Using iRods Rules and micros-services

Mwan@diceresearch.org

Overview

- Structure of the rule language
- Components of the rules language and how they fit together
- Use of rule language for rule and workflow designs.

iRods Rules

- Each rule defines
  - An action for an event
  - Condition
  - Action chains (micro-services and rules)
  - Recovery chains
- Invoked by servers to enforce policies
- Invoked by clients to run workflows on servers
- Rule types
  - Atomic -- applied immediately
  - Deferred -- run at a later time in the background
  - Periodic – run at a fix time interval

Format of a Rule

- Action | Condition | MS$_1$, …, MS$_n$ | RMS$_1$, …, RMS$_n$

- Action
  - Name of action to be performed
  - Name known to the server and invoked by server
- Condition – condition under which the rule apply
- Micro-services - If applicable micro services will be executed
- Recovery micro-service - If any micro service fails, recovery micro service(s) executed to maintain transactional consistency
- Example of MS/RMS

createFile(*F)	removeFile(*F)
ingestMetadata(*F,*M)	rollback

Condition

- Condition under which this Rule applies
- Examples
  - $rescName == demoResc8
  - $objPath like /x/y/z/*

- Many operators
  - ==, !=, >, <, >=, <=
  - %%, !!   (and, or)
  - expr like reg-expr , expr not like reg-expr , expr ::= string

Rule parameters

- A long list of session system parameters –
  - Start with ‘$’
  - Similar to global parameters
  - $rescName , $objPath, $rescGroupName, $dataType, $dataSize, $chksum, $dataOwnerName, $dataId, $collId, $dataExpiry, $dataCreate
- Parameters that are passed between micro-services
  - Start with * - variable
  - Literal string

Micro-services (MSs)

- Well-defined Server-side Procedures and Functions
- C functions on servers
- MSs can be chained to form workflow using ‘##’
  - msiDataObjOpen(*A,*S_FD)##msiDataObjRead(*S_FD,10000,*R_BUF)##msiDataObjClose(*D_FD,*stat)
- Flow control
  - whileExec - while loop
  - forExec – for loop
  - forEachExec – for each in the table or list
  - break
  - ifExec – if-else

Micro-services – flow control examples

- whileExec
  - assign(*A,0)##whileExec(*A < 20 ,writeLine(stdout,*A)##assign(*A, *A + 4), nop##nop)
  - forExec
    - forExec(assign(*A,0), *A < 20 , assign(*A,*A + 4), writeLine(stdout,*A),nop)
  - ifExec
    - ifExec(*A > *D,assign(*A,*D),nop,assign(*D,*A),nop)

Other Micro-services

- delayExec - execute MSs at a later time
  - Exec by the iRods batch server (irodsReServer) in the background
  - Example
    - delayExec(<PLUSET>1m</PLUSET>,msiReplColl(*desc_coll,*desc_resc,backupMode,*outbuf),nop)
  - Time keywords
    - PLUSET – exec after the specified time has passed
    - ET – exec at the specified time (<ET>23:00</ET>)
    - FT – repeat exec at the specified frequency
    - Can be combined
      - <PLUSET>1m</PLUSET><EF>5m</EF>
- remoteExec – execute MSs on remote servers
  - remoteExec(andal.sdsc.edu,null,msiSleep(10,0)##writeLine(stdout,open remote write in andal), nop)
- assign - assign a value to a parameter
- writeString - write a string to stdout buffer
- writeLine - write a line (with end of line) to stdout buffer

Micro-Services parameters

- Micro-services communicate through:
  - Arguments/Parameters
    - Input from the initiator (client/server)
      - Lieterals
      - Variables
        - start with *
        - Output of a MS can be used as input of another MS in a MS chain
  - System Session Parameters
    - Satart with “$”
    - Valid across rule invocations
  - Persistent data – iCat
    - Query the iCat
    - Valid across sessions
  - XMessages – out-of-band communications
    - Sender obtains send/receive tickets
    - Pass receive ticket to receivers
    - Receiver use ticket to read msg
    - Msg exchange
      - Between Parallel Session
      - Between the batch manager and the task manager on the task status

Example of passing parameters between Micro-services

- trimColl.ir file:
  - myTestRule||acGetIcatResults(*Action,*Condition,*B)##forEachExec(*B,msiDataObjTrim(*B,tgReplResc,null,1,null,*C),nop)|nop##nop
  - *Action=trim%*Condition= COLL_NAME = '/tempZone/home/rods/loopTest'
  - *Action%*Condition
- irule –F trimColl.ir

Micro-services for managing the core.irb file

- showCore.ir – list the rules in core.irb

```
myTest||msiAdmShowIRB(*A)|nop
```

- chgCoreToOrig.ir - replace core.irb with core.irb.orig

```
myTest||msiAdmChangeCoreIRB(*A)|nop
*A=core.irb.orig
```

- chgCoreToCore1.ir – append the rules in core.irb1 on top of core.irb

```
myTest||msiAdmAppendToTopOfCoreIRB(*A)|nop
*A=core.irb.1
```

  - whatever rules on top are executed first
  - Rules in core.irb.1 overides core.irb