RC25260 (W1201-029) January 16, 2012
Computer Science

IBM Research Report

AHAFS Enables AIX® Event Monitoring without Writing Code

Joefon Jann, Niteesh Dubey
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

AHAFS Enables AIX® Event Monitoring Without Writing Code

Authors: Joefon Jann and Niteesh Dubey, IBM T.J. Watson Research Center, NY, USA
Email: joefon@us.ibm.com, niteesh@us.ibm.com

Starting with AIX 6100-06 and 7100-00, IBM® introduced the AIX Event Infrastructure for monitoring pre-defined and user-defined system events—such as modification of a file’s content, utilization of a filesystem exceeding a user-defined threshold, death of a process or a change in the value of a kernel tunable parameter—without the high overhead of polling. This infrastructure can automatically notify registered users or processes instantly about the occurrences of such events, with information useful for maintaining and improving the health and security of the running AIX instance. The information provided in the notification includes the what, when, who, and where the event happened, and can include the whole function call-chain that triggered the event.

At the core of the AIX Event Infrastructure is a pseudo-filesystem: Autonomic Health Advisor FileSystem (AHAFS), which is implemented as a kernel extension. AHAFS mainly acts as a mediator to take the requests of event registration, monitoring and unregistering from the processes interested in monitoring for events. It forwards the requests to the corresponding event producers (code responsible for triggering the occurrence of an event) in the kernel space, processes the callback functions when the event occurs, and notifies the registered users or processes with useful information.

The key features of this infrastructure are:

- It requires no new API for monitoring events. The monitoring applications just need to use the existing filesystem interfaces (e.g. **open(), write(), select(), read(), close()**) in AIX, which are supported by many programming languages like C, C++, Java, Perl, etc., to register/monitor/unregister the events.
- It provides detailed information—such as stack trace, program name, time stamp, user and process information—about the occurrence of an event.
- The same event can be monitored by many users or processes, each with a different threshold.
- Different levels of information can be extracted by the different users or processes upon the occurrence of an event.
- Any component or subcomponent in the kernel space, including kernel extensions and device drivers, can register its own event producers to AHAFS to enable the monitoring of its events. For example, it can monitor events on files, filesystems, kernel tunables on CPU or memory, and even networking events.

Setting up AHAFS

By default, the AHAFS fileset—bos.ahafs—comes with AIX. If bos.ahafs is installed, you can type the following commands as a root user to create a mount point (typically **/aha**) and then mount the AHAFS filesystem as follows:

```
mkdir /aha
mount -v ahafs /aha /aha
```

The mounting of the AHAFS will create the following items:

- The file **/aha/evProds.list**, which contains the list of pre-defined and user-defined event producers available to this AIX instance. This is a special file whose content can be listed by typing:

  ```
      cat /aha/evProds.list
  ```

- The components for grouping the monitor factories. These are the subdirectories **under /aha named mem/, cpu/, fs/,** etc.
- The monitor factories (i.e. subdirectories under the component subdirectories with the filetype of **.monFactory**) each of which corresponds to an event producer for AHAFS. One can also list the available monitor factories by typing:

  ```
      du -a /aha  | grep .monFactory
  ```

Table 1: lists the pre-defined event producers that come with AIX:

Pre-defined Event Producer	**It Notifies Consumers When?**
utilFs	The utilization of a monitored filesystem crosses the user-specified threshold.
modFile	The contents of a monitored file are modified.
modFileAttr **(6100-07 and 7100-01)**	The attributes (e.g. modebits, access control list, ownership) of a monitored file are modified.
modDir	A file or subdirectory is created, renamed or deleted under a directory.
schedo	The value of a monitored scheduler-tunable has been changed.
vmo	The value of a monitored VMM-tunable has been changed.
waitTmCPU	The average wait time of all the threads waiting for CPU resources exceeds the user-specified threshold.
waitersFreePg	The number of waiters to get a free page frame in the last second exceeds the user-specified threshold.
waitTmPginOut	The average wait time of all the threads waiting for page-in or page-out operations to complete exceeds the user-specified threshold.
processMon or pidProcessMon	The monitored process exits.

Cluster Aware AIX (CAA), which is included in the standard and enterprise editions of AIX, provides the following event producers for the cluster.

