table loads in one step all the nodes that should be monitored in a DATE environment. As the full node name is necessary,with the domain (for example, it should be pcald31.cern.ch and not pcald31), and the names in the DATE_ROLES table are without the domain name, the appropriate domain should be entered in the text box next to this button.

C. Controlling monitoring and plot creation processes

The tab *Monitor Control* (see Figure 4) is the GUI that controls the *Affair Monitor* and *Data Processor* processes, and provides their status. A detailed description of this tab follows.

The right half contains four columns, with a Refresh current status button at the bottom:

- The first column gives a list of the possible monitoring sets that can be selected. The selection is done by clicking on one or more entries. The three remaining columns refer, row by row, to the monitoring set in this column.
- The second column gives the actual status of how many of the *nodes* are successfully monitored (i.e. have performance data collected) for a particular monitoring set. If the entry is pink with "*not monitored*" text, then the process is down. If it is yellow, with a text such as "*10/50 nodes monitored*" then the process is up, but only a fraction of the full number of the selected *nodes* is actually receiving performance data. If it is green, then all the *nodes* are receiving data.
- The third column informs if currently data is also being stored in ROOT files for offline processing.
- The last column informs if currently plots are generated for the web interface (i.e. if the Data Processor process is up).

The (re)start monitoring for selected monitoring sets button on top of the left half of the tab controls the monitoring of the selected monitoring set(s). It starts the processes (if a proccess is already active, it is killed first) selected by the two sets of radio button choices below it:

- The default behavior of the *AFFAIR Monitor* is to create performance plots accessed via the Web interface. It is possible to disable the plot creation by clicking on the appropriate radio button. This can be useful if only *AFFAIR Control* is used to inspect the performance.
- In addition to temporary storage, for a chosen monitoring set *AFFAIR Monitor* can write data also to permanent ROOT offline storage. If there is no need for reanalysis of old data, then the *monitor, with no offline storage* option should be chosen (this is the default option).

Pressing the Start default DATE monitoring (offline,with plots) button (re)starts the processes needed for a typical DATE setup. LDC,GDC, and DATEMON monitoring is started for all the selected *nodes*, with offline storage.

Pressing the Stop monitoring for selected monitoring sets stops the monitoring and plot creation processes for the selected monitoring sets).

VII. CONFIGURATION TABLES

There are several MySQL configuration tables regulatating what is being monitored and what is being plotted. They are all accessed via the *Affair Control* GUI

The most important ones are described in this section.

The default values the AFFAIR_GDC, AFFAIR_LDC, AFFAIR_DATEMON, and AFFAIR_MONSET_DEFAULTPARS tables are filled with (see Section VI.A) do not need to be edited for normal operation, and it is usually only the AFFAIR_COMPUTERLIST table that needs to be configured (see Section VII.A).

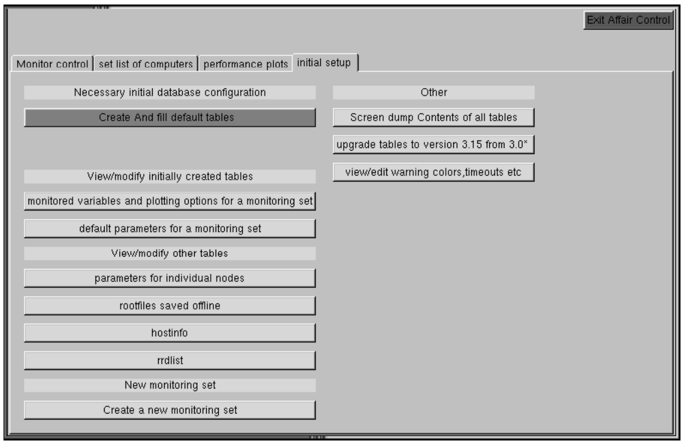

FIG. 2 Affair Control tab from where the initial setup and configuration is performed

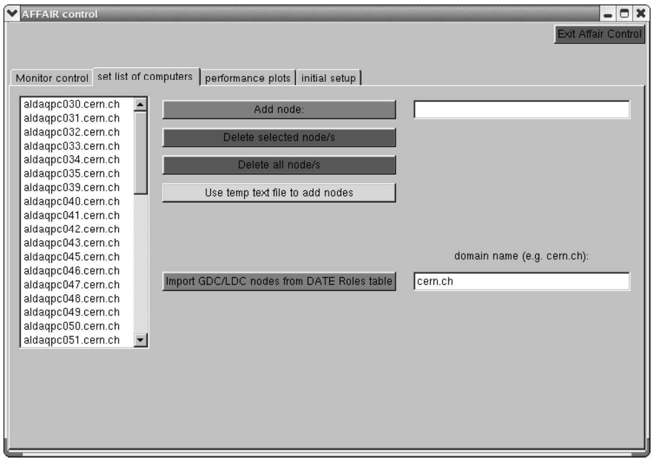

FIG. 3 Affair Control tab from where the list of computers monitored is controlled

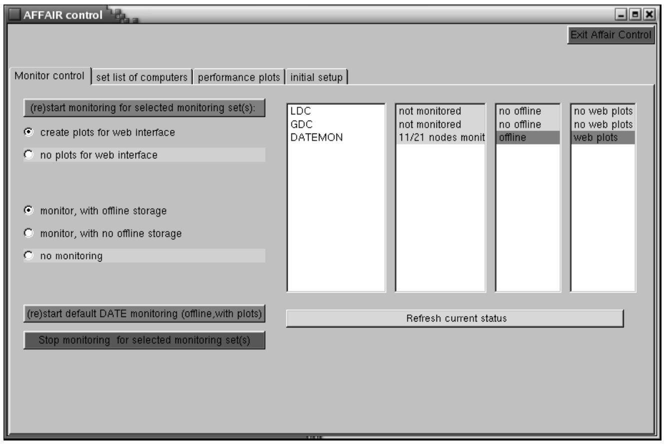

FIG. 4 Affair Control tab where the monitoring processes can be started or stopped, and their status viewed

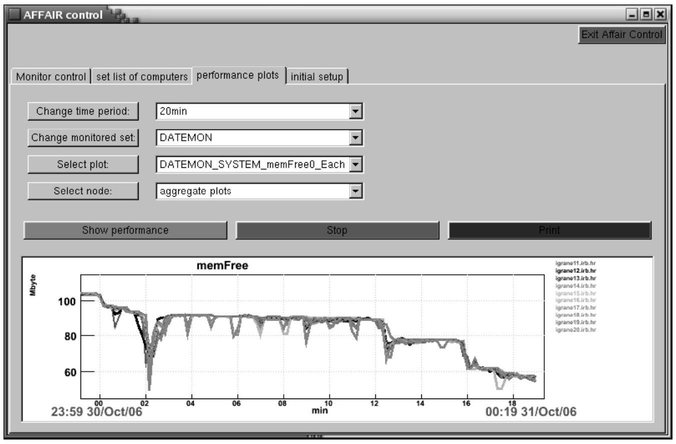

FIG. 5 Affair Control tab giving access to all the performance plots

MySQL table	comment
AFFAIR_COMPUTERLIST	defines which nodes should be monitored
AFFAIR_GDC	configures GDC plots
AFFAIR_LDC	configures LDC plots
AFFAIR_DATEMON	configures DATEMON plots
AFFAIR_MONSET_DEFAULTPARS	provides default parameters for the Collectors on the *monitored nodes*
AFFAIR_COMPUTERPARS	table listing the final parameters for all *monitored nodes* and *monitoring sets*
AFFAIR_ROOTFILELIST	list of all ROOT files,including their start and end times. Used for the offline analysis
AFFAIR_PARS	Various setup parameters, such as number of nodes per plot

TABLE II All the configuration files

A. AFFAIR_COMPUTERLIST table

This table contains the full list of *nodes* to be monitored in a DATE environment. It is accessed with the *set list of computers* tab (see Figure 3). Any new *node* that will be monitored in a DATE environment needs to be included in this table. Note that if a node is removed by the DATE *runControl*, there is no need to remove it from this table. It is important to write the full node name, including the domain, such as pcdaq01.cern.ch (i.e. with cern.ch), as this

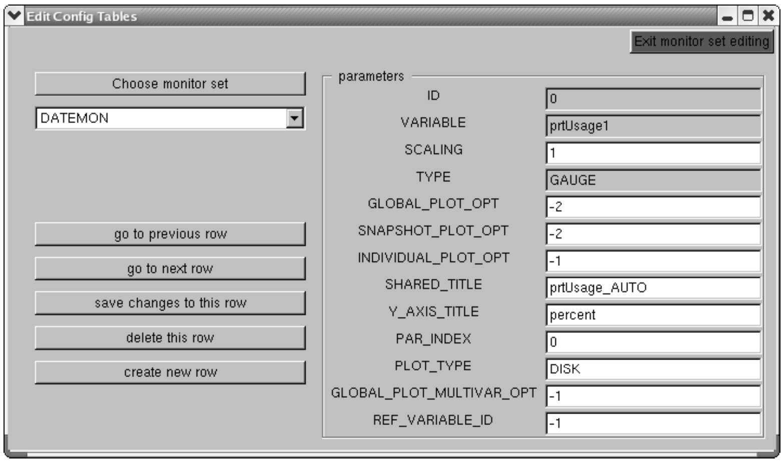

FIG. 6 Window opened by clicking the view/edit plotting options in the *initial setup* tab, showing the plotting parameters for the different monitoring sets.

