BESSY

Berliner Elektronenspeicherring-Gesellschaft
für Synchrotronstrahlung m.b.H.

Administration of Soft IOCs under Linux

Ralph Lange (BESSY)

Leibniz
Gemeinschaft

Objectives

- BESSY II is still on R 3.13.9
- The MLS Control System (based on 3.14) uses soft (host based) IOCs in production for the first time
- An infrastructure for soft IOCs had to be set up, that ensures
  - Soft IOCs are started and stopped by the system (like system services)
  - The application developer may manually start/stop any soft IOC
  - Soft IOCs may be shifted easily to another host in case of hardware failure
  - Console access and logging has to be done the same way as for VME IOCs
  - CA should be able to distinguish between soft IOCs (access security, client side debugging)

Running the Soft IOCs

Considerations

- Virtualization (VMware) was looked at and dropped: too thick for the small benefit of individual IPs for the soft IOCs
- Running the soft IOCs as dedicated users (one for each soft IOC) at least allows Access Security to tell them apart and keeps process listings obvious
- Background operation with attachable console is provided by using the screen application

Approach

- All soft IOCs run on one dedicated Linux host – another host is completely configured as cold standby
- An /etc/init.d/softIOC script
  - reads in a simple configuration file, allowing to set environment variables for each of the soft IOCs, and select the soft IOCs that are started automatically
  - takes a list of IOC names as argument to allow manual starting/stopping
  - starts the soft IOC in a screen session run as the dedicated user
- This fits neatly into the regular system service administration framework

Accessing the Soft IOC Console

Considerations

- When the soft IOC gets started, its screen session gets the soft IOC name
- Using that name, you can use the screen client to connect to an existing session
- For ssh connections, you can configure a command to be executed for each key into the ~/.ssh/authorized_keys file

Approach

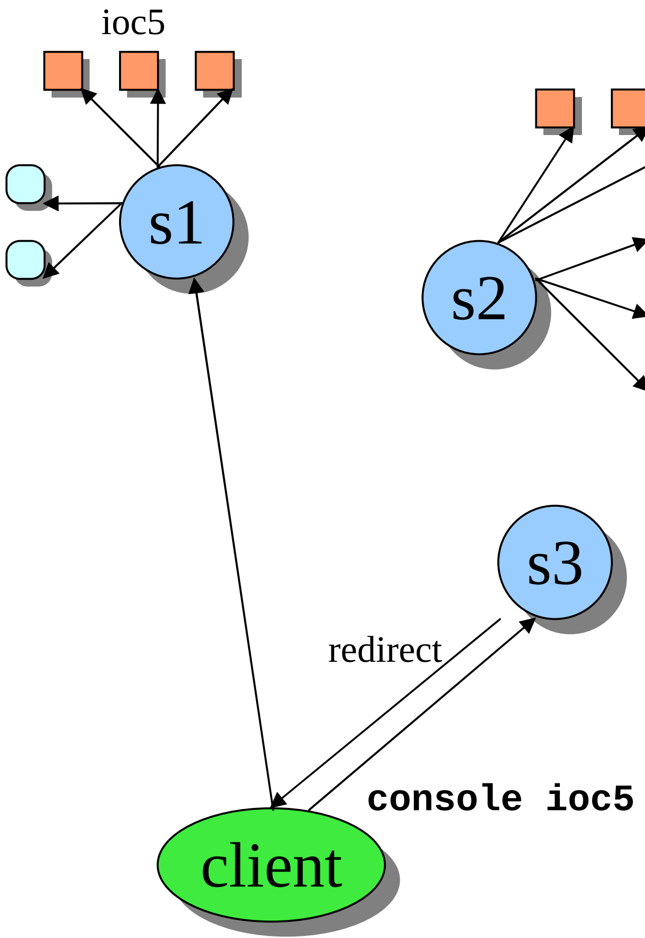
- A key pair for console access was created
- For each of the soft IOC users, the matching screen client command was put in the ~/.ssh/authorized_keys file with the console access key
  E.g., for the user sioc5cp running the soft IOC sioc5cp under the screen session name sioc5cp, the line reads:
  `command="screen -r sioc5cp" ssh-rsa AAAAB3NzaC1yc2EAAAA.....`
- Connecting by ssh to the user sioc5cp on the soft IOC host using the console access key will directly connect you to the soft IOC's console

Providing Console Access and Logging for all IOCs

Conserver

- allows multiple users to watch a serial console at the same time (write access for one at a time; you can force taking over a console if you have sufficient rights)
- does authentication/authorization: user names, access rights (read/write/admin), password file or PAM
- logs the data, adding timestamps (every *m* lines, every *n* seconds, configurable format)
- plays back a motd and/or a configurable amount of history when client connects to a console
- client-server architecture using TCP (server can allow or require SSL)
- a network of multiple console servers sharing the same configuration will redirect a client connection to the appropriate server
- console server connects to the consoles using telnet, local serial line, arbitrary client application

Example: The Conserver Setup at BESSY



s3 is configured as default console server

- Two console servers (one in each facility) monitor the IOC consoles (using telnet to terminal servers) and soft IOC consoles (using ssh)
- A third server is fully configured and can take over easily when there's a failure
- The client machines can use any one of these servers. The suggested way is defining an DNS alias “console” for all domains and configure all machines to go to “console”
- Typing “console <iocname>” will get you to the console, no matter where it is hosted
- Logrotate is configured to keep two weeks of individual logs, two months of an unified log
- Access to log files is provided by cron jobs rsync'ing all log files to one location and running a web server providing access to that set of files

Logfiles and Problems

Who cares about log events?

Nobody cares about log events.

Everybody cares about the problems that cause log events.

(John P. Rouillard at the LISA 2004 conference)

Problems can be:

- something that happened
- something that didn't happen
- something that didn't happen fast enough after something else happened
- something that happened too often in a certain time span
- something that didn't happen at a certain time
- ...

The log events have to be correlated to each other and to time to generate useful problem information.

In the Pipe: Logfile Analysis

SEC (Simple Event Correlator)

- SEC is a highly configurable, rule-based tool for analyzing events in a file stream and take actions based on the results.
- Events are matched by regular expressions, perl subroutines etc.
- Actions can be user-specified shell scripts or programs
- SEC will be used to create a logfile analysis mechanism that identifies events and takes appropriate actions. Write email, send SMS messages, create problem reports in our Trac system, ...

In the Pipe: Logfile Analysis

Correlations

- *Possible correlations cover a wide complexity range:*

  Single - match input event and execute an action list.

  ...

  Pair - match input event, execute an action list, and ignore the following matching events until some other input event arrives. On the arrival of the second event execute another action list.

  ...

  SingleWith2Thresholds - count matching input events during t1 seconds and if a given threshold is exceeded, execute an action list. Then start the counting of matching events again and if their number per t2 seconds drops below the second threshold, execute another action list. Both event correlation windows are sliding.

- Correlations work across log files