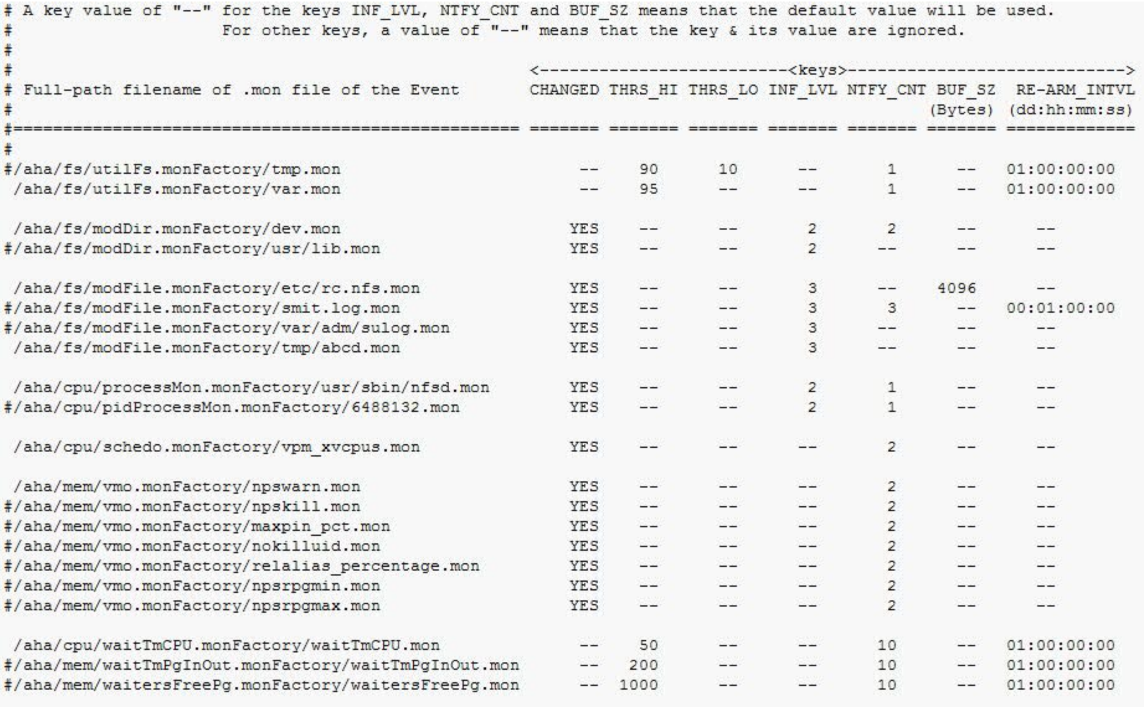
Node/host:	**nodeList, nodeState, nodeContact, linkedCl, nodeAddress**
Network:	**networkAdapterState**
Disk:	**diskState, clDiskList, clDiskState, repDiskState, vgState**

No Coding is Required

The AHAFS is shipped with many useful sample programs and scripts under the directory **/usr/samples/ahafs/** designed to make AHAFS very easy to use. The directory **/usr/samples/ahafs/bin/** contains the script **aha.pl** and its input file **aha-pl.inp** which can be used to monitor events without writing any code. All you need to do is make a copy of the **aha-pl.inp** file and modify it by uncommenting lines in the file for things you want to monitor; or modify some key values if you want to use values other than the defaults. Then invoke the **aha.pl** script specifying your tailored **.inp** file, and optionally specifying the list of email IDs to which you want the monitoring reports to go to.

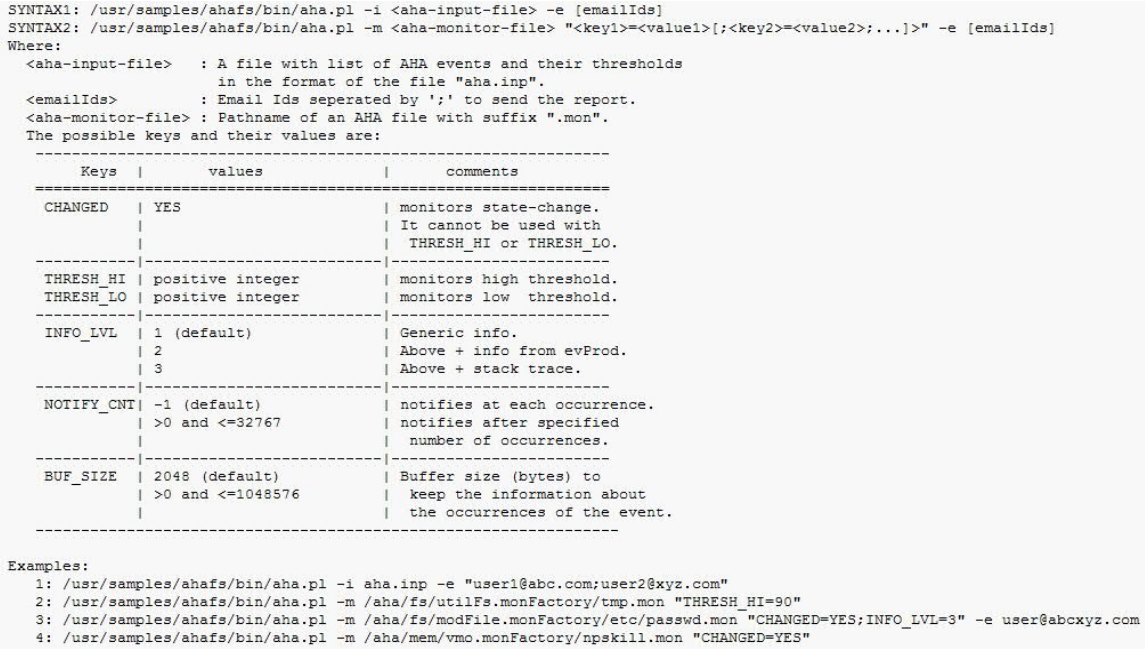
Figures 1 and 2 illustrate examples of these details:

Figure 1: Content of the file /usr/samples/ahafs/bin/aha-pl.inp



```
# A key value of "--" for the keys INF_LVL, NTFY_CNT and BUF_SZ means that the default value will be used.
#                     For other keys, a value of "--" means that the key & its value are ignored.
#
#                                                    <-------------------------<keys>---------------------------->
# Full-path filename of .mon file of the Event       CHANGED THRS_HI THRS_LO INF_LVL NTFY_CNT BUF_SZ  RE-ARM_INTVL
#                                                                                             (Bytes) (dd:hh:mm:ss)
#=================================================== ======= ======= ======= ======= ======= ======= =============
#
#/aha/fs/utilFs.monFactory/tmp.mon                        --    90      10      --       1      --   01:00:00:00
 /aha/fs/utilFs.monFactory/var.mon                        --    95      --      --       1      --   01:00:00:00

 /aha/fs/modDir.monFactory/dev.mon                       YES    --      --       2       2      --      --
#/aha/fs/modDir.monFactory/usr/lib.mon                   YES    --      --       2      --      --      --

 /aha/fs/modFile.monFactory/etc/rc.nfs.mon               YES    --      --       3      --    4096      --
#/aha/fs/modFile.monFactory/smit.log.mon                 YES    --      --       3       3      --   00:01:00:00
#/aha/fs/modFile.monFactory/var/adm/sulog.mon            YES    --      --       3      --      --      --
 /aha/fs/modFile.monFactory/tmp/abcd.mon                 YES    --      --       3      --      --      --

 /aha/cpu/processMon.monFactory/usr/sbin/nfsd.mon        YES    --      --       2       1      --      --
#/aha/cpu/pidProcessMon.monFactory/6488132.mon           YES    --      --       2       1      --      --

 /aha/cpu/schedo.monFactory/vpm_xvcpus.mon               YES    --      --      --       2      --      --

 /aha/mem/vmo.monFactory/npswarn.mon                     YES    --      --      --       2      --      --
#/aha/mem/vmo.monFactory/npskill.mon                     YES    --      --      --       2      --      --
#/aha/mem/vmo.monFactory/maxpin_pct.mon                  YES    --      --      --       2      --      --
#/aha/mem/vmo.monFactory/nokilluid.mon                   YES    --      --      --       2      --      --
#/aha/mem/vmo.monFactory/relalias_percentage.mon         YES    --      --      --       2      --      --
#/aha/mem/vmo.monFactory/npsrpgmin.mon                   YES    --      --      --       2      --      --
#/aha/mem/vmo.monFactory/npsrpgmax.mon                   YES    --      --      --       2      --      --

 /aha/cpu/waitTmCPU.monFactory/waitTmCPU.mon              --    50      --      --      10      --   01:00:00:00
#/aha/mem/waitTmPgInOut.monFactory/waitTmPgInOut.mon      --   200      --      --      10      --   01:00:00:00
#/aha/mem/waitersFreePg.monFactory/waitersFreePg.mon      --  1000      --      --      10      --   01:00:00:00
```