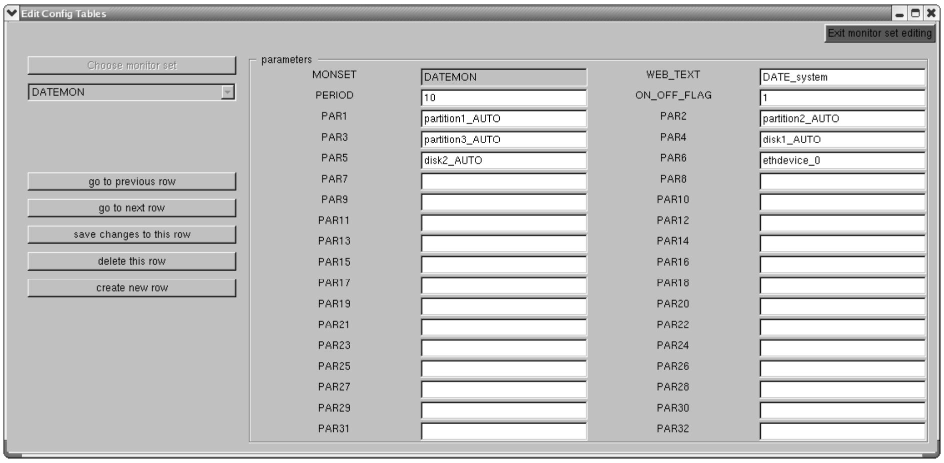

FIG. 7 Window opened by clicking the view/edit monitoring set defaults in the *initial setup* tab, showing the default parameters for the different monitoring sets.

ensures consistency in DIM service declarations between the *Collectors* and the *AFFAIR Monitor*.

The GUI provides several ways to control the list of *monitored nodes*:

- *nodes* can be added one by one by entering the name in the text box next to the Add node button, and then clicking this button
- selected *nodes* in the column can be removed by clicking on the Delete selected nodes button
- The delete all nodes button removes all nodes
- Clicking the button Import GDC/LDC nodes from DATE Roles table loads in one step all the nodes that should be monitored in a DATE environment. As the full node name is necessary, and the names in the DATE_ROLES table are without the domain name, the appropriate domain should be entered in the text box next to this button.

B. AFFAIR_GDC, AFFAIR_LDC, and AFFAIR_DATEMON tables

These tables define which plots are created for the Web server, and how they look (units, scaling, labels,etc). They are accessed with the view/edit plotting options button in the *intial setup* tab (see Figure 6). Every monitored variable has a row dedicated to it. The order of the rows is the same as the order of the monitored data sent from the nodes. Each of the rows has the parameters presented in table III. The format of the table and the default values (see Section VI.A) are shown in Tables IV, V, and VI.

C. AFFAIR_MONSET_DEFAULTPARS and AFFAIR_COMPUTERPARS tables

The AFFAIR_MONSET_DEFAULTPARS table provides the default parameters sent by the *AFFAIR Monitor* to the *Collectors*, with a row for each *monitoring set*. It is accessed with the view/edit default parameters for all nodes button in the *intial setup* tab. (see Figure 7).

If a parameter is set to AUTO, then the *Collector* will determine its value. For example if the paramater for the ethernet card is set to AUTO, then the *Collector* will use the card with the largest traffic. If a parameter has an underscore, then the text to the right of the underscore is treated as described above, while the text to the left is used to describe the parameter in the GUI for the AFFAIR_COMPUTERPARS table (see Figure 8).

The AFFAIR_MONSET_DEFAULTPARS table is also used to control which *monitoring set* is displayed on the Web and the text that accompanies it. The format of the table is shown in Table VII, while the default values are shown in Table VIII.

The final parameters returned from the nodes and used in subsequent AFFAIR running are stored in the AFFAIR_COMPUTERPARS table, with a separate entry for each *monitored node* and *monitoring set*. The format of

Variable paramater	Description
ID	The index of the variable. In the first line it is set to 0 (and thus is the first variable monitored), and sequentially goes up to the last variable.
Name	The name of the variable. Every variable must have a unique name
scaling constant	Used to scale the data. For example, 0.00097656 will scale measured data from KB to MB
Type	The data source type used for creating the Round Robin Databases. The value GAUGE means that the value itself is stored, and not the differences from previous values
Global plot option	Indicates how the global plots should be created: • -1: the global plots are created only in the superimposed way. An example is Figure 14 • -2: the global plots are not created at all • 0 or higher: creates a graph of the sum of this variable over all computers. To get different graphs displayed on different plots, set this parameter for each variable to its index (ID). To group several graphs on the same plot, set this parameter for all the variables of the group to the lowest variable index (ID) in the group. An example of this is Figure 12
Snapshot plot option	Indicates how snapshot plots of the current time should be created: • -1: bar plot with only this monitored variable • -2: not created • 0 or higher: creates a graph of the sum of this monitored variable and the consecutive variables having this ID. It is the ID of the first variable which should be summed. Examples are Figure 15 and Figure 16
Individual plot option	Indicates how the plots for an individual computer should be created: • -1: graph with only this monitored variable • -2: not created • 0 or higher: creates a graph of the sum of this monitored variable and the consecutive variables having this ID. It is the ID of the first variable which should be summed. An example is Figure 14
Shared title	Title used by plots when individual, snapshot or global plot option >= 0. It consists of two parts, separated by an underscore. The first part is the title of the set of variables, while the second is the label of each variable. For example: CPU_User, CPU_Nice. If the second part is *AUTO*, then the appropriate label from *computers.conf* is used
Y axis title	Label shown on the y axis
index in computers.conf	Used to display the actual parameter (e.g. 0 for ethernet card 0, /hda2,...) on the plot. The value is the actual index of the variable in *computers.conf*, starting from the first parameter (i.e. the 5th position, which is defined as index 0, see table IX)
plot type	Groups the plots in the Web interface. This way only a few related plots (such as CPU, IO, ...) are shown at one time
Global plot option - multiple variables	Makes possible to have superimposed graphs of *different* monitored variables. This is used to display on one plot the DDL throughput for *all* the used DDLs (defined by the last parameter in this configuration file) for up to 20 different *monitored nodes*. It is the ID of the first variable which should be superimposed.
ID of reference variable	Plots the graphs only if the value of the monitored variable referred is >= 0. (For example, this is used to display only active DDLs (active if idDdl -monitored variables 8 to 31 (see Table XI)- is >= 0), out of a maximum of 32. So if only one DDL is active, only one graph instead of 32 will be shown)
Low level warning	In the web tabular output format paints cell blue if value is below this number
High level warning	In the web tabular output format paints cell red if value is above this number

TABLE III Parameters for the AFFAIR_GDC, AFFAIR_LDC, and AFFAIR_DATEMON tables

0	runNumber	1.0	GAUGE	-2	-1	-2	XXX	count	-1	RUN	-1	-1
1	MBytesRec	0.00097656	GAUGE	-2	-1	-1	MBytesIn	MBytes	-1	IO	-1	-1
2	MBytesInj	0.00097656	GAUGE	-2	-1	-2	MBytesRec	MBytes	-1	IO	-1	-1
3	runStatus	1.0	GAUGE	-2	-1	-2	XXX	status	-1	RUN	-1	-1
4	eventsRec	1.0	GAUGE	-1	-1	-1	XXX	count	-1	EVENT	-1	-1
5	eventCount	1.0	GAUGE	-2	-1	-2	XXX	count	-1	EVENT	-1	-1
6	MBytesRecRate	0.00097656	GAUGE	6	6	-1	bandwidth_Rec	MBytes/sec	-1	IO	-1	-1
7	MBytesInjRate	0.00097656	GAUGE	-2	6	-2	bandwidth_Inj	MBytes/sec	-1	IO	-1	-1

TABLE IV Default parameters for the AFFAIR_GDC table

0	runNumber	1.0	GAUGE	-2	-2	-2	XXX	count	-1	RUN	-1	-1
1	MBytesRec	0.00097656	GAUGE	-2	-2	-1	MBytesIn	MBytes	-1	IO	-1	-1
2	MBytesInj	0.00097656	GAUGE	-2	-2	-1	MBytesRec	MBytes	-1	IO	-1	-1
3	runStatus	1.0	GAUGE	-2	-2	-2	XXX	status	-1	RUN	-1	-1
4	eventsRec	1.0	GAUGE	-1	-2	-1	XXX	count	-1	EVENT	-1	-1
5	eventCount	1.0	GAUGE	-2	-2	-2	XXX	count	-1	EVENT	-1	-1
6	MBytesRecRate	0.00097656	GAUGE	6	6	-1	bandwidth_Rec	MBytes/sec	-1	IO	-1	-1
7	MBytesInjRate	0.00097656	GAUGE	-2	6	-1	bandwidth_Inj	MBytes/sec	-1	IO	-1	-1
8	idDdl1	1.0	GAUGE	-2	-2	-2	XXX	count	-1	DDL	-1	-1
9	idDdl2	1.0	GAUGE	-2	-2	-2	XXX	count	-1	DDL	-1	-1
10	idDdl3	1.0	GAUGE	-2	-2	-2	XXX	count	-1	DDL	-1	-1
⋮	⋮	…	…	…	…	…	…	…	…	…	…	…
31	idDdl24	1.0	GAUGE	-2	-2	-2	XXX	count	-1	DDL	-1	-1
32	byteDdl1	1.0	GAUGE	-2	32	32	DDLbandwidth_AUTO	MBytes/sec	1	DDL	32	8
33	byteDdl2	1.0	GAUGE	-2	32	32	DDLbandwidth_AUTO	MBytes/sec	2	DDL	32	9
34	byteDdl3	1.0	GAUGE	-2	32	32	DDLbandwidth_AUTO	MBytes/sec	3	DDL	32	10
⋮	⋮	…	…	…	…	…	…	…	⋮	…	…	⋮
55	byteDdl24	1.0	GAUGE	-2	32	32	DDLbandwidth_AUTO	MBytes/sec	24	DDL	32	31
56	fragDdl1	1.0	GAUGE	-2	56	56	DDLfragment_AUTO	count/sec	1	DDL	56	8
57	fragDdl2	1.0	GAUGE	-2	56	56	DDLfragment_AUTO	count/sec	2	DDL	56	9
58	fragDdl3	1.0	GAUGE	-2	56	56	DDLfragment_AUTO	count/sec	3	DDL	56	10
⋮	⋮	…	…	…	…	…	…	…	⋮	…	…	⋮
79	fragDdl24	1.0	GAUGE	-2	56	56	DDLfragment_AUTO	count/sec	24	DDL	56	31

TABLE V Default parameters for the AFFAIR_LDC table

0	prtUsage1	1.0	GAUGE	-2	-2	-2	prtUsage_AUTO	percent	0	DISK	-1	-1
1	prtUsage2	1.0	GAUGE	-2	-2	-2	prtUsage_AUTO	percent	1	DISK	-1	-1
2	prtUsage3	1.0	GAUGE	-2	-2	-2	prtUsage_AUTO	percent	2	DISK	-1	-1
3	prtFree1	1.0	GAUGE	-2	-2	-2	prtFree_AUTO	Mb	0	DISK	-1	-1
4	prtFree2	1.0	GAUGE	-2	-2	-2	prtFree_AUTO	Mb	1	DISK	-1	-1
5	prtFree3	1.0	GAUGE	-2	-2	-2	prtFree_AUTO	Mb	2	DISK	-1	-1
6	DiskRead1	1.0	GAUGE	-2	-2	-2	diskRead_AUTO	blocks/sec	3	DISKRW	-1	-1
7	DiskRead2	1.0	GAUGE	-2	-2	-2	diskRead_AUTO	blocks/sec	4	DISKRW	-1	-1
8	DiskWrite1	1.0	GAUGE	-2	-2	-2	diskWrite_AUTO	blocks/sec	3	DISKRW	-1	-1
9	DiskWrite2	1.0	GAUGE	-2	-2	-2	diskWrite_AUTO	blocks/sec	4	DISKRW	-1	-1
10	CPUUser	1.0	GAUGE	-1	10	10	CPU_User	percent	-1	CPU	-1	-1
11	CPUNice	1.0	GAUGE	-1	10	10	CPU_Nice	percent	-1	CPU	-1	-1
12	CPUSys	1.0	GAUGE	-1	10	10	CPU_System	percent	-1	CPU	-1	-1
13	CPUIdle	1.0	GAUGE	-1	10	10	CPU_Idle	percent	-1	CPU	-1	-1
14	MBytesIn	0.00097656	GAUGE	14	14	-1	bandwidth_In	MByte/sec	-1	IO	-1	-1
15	MBytesOut	0.00097656	GAUGE	14	14	-1	bandwidth_Out	MByte/sec	-1	IO	-1	-1
16	swpFree	1.0	GAUGE	-1	-2	-1	swp	Mbyte	-1	SYSTEM	-1	-1
17	memFree	1.0	GAUGE	-1	-2	-1	mem	Mbyte	-1	SYSTEM	-1	-1
18	logSess	1.0	GAUGE	-2	-2	-2	log	count	-1	SYSTEM	-1	-1
19	pswpin	1.0	GAUGE	-2	-2	-2	pin	Mbyte/sec	-1	SYSTEM	-1	-1
20	pswpout	1.0	GAUGE	-2	-2	-2	pout	Mbyte/sec	-1	SYSTEM	-1	-1

TABLE VI Default parameters for the AFFAIR_DATEMON table

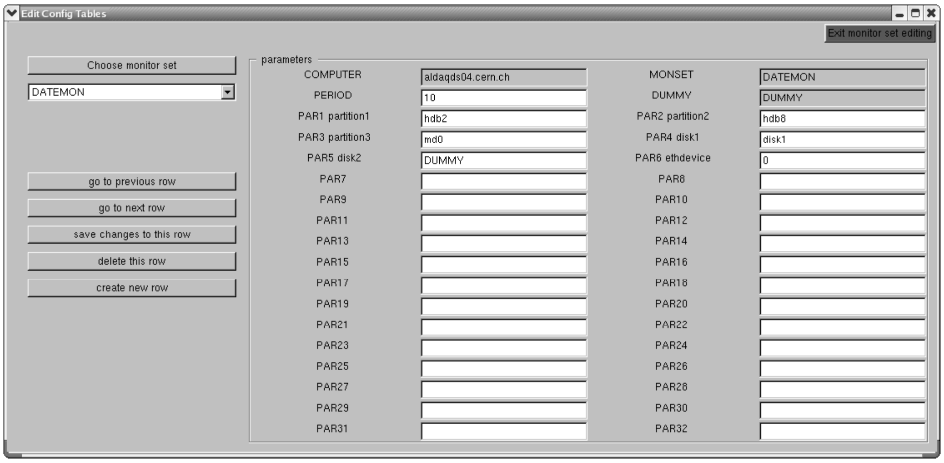

FIG. 8 Window opened by clicking the view/edit parameters for individual nodes in the *initial setup* tab, showing the parameters for individual nodes.

monitoring set	text displayed on web (underscore becomes blank space)	monitoring time period	0/1 flag to inform if monitoring set used or not	sequence of up to 32 parameters sent to monitored nodes

TABLE VII format of the monsetDefault.conf configuration file. In the case of DATEMON, the eight parameters sent correspond to time period,unused, partition name 1 and 2 and 3, disk name 1 and 2, and the ethernet device (see table XII). In the case of GDC, there are four parameters, the first is the time period, while the other three are unused (but must be written anyway, for compabitibility). In the case of LDC, the 28 parameters sent correspond to time period,three unused, and the names of the 24 DDLs.

the AFFAIR_COMPUTERPARS table is shown in Table IX, while Table X shows an example of generated entries at runtime.

The parameters in the AFFAIR_COMPUTERPARS table can be changed (for example, to switch to a less used ethernet card, say from 0 to 1, or eth0 to eth1) and these will be used next time either the *AFFAIR Monitor* or the appropriate *Collector* starts. The parameters are changed using the GUI, by clicking the view/edit parameters for individual nodes button in the *intial setup* tab (see Figure 8).

VIII. DETAILED DESCRIPTION

In this section a more detailed description of AFFAIR components is presented.

A. Monitored values

The performance data is separately analysed and plotted for each *monitoring set*. Tables XI and XII show the monitored values for the different *monitoring sets*. The definition of the partitions 1 to 3, of the disks 1 and 2, and of the network card is done via configuration file (see section VII.C).

DATEMON	DATE_system	10	1	partition1_AUTO	partition2_AUTO	partition3_AUTO	disk1_AUTO	disk2_AUTO	ethdevice_0
GDC	GDC	10	1	DUMMY	DUMMY				
LDC	LDC	10	1	DUMMY	DUMMY	LDCi_AUTO *(repeated 24 times,i runs from 1 to 24)*			

TABLE VIII Default values in the AFFAIR_MONSET_DEFAULTPARS table

computer name	monitor set	time interval	dummy	variable specific parameters

TABLE IX format of the AFFAIR_COMPUTERPARS table

pcdaq01.cern.ch	DATEMON	10	DUMMY	hda2	hda3	hda5	disk1	disk2	0
pcdaq02.cern.ch	DATEMON	10	DUMMY	hda2	hda3	hda5	disk1	disk2	0
pcdaq03.cern.ch	DATEMON	10	DUMMY	hda2	hda3	hda5	disk1	disk2	0
pcdaq04.cern.ch	DATEMON	10	DUMMY	hda2	hda3	hda5	disk1	disk2	0
pcdaq05.cern.ch	DATEMON	10	DUMMY	hda2	hda3	hda5	disk1	disk2	0
pcdaq06.cern.ch	DATEMON	10	DUMMY	hda2	hda3	hda5	disk1	disk2	0
pcdaq01.cern.ch	GDC	10	DUMMY	DUMMY	DUMMY				
pcdaq02.cern.ch	GDC	10	DUMMY	DUMMY	DUMMY				
pcdaq03.cern.ch	LDC	10	DUMMY	DUMMY	DUMMY	215.0	0.0 *(repeated 23 times)*		
pcdaq04.cern.ch	LDC	10	DUMMY	DUMMY	DUMMY	406.0	0.0 *(repeated 23 times)*		
pcdaq05.cern.ch	LDC	10	DUMMY	DUMMY	DUMMY	105.0	0.0 *(repeated 23 times)*		
pcdaq06.cern.ch	LDC	10	DUMMY	DUMMY	DUMMY	204.0	205.0	0.0 *(repeated 22 times)*	