Figure 2: SYNTAX of the script /usr/samples/ahafs/bin/aha.pl



```
SYNTAX1: /usr/samples/ahafs/bin/aha.pl -i <aha-input-file> -e [emailIds]
SYNTAX2: /usr/samples/ahafs/bin/aha.pl -m <aha-monitor-file> "<key1>=<value1>[;<key2>=<value2>;...]>" -e [emailIds]
Where:
  <aha-input-file>   : A file with list of AHA events and their thresholds
                       in the format of the file "aha.inp".
  <emailIds>         : Email Ids seperated by ';' to send the report.
  <aha-monitor-file> : Pathname of an AHA file with suffix ".mon".
  The possible keys and their values are:
   ---------------------------------------------------------------
        Keys  |       values              |      comments
   ===============================================================
    CHANGED   | YES                       | monitors state-change.
              |                           | It cannot be used with
              |                           |  THRESH_HI or THRESH_LO.
   -----------|---------------------------|------------------------
    THRESH_HI | positive integer          | monitors high threshold.
    THRESH_LO | positive integer          | monitors low  threshold.
   -----------|---------------------------|------------------------
    INFO_LVL  | 1 (default)               | Generic info.
              | 2                         | Above + info from evProd.
              | 3                         | Above + stack trace.
   -----------|---------------------------|------------------------
    NOTIFY_CNT| -1 (default)              | notifies at each occurrence.
              | >0 and <=32767            | notifies after specified
              |                           |  number of occurrences.
   -----------|---------------------------|------------------------
    BUF_SIZE  | 2048 (default)            | Buffer size (bytes) to
              | >0 and <=1048576          |  keep the information about
              |                           |  the occurrences of the event.
   ----------------------------------------------------------------

Examples:
   1: /usr/samples/ahafs/bin/aha.pl -i aha.inp -e "user1@abc.com;user2@xyz.com"
   2: /usr/samples/ahafs/bin/aha.pl -m /aha/fs/utilFs.monFactory/tmp.mon "THRESH_HI=90"
   3: /usr/samples/ahafs/bin/aha.pl -m /aha/fs/modFile.monFactory/etc/passwd.mon "CHANGED=YES;INFO_LVL=3" -e user@abcxyz.com
   4: /usr/samples/ahafs/bin/aha.pl -m /aha/mem/vmo.monFactory/npskill.mon "CHANGED=YES"
```

Simple Steps/Calls

The directory **/usr/samples/ahafs/samplePrograms/evMon/** contains simple sample programs, written in C, Java or Perl, to monitor for one AHAFS event, and you can use them as is. If you want to add AHAFS monitoring to an existing monitoring program written in C, you can simply add five function calls, as listed below. For example, to monitor for modifications to the contents of the **/etc/passwd** file, your program will:

1. Call **open()** to create and open the event file so as to get a file-handle/descriptor: e.g. Call open() on the .mon file **/aha/fs/modFile.monFactory/etc/passwd.mon**
2. Call **write()** to register for event monitoring by writing your monitoring interests: e.g. Call write() to write the monitoring interest of "CHANGED=YES" into the file opened in step 1. In fact, all of the monitoring interests are expressed as *<key>=<value>* pairs separated by a semi-colon (;) or space. A complete list of keys and their possible values can be found in the Reference section.
3. Call **select()** on the above file-handle to start monitoring, and wait until the event occurs (i.e. the file **/etc/passwd** has been modified).
4. Call **read()** on the above file-handle to get information about the occurrence of the event.
5. Call **close()** or **exit()** on the above file-handle to unregister the monitoring of the event.

Additional examples of how to add event producers from your own kernel extension code or device drivers code, enabling users to monitor events, are provided under the **/usr/samples/ahafs/samplePrograms/kextEvProd/** directory that comes with AIX.

On the Horizon

Enhancements to AHAFS are planned in upcoming releases of AIX, so keep an eye out for those. We welcome feedback on what you monitor with AHAFS, which features or event producers you find most useful, and what enhancements to AHAFS will make your AIX instances even more healthy and carefree.

Bios

Joefon Jann is a Distinguished Engineer at the IBM Watson Research Center in New York. After contributing to the Deep Blue Chess machine and HPC computing on the SP2/SP3, she continued to work at IBM Watson Research in the past 15 years on research projects that have contributed to improving AIX and PowerVM in resiliency, performance, usefulness and consumability. She can be contacted at joefon@us.ibm.com.

Niteesh Dubey is a Senior Engineer at the IBM Watson Research Center in New York. He has worked there for more than a decade on research projects that have contributed to improving AIX and PowerVM in resiliency, performance, usefulness and consumability. He can be contacted at niteesh@us.ibm.com.

References

"AIX Event Infrastructure for AIX and AIX Clusters—AHAFS"
http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.baseadmn/doc/baseadmndita/aix_ev.htm

"Operating System and Device Management," AIX 7.1 manual, the AHAFS chapters are pages 539-612
http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.baseadmn/doc/baseadmndita/baseadmndita_pdf.pdf

"AIX 7.1 Differences Guide" Redbook publication SG24-7910-00 12/2010, pages 222-234 describe the AIX Event Infrastructure
http://www.redbooks.ibm.com/redbooks/pdfs/sg247910.pdf

"An Introduction to Event Monitoring Using the AIX Event Infrastructure," developerWorks article
http://www.ibm.com/developerworks/aix/library/au-aix_event_infrastructure/index.html?ca=drs-

"Monitoring Events in an AIX Cluster," developerWorks article
http://www.ibm.com/developerworks/aix/library/au-aixcluster/index.html