TABLE X An example of entries of a filled (at runtime) AFFAIR_COMPUTERPARS table for a setup with four LDC nodes and two GDC nodes. Three of the LDCs have one DDL, while the remaining one has two DDLs. The format of the DDL names is *RORC serial number.channel number*

B. Round Robin Databases

The Round Robin Databases(6) (RRD) is a very efficient temporary storage mechanism that enables the incoming data to be stored with minimal delay. RRDs work with a fixed amount of data, corresponding to a fixed time deph. The data is structured in rows, with one row for a particular time. The column width is the number of variables monitored, with each a real number. For every node, each *monitoring set* has its own RRD, that contains the information shown in table XIII. The data is always averaged for the given time resolution, which enables only a small amount of data to be read out, making plot creation efficient. The maxima for each row can also be recorded by the RRDs. This feature is used in plotting the maxima as dashed lines, superimposed over the mean values. Of course, for the highest resolution (10 sec in our case) the maxima and mean are the same numbers. As the data is averaged for a particular time resolution, the RRDs can not be used to create histograms. These, however, can be made offline using the permanent data storge. The time resolution can be changed on an existing RRD file by using the *rrdtool - tune* function (for details see the description of the *rrdtool* function on the following Web page: *http://people.ee.ethz.ch/õetiker/webtools/rrdtool/doc/rrdtool.en.html*).

C. Web interface

The function of the AFFAIR Web interface is to provide links to plots as well as to performance data. The *eps* format of the plots created by the *Data Processor* (see section VIII.G) is not suitable for Web access, so the Web interface first converts the requested plots to a *gif* format, using a system call to the *$AFFAIR_MONITOR/support/convert* function.

ID	Description
0	run number
1	total MB recorded
2	total MB injectred
3	run status
4	number of events recorded
5	event count
6	data recorded rate in MB/sec
7	data injected rate in MB/sec
8 to 31	DDL id (only for LDC.conf)
32 to 55	DDL data rate (only for LDC.conf)
56 to 79	DDL fragment number rate (only for LDC.conf)

TABLE XI LDC and GDC monitored values

ID	Description	ID	Description
0	percent of partition 1 filled	11	Nice CPU as percentage of total CPU
1	percent of partition 2 filled	12	System CPU as percentage of total CPU
2	percent of partition 3 filled	13	Idle CPU as percentage of total CPU
3	amount of disk free of partition 1 in MBytes	14	bandwidth into network card in MB/sec
4	amount of disk free of partition 2 in MBytes	15	bandwidth out of network card in MB/sec
5	amount of disk free of partition 3 in MBytes	16	amount of swap free in MBytes
6	blocks/sec read from disk 1	17	amount of RAM free in MBytes
7	blocks/sec read from disk 2	18	number of sessions logged in
8	blocks/sec written to disk 1	19	pages swapped in in MBytes/sec
9	blocks/sec written to disk 2	20	pages swapped out in MBytes/sec
10	User CPU as percentage of total CPU		

TABLE XII System monitoring (monitored values for the DATEMON monitoring set). The definition of partitions 1,2, 3, the disks 1,2, and the network card is done via the *Affair Control* GUI (see section VII.C)

time resolution	total time depth
10 sec	1 hour
60 sec	6 hour
4 min	24 hour
28 min	7 days
2 hours	1 month
6 hours	3 months
1 day	1 year

TABLE XIII Relation between the time resolution and the total time depth

A screenshot of the AFFAIR Web interface top page is shown in Figure 9. The links on the left provide access to summary plots (see section VIII.D) as well as current performance data in a tabular text format. The links on the right generate detailed plots for each *monitored node*.

There are no histograms in the Web interface as the information from them can not be extracted from the temporary databases (so called round robin databases (RRDs). See section VIII.B).

On each page there is a choice of the available plot time intervals, starting from the previous 20 minutes and going up to the last month. The refresh time of the page can be chosen from the provided time intervals (between 30 seconds and infinity), see Figure 9. There is also a choice of the type of plots to be viewed (such as IO or DISK), as defined in Section VII.B. This is useful to only view a subset of the plots at one time.

The current data can be accessed in a tabular format by clicking on the GDC or LDC or DATEMON *values* link on the left. An example is given in Figure 10. Each *node* has a separate row, while the columns are the variables monitored. Sorting (ascending or descending) may be done on any variable,or on the name of the *node*. The average and summed values for each variable are given at the bottom of the table. If a*node* has not been updated for more than one minute, it is declared dead, and placed in a separate pink colored tabled at the bottom of the page.

The *calendar* button at the top of the Web interface gives access to old performance plots via a calendar with links to all the dates. They are stored every day at midnight, and give the system behavior for the previous 24 hours. An example is given in Figure 11.

D. Plots

The graphs provided by the web interface are of the following types:

- *Global plots*. These are summary plots which show performance as a function of time for a number of *monitored nodes* on one plot:
  - *superimposed*: the performance data for up to 20 *monitored nodes* is shown superimposed on one plot, enabling a global overview of their behavior, see Figure 13. If the number of *monitored nodes* is more than 20, additional plots are created for readability.
  - *summed*: the aggregate value is shown. This is very useful to measure, for example, the total GDC or network throughput as a function of time, see Figure 12.

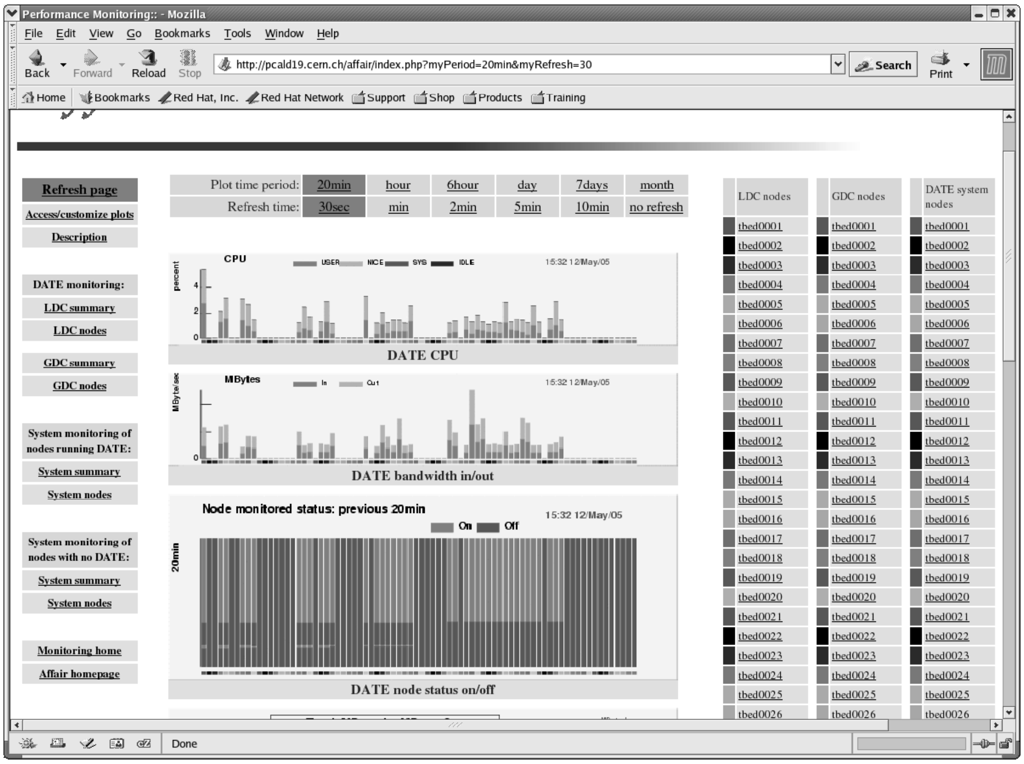

FIG. 9 Top page of the web interface

- *Individual plots*: they show performance as a function of time for a specific *monitored node*. They can show individual metrics, or several metrics can be shown on one plot. This summed option is useful to observe, for example, the user/system/nice/idleCPU values all together on one plot, see Figure 14.
- *Snapshot*: These plots give a snapshot of all *monitored nodes* at the current time and are displayed in the top page of the Web interface (see Figure 9). They provide a clear overview of the entire cluster status. Examples are given in Figures 15 and Figure 16.
- *Status*: The top page also provides the on/off status of all the *monitored nodes* for the selected time period, see Figure 17.

In addition to the average values, the maxima for a particular bin are drawn as superimposed dashed lines in the *superimposed* and *Individual* plots.

E. Collectors

The *Collectors* are the processes which gather the performance data on each *monitored node*. They execute an endless loop, with the period provided by the *AFFAIR Monitor*. At the end of each loop the data is extracted and sent to the *AFFAIR Monitor*. The data is obtained in several ways:

- *System Collector* either parses the /proc directory, or executes system calls. The actual executable is $AFFAIR/src/dim/CollectorDATEMON.
- *DATE Collector* gets all its data from the DATE shared memory. The actual executable is $AFFAIR/src/dim/CollectorDATE, and it requires the SMI handle as a parameter, as well as setting of the DATE_ROLE variable.

Both processes are started via the *AFFAIR/startAffair.sh* script. It starts the *DATE Collector* process if it is not up, and does nothing if it is down. If the *AFFAIR/startAffair.sh* is started with a parameter (default way the

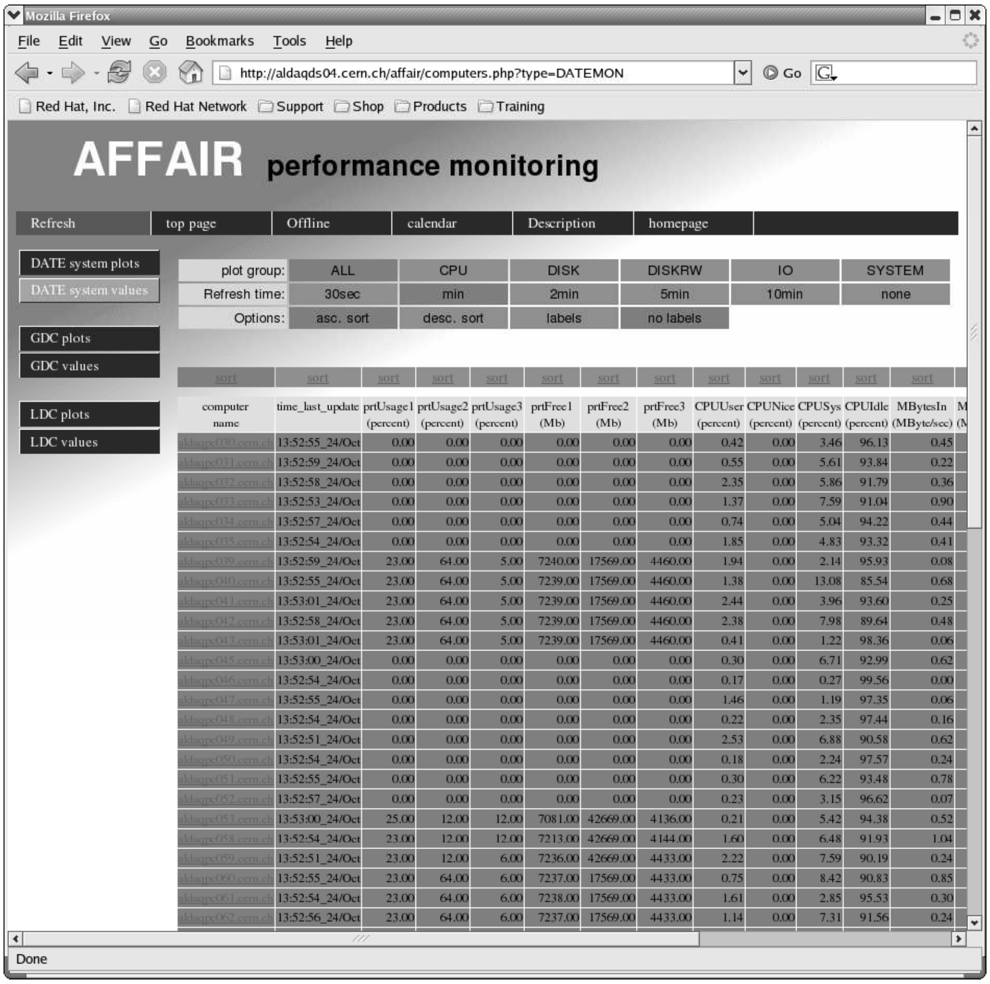

FIG. 10 Web interface to current performance data in tabular form

DATE run control operates.See section V), the parameter is assumed to be the SMI handle, and the *DATE Collector* processes is also started.

F. AFFAIR Monitor

The *AFFAIR Monitor* is the AFFAIR component on the *monitoring station* that gets performance data via DIM communication channels and stores it in temporary round robin databases (RRD) as well as in ROOT permanent storage.

When the *AFFAIR Monitor* is started, it reads the *AFFAIR_COMPUTERLIST* table selected by the *Affair Control* GUI to get a list of all the nodes used, and subscribes to the LDC, GDC and DATEMON services on all the nodes. It then sends configuration parameters to the nodes (see section VII.C), and receives back the final parameters used, as well as some system information, like the kernel version, CPU speed or RAM size. Several configuration and log files are updated with this data.

The *AFFAIR Monitor* then continuously receives (with a period specified in the parameters sent to the *Collectors*) performance data. The data is sent both to the appropriate RRD, and -if offline storage is chosen-, to ROOT storage files. If an RRD and ROOT file does not yet exist for a particular *monitored node* and *monitoring set* they will be created on the fly.

The format for the *AFFAIR Monitor* process is:

- $AFFAIR_MONITOR/src/root/AffairMonitor dummy monitoring_set offline_storage_option

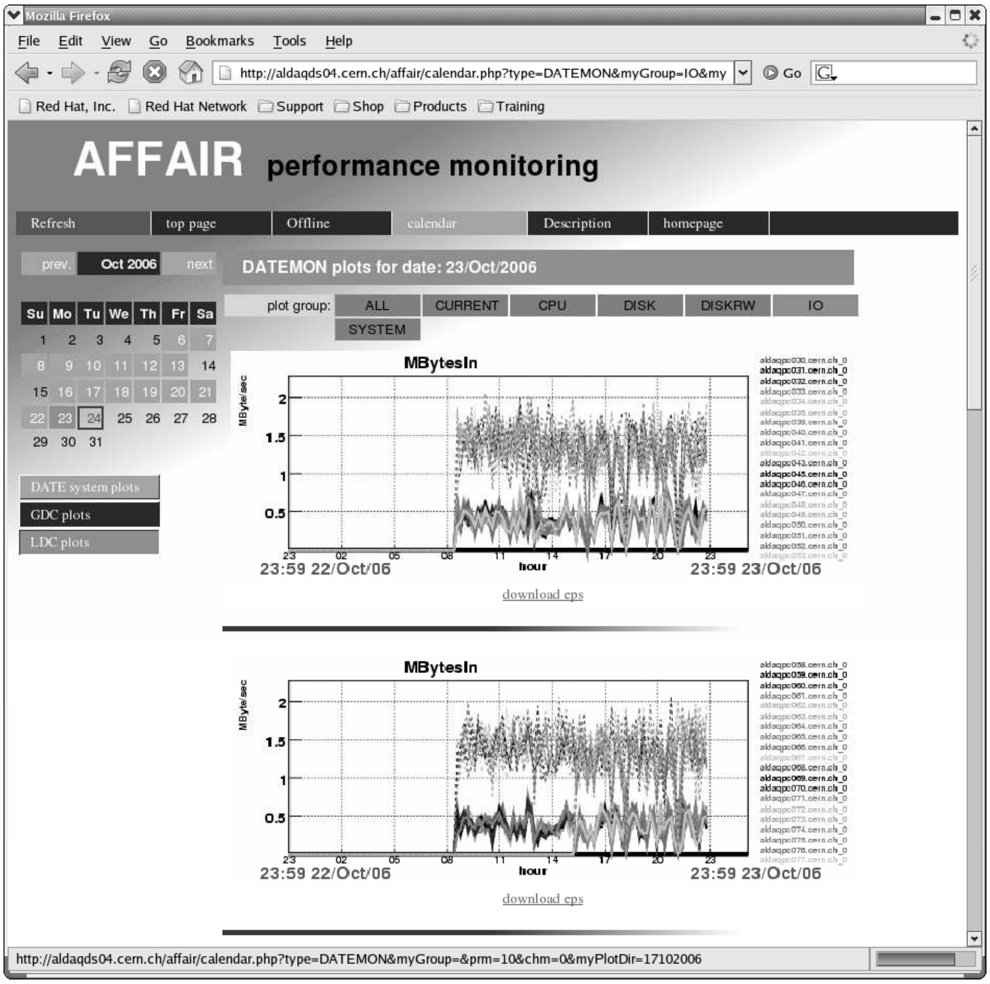

FIG. 11 Web interface to previous performance behavior

The *dummy* parameter is here only for backward compatibility with previous AFFAIR versions with no MySQL support. It is in place of the file containing the list of computers. The *monitoring_set* parameter can be DATEMON,GDC or LDC. The *offline_storage_option* parameter can be either 0, or 1. If it is set to 0, performance data is stored only in RRDs, while if it is set to 1 then performance data is also stored in offline ROOT filesx.

G. Data Processor

The *Data Processor* is ROOT based code to create plots for the Web interface and to perform detailed offline analysis of the monitored data.

The *Data Processor*s loop continuously over all the RRDs, generate global plots and place them in the *$AFFAIR_MONITOR/plots* directory. The plots are in *eps* format since this is the ROOT output format.

The latencies between cycles depends on the number of nodes, and is usually under 1 minute. There is a number of flags regulating the behavior of the *Data Processor*s in the format *-flag=value*, as described in table XIV.

H. Automating the monitoring station

The $AFFAIR_MONITOR/startProcess.sh script enables control of the AFFAIR processes without needing detailed knowledge of the parameters required. It also checks if the selected process is up, and only starts it if it is down. This makes it suitable for cron job operations, which are a good way to ensure that the *monitoring station* processes start at boot time and that they smoothly recover from possible crashes.

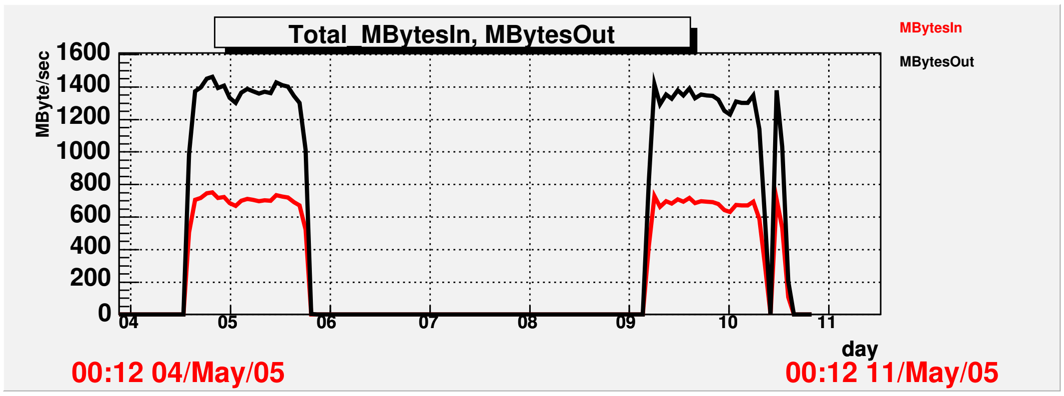

FIG. 12 Aggregate in and out transfer rate for the past seven days

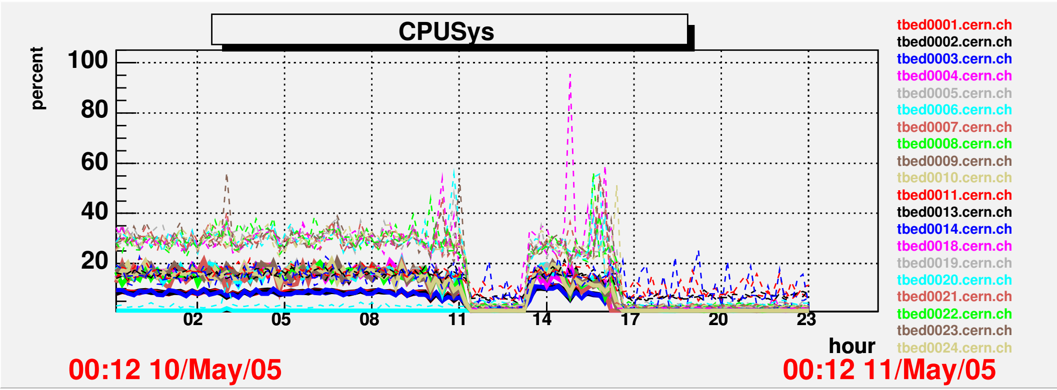

FIG. 13 System CPU status for the past day for 20 nodes

The format for starting the *Data Processor* with default options using the startProcess.sh script is:

- $AFFAIR_MONITOR/startProcess.sh startDataProcessor monitoring_set

The *monitoring_set* parameter can be DATEMON,GDC or LDC.
The format for starting the *AFFAIR Monitor* using the startProcess.sh script is:

- $AFFAIR_MONITOR/startProcess.sh startMonitor dummy monitoring_set offline_storage_option

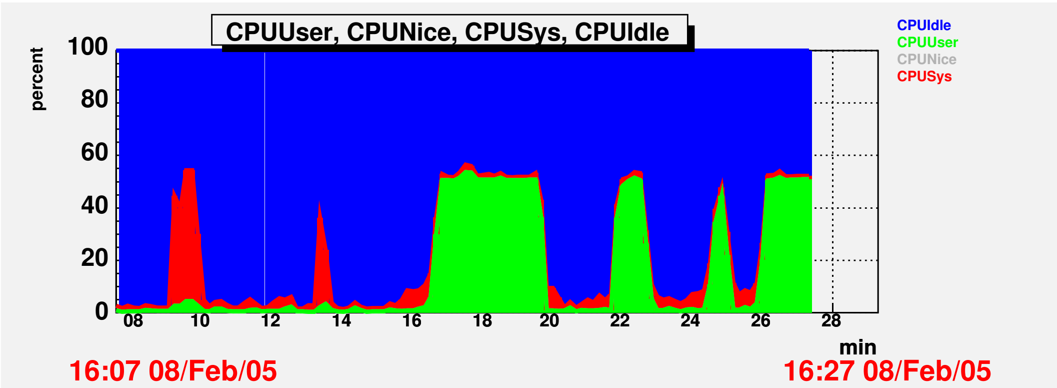

FIG. 14 CPU status for the past 20 minutes for one node

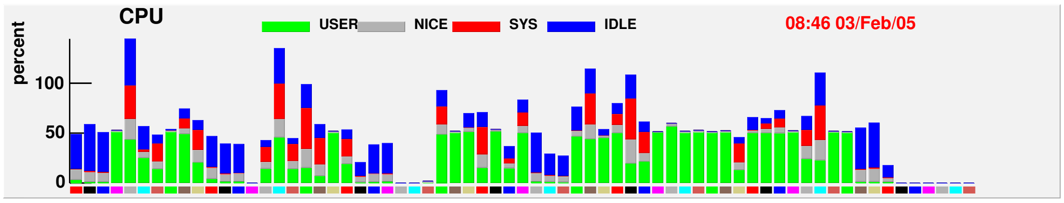

FIG. 15 Current CPU status of all nodes. The bottom is color coded to match the *node* links on the right side of the AFFAIR Web pages

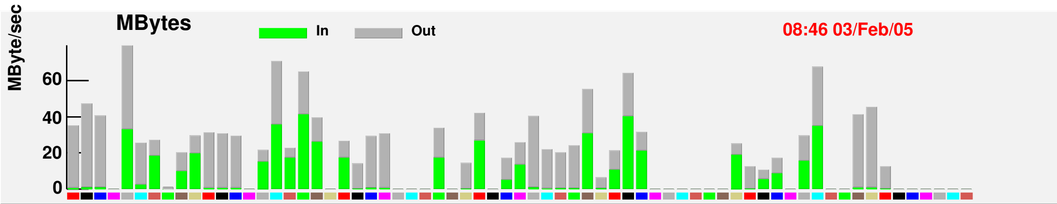

FIG. 16 Current in and out transfer rate of all nodes

The *dummy*,*monitoring_set*, and *offline_storage_option* parameters are explained in Section VIII.F.

There are several options that combine several processes in one command. For example the following command starts monitoring the *nodes* with offline storage for the LDC,GDC and DATEMON *monitoring sets* and with the default plots produced for the web interface:

- $AFFAIR_MONITOR/startProcess.sh monitor

This can be made a cron job that gets started for example every minute by using the *crontab -e* shell command and entering:

- */1 * * * * source ~/.bash_profile; source ~/.bashrc; $AFFAIR_MONITOR/startProcess.sh monitor >&/dev/null &

To disable offline storage, the *monitor_nopermstore* parameter is used instead of *monitor*. The *startProcess.sh* script starts the *AFFAIR Monitor* and several instances of the *Data Processor* (see section VIII.G) with the appropriate flags.

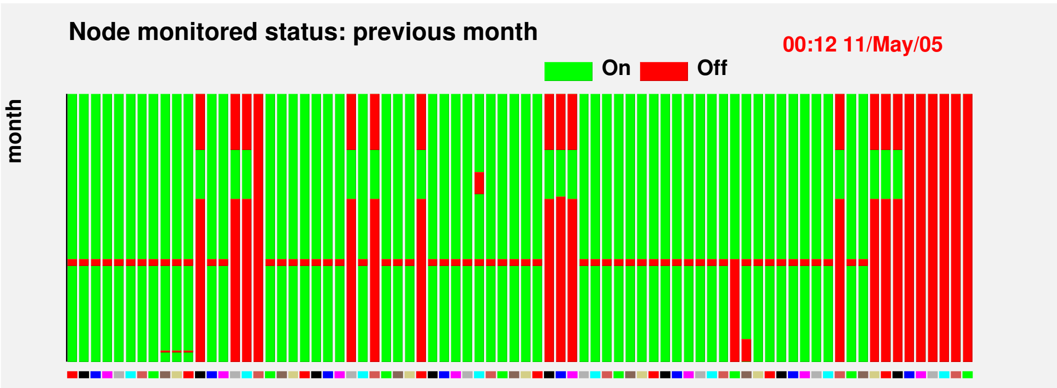

FIG. 17 Status of nodes during the past month

Flag	Description
m	monitorSet. Must be one of the following: GDC, LDC, DATEMON
r	run option. Must be one of the following: • ANA_GLOBAL (offline analysis, creates global plots using a file list), • ANA_EACH (offline analysis, creating plots for individual nodes using a file list), • RRD_GLOBAL(continuously creates global plot, using rrds), • RRD_EACH (continuously creates plots for an individual computer, using rrds), • RRD_ONE_COMPUTER (creates plots once for an individual computer, using rrds. This is called by the web server)
x	number of bins in the x direction for the graphs. Default is 120
p	period option (separate procees for last hour, 20min, day, hourday,month, year, all) This is only relevant with the RRD run options. Default is all
s	sum or superimpose option for global plot creation (SUPERIMPOSE, SUM, BOX,SUM_AND_SUPERIMPOSE, ALL_PLOT_TYPES). Default is ALL_PLOT_TYPES. Only relevant with a GLOBAL run option
b	beginDate (format is Day:Month:Year/Hour:Min:Second). Used with -e. Only relevant with an ANA run option
e	endDate (same format as BeginDate)
c	filename with the list of computers. Default is to use the rrds in the $AFFAIR_MONITOR/rrd directory. With the RRD_ONE_COMPUTER run option it is the actual computer name.
v	index of the variable of a particular *monitoring set* (selected with the -m flag). Used, for example, to analyse a particular run. This is only used with an ANA run option, and requires the -l and -h flags to be set to define the desired range. The time period defined with the -b and -e flags should be larger than this range.
l	lower value of the variable defined with -v flag
h	higher value of the variable defined with -v flag

TABLE XIV Parameters for the Data Processor

IX. OFFLINE STORAGE AND ANALYSIS

The ROOT file is closed and transfered from the *$AFFAIR_MONITOR/rootfiles* to the *$AFFAIR_MONITOR/backup* directory after about five hours of data taking. It is then renamed into a filename indicating the start and end of data taking, as well as the *monitored node* name and the *monitoring set* type, with the following format:

- NodeName_MonitorSet_HH_mm_DD_MMM_YY__HH_mm_DD_MMM_YY.root

For example, if a GDC ROOT file for the *monitored node* pcdaq01.cern.ch was created at 23:52 on the 5th August 2005 and closed at 9:35 on the 6th August, it is named *pcdaq01.cern.ch_GDC_23_52_05_Aug_05__09_35_06_Aug_05.root*

The list of ROOT files saved can be viewed by clicking the view/edit rootfiles saved offline button in the *intial setup* tab

A. Web interface to offline analysis

Detailed analysis of performance data over any time period can be performed offline using the web interface, accessed by the *Offline analysis* button (figure 18). The start times and end times are entered in the appropriate boxes. The time format is HH/MM/SS:DD/MM/YYYY (hour/minute/second:day/month/year). Next, the *monitoring set* is chosen from the *monitoring set* combo box. If instead plots for only one *node* are desired, then this is chosen from the *Generate plots for one node?* combo box. Pressing at this point the *create plots* button will load and process all the ROOT files that contain data for the desired time range and *monitoring set*, using the information in the AFFAIR_ROOTFILELIST table. Note that data for at most the last five hours will not be available, as it is still being currently stored, and has not been placed in the $AFFAIR_MONITOR/backup directory. Depending on the time range selected, the speed of your machine, and the number of *nodes* involved, the processing and plot creation

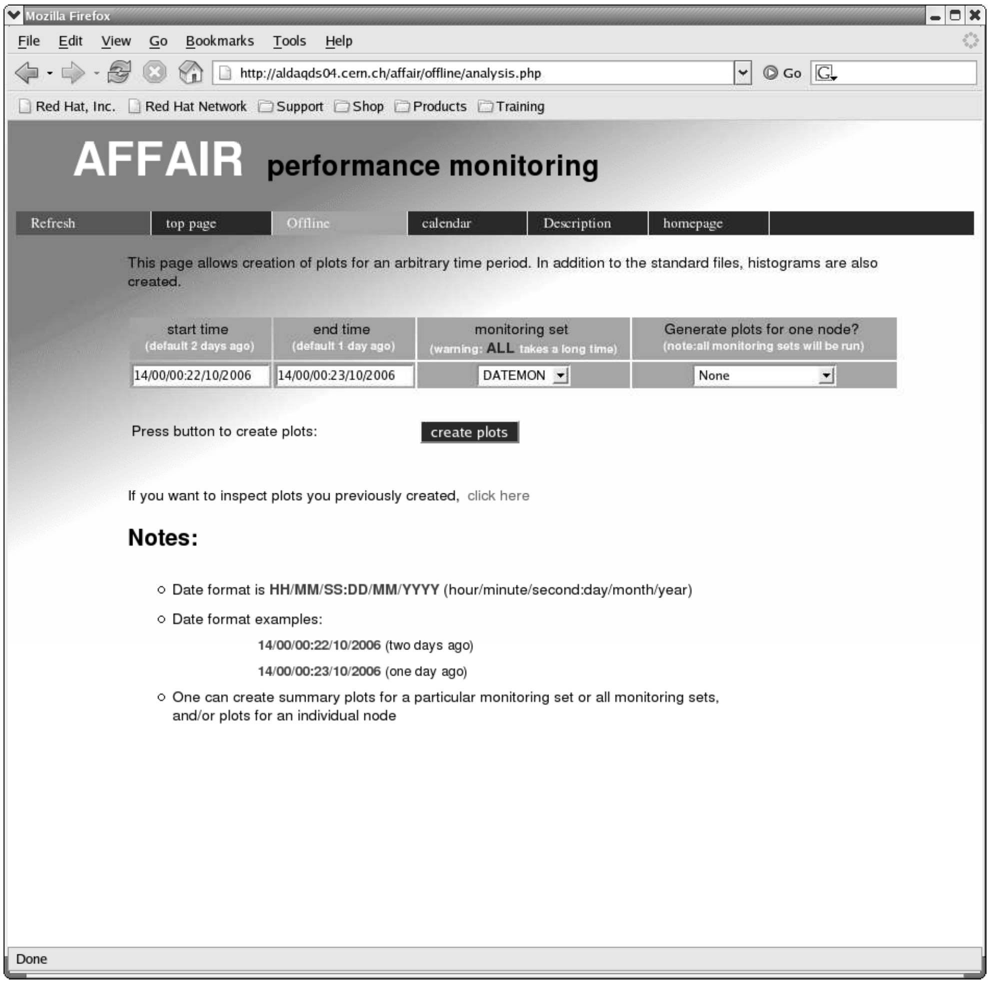

FIG. 18 Offline analysis interface

may take from less than one minute to more than ten minutes. After this all the plots are accesible via the Web interface. Histograms are now also produced, unlike the web based graphs. The performance graphs created are placed in the $AFFAIR_MONITOR/plots/*user* directory with a *user* in the suffix of postscript file name.

For security reasons password protection needs to be set up. The procedure consists of creating three files in *$AFFAIR_MONITOR/web/offline* directory:

- .htaccess
- .htgroup
- .htpasswd

The *.htaccess* file should look like this:

- AuthUserFile /local/affair/web/offline/.htpasswd
- AuthGroupFile /local/affair/web/offline/.htgroup
- AuthName affair_analysis
- AuthType Basic
- <LIMIT GET>
- require group allowed

- </LIMIT>

It is necessary to have the full path for the *AuthUserFile* and *AuthGroupFile* parameters. The *.htgroup* file contains the names of the users. For example, if they are Tom, Dick, and Harry, it would be:

- allowed: Tom Dick Harry

The *.htpasswd* file is created using the *htpssswd* command. For our example, the procedure would be:

- htpasswd -c $AFFAIR_MONITOR/web/offline/.htpasswd Tom
- htpasswd $AFFAIR_MONITOR/web/offline/.htpasswd Dick
- htpasswd $AFFAIR_MONITOR/web/offline/.htpasswd Harry

The *-c* flag is applied only for the first user to create the *.htpasswd* file.

X. LOGGING

The default outputs of *AFFAIR Monitor*, *Data Processor* and *Collector* are sent to *stdout*. In the cron jobs examples given above all the outputs are sent to */dev/null*. This can instead be redirected to any logfile, as the following example shows for the *Data Processor*:

- source $AFFAIR_MONITOR/startProcess.sh startDataProcessor DATEMON >> $AFFAIR_MONITOR/log/dataProcessor.log

A much more verbose output is achieved by adding the *verbose* option as the last parameter of any *$AFFAIR_MONITOR/startProcess.sh* call.

XI. DIRECTORY STRUCTURE

The AFFAIR code is contained in the $AFFAIR_MONITOR directory and distributed in a directory structure as shown in table XV.

XII. UPGRADE

AFFAIR may be upgraded independently of DATE. Unpacking the file affairMonitorNEWVERSION.tar.gz in the $AFFAIR_MONITOR directory and running *make* does not overwrite existing configuration files and databases.

In the case when the upgrade is from 3.0 to 3.2, then one needs to create several new tables, and hence an additional step is required. This is achieved by pressing the button upgrade to version xxx.

XIII. ADDING A NEW MONITORING SET

In addition to the monitoring sets included in AFFAIR (namely DATEMON,LDC,and GDC), it is possible to create new monitoring sets, with new monitored variables. The skeleton *c* files *myMonitoredValues.h* and *myMonitoredValues.c* in the *$AFFAIR/src/dim* directory on the nodes are used to code in the new metrics monitored, while the *Affair Control* GUI is used to define the variables and parameters used, as well as the plotting options.

A. Modifying AFFAIR on the monitoring station

The new monitoring set is created on the monitoring station using the *Affair Control* GUI by clicking on the button marked Create new monitoring set in the *initial setup tab*. The gui in figure 19 appears. Default values are given for the monitoring set name (*MYNEWMONSETSET0*), web text (emphMYNEWMONSETSET0), period monitored (10 seconds), and the flag if the monitoring set is actually used (see section XX for a more detailed explanation). These should be changed to the desired values. In addition, parameters that are passed to the monitored nodes can also be entered here. The rules are the same as before, as explained in section VII.C.

	config		configuration files
	backup		permanent ROOT data storage files
	support		perl and bash scripts
	docs		manual
	rootfiles		current updated ROOT files
	rrd		round robin database location
	plots		generated plots for the Web interface
$AFFAIR_MONITOR	log		log files for the Web interface
	src	dim	code and executables for the *DATE Collector* and *System Collector*
		root	code and executables for the *AFFAIR Monitor*, *Data Processor* and *AFFAIR Control*
	dim		original DIM code
	smixx		original SMI code
	web		PHP and html code for the Web interface

TABLE XV Affair directory structure

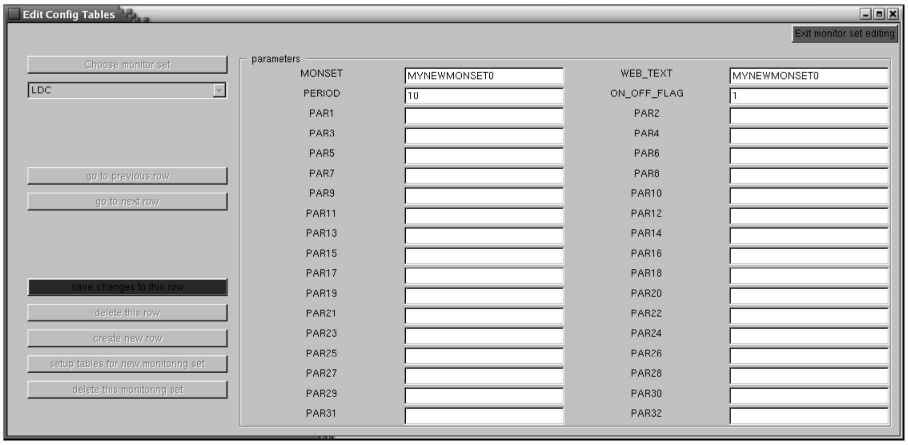

FIG. 19 new monitoring set GUI

To save the changes and additions, the *save changes to this row* button needs to be clicked. This opens a new window where the monitored variables and their plotting options are defined, as described in section VII.B and table III. Initially one variable is created, named *variable0* (see figure ref 20). Change this to the desired name (for example *triggerCount*), possibly change the plotting options, and save the changes by pressing *save changes to this row*. To add a new variable, click *create new row*. Repeat the procedure until all the desired variables are added.

B. Modifying AFFAIR on the monitored nodes

The directory where the AFFAIR code on the *monitoring station* will be downloaded and installed needs to be declared as AFFAIR. The affairCollectorVERSION.tar.gz file (e.g. affairCollector3.20.tar.gz) needs to be downloaded

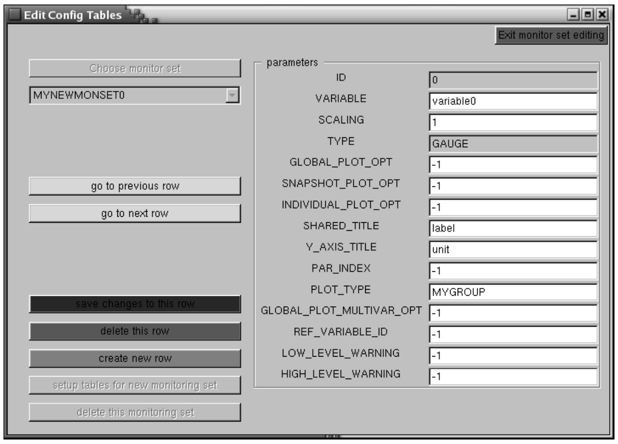

FIG. 20 new variables GUI

to this directory from the AFFAIR site *http://www.cern.ch/affair*. The environment variables for DIM and SMI need to be declared. If DIM and SMI are already installed on the monitored node, then these can be used. Otherwise, one can use the DIM and SMI code that comes with Affair. The best method is to add them to the */.bashrc* file. A typical example,with the Affair DIM/SMI code, is:

- export AFFAIR=/local/affair
- export DIM_DNS_NODE=lxs5013.cern.ch
- export DIMDIR=$AFFAIR/dim
- export SMIDIR=$AFFAIR/smixx
- export DIMBIN=$DIMDIR/linux
- export SMIBIN=$SMIDIR/linux
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DIMBIN:$SMIBIN :$AFFAIR/src/dim;

The $AFFAIR/src/dim directory contains the skeleton *c* files *myMonitoredValues.h* and *myMonitoredValues.c*,and are the only files that need to be modified. All the initialisations, DIM communication, setups, etc are hidden from them and hence the only code needed is the one that actually extracts the wanted values. The first step consists of changing in *myMonitoredValues.h* the lines:

- #define MY_MONVALUE_COUNT 0
- #define MY_PARAMETERS_COUNT 0

to the number of variables as defined in the previous step on the monitoring station via the GUI. For example, if 5 variables are added to the new monitoring set, and two parameters are set by the monitoring station, then change this to:

- #define MY_MONVALUE_COUNT 5
- #define MY_PARAMETERS_COUNT 2

Next, the appropriate code to extract the monitored values is added to *myMonitoredValues.c*. It should be clear what to do by inspecting the code in this file, which contains two functions. The function:

- double getMyMonitoredValues(int monitoredValueIndex)

contains the actual code that gives the monitored values. For example, if we have 5 variables to be monitored, giving in our simplified case the number 1.0, 2.0, 3.0, 5.0, 8.0 respectively , the function would look like this:

- double GetMyMonitoredValues(int i) {
- double myValue;
- if (i==0) myValue = 1.0;
- if (i==1) myValue = 2.0;
- if (i==2) myValue = 3.0;
- if (i==3) myValue = 5.0;
- if (i==4) myValue = 8.0;
- return(myValue);
- }

The function:

- char* GetMyParameters(int parid, char * par, int isParSetByAffairMonitor)

controls what parameters get passed from the monitoring station, and which parameters get returned to the monitoring station. For example, if we defined on the monitoring station GUI that we will have two parameters, with the first fixed, while the second is determined by the node, then the code would like like this:

- char * GetMyParameters(int parid, char * par, int isParSetByAffairMonitor)
- if (isParSetByAffairMonitor == 1) {
- sprintf(fParameterList[parid],par);
- } else {
- if (parid == 1) sprintf(fParameterList[parid],”hda7”);
- }
- return fParameterList[parid];
- }

The program is compiled by typing *make* in the $AFFAIR directory. The program is started by:

- $AFFAIR/src/dim/MyCollector -m MYNEWMONSETSET0

It can also be started by a *cron* job, which also checks if the process is down, and restarts it:

- */1 * * * * source ~/.bash_profile; source ~/.bashrc; $AFFAIR/startProcess.sh startMyCollector MYNEWMONSETSET0 >&/dev/null &

XIV. EDITING PARAMETERS USED FOR PLOTTING AND MONITORING

Several setup parameters can bu controlled via the GUI. They are accessed by clicking on the *view/edit warning colors, timeouts etc* button in the *initial setup tab* window, as shown in figure 21. If shaded in gray, the feature is not implemented yet, but will be soon. One important setup parameter is USE_TIME_FROM_REMOTE_NODE. If this is set to 1, then the timestamp of the arrived data will be the one on the monitored node. This option should be used if all the nodes have synchronized times. If this is not set to 1, then the local time on the monitoring station will be used as the timestamp for the arriving data.

References

[1] http://www.cern.ch/affair, the AFFAIR web site
[2] http://root.cern.ch, the ROOT web site
[3] http://httpd.apache.org, the apache web site
[4] http://www.mysql.com/, the mysql web site
[5] http://www.php.net, the PHP web site
[6] http://people.ee.ethz.ch/õetiker/webtools/rrdtool/, the rrdtool web site

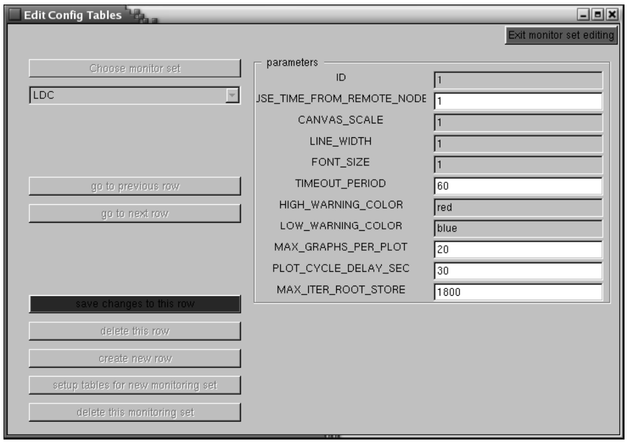

FIG. 21 setup parameters