



Integrate z/VM into your z/OS and Linux for System z Scheduling Environments

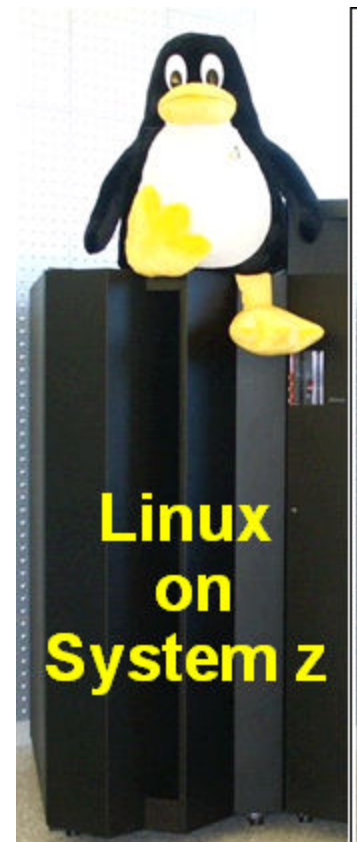
Mike Sine
IBM

Friday, February 8, 2013
Session Number: 12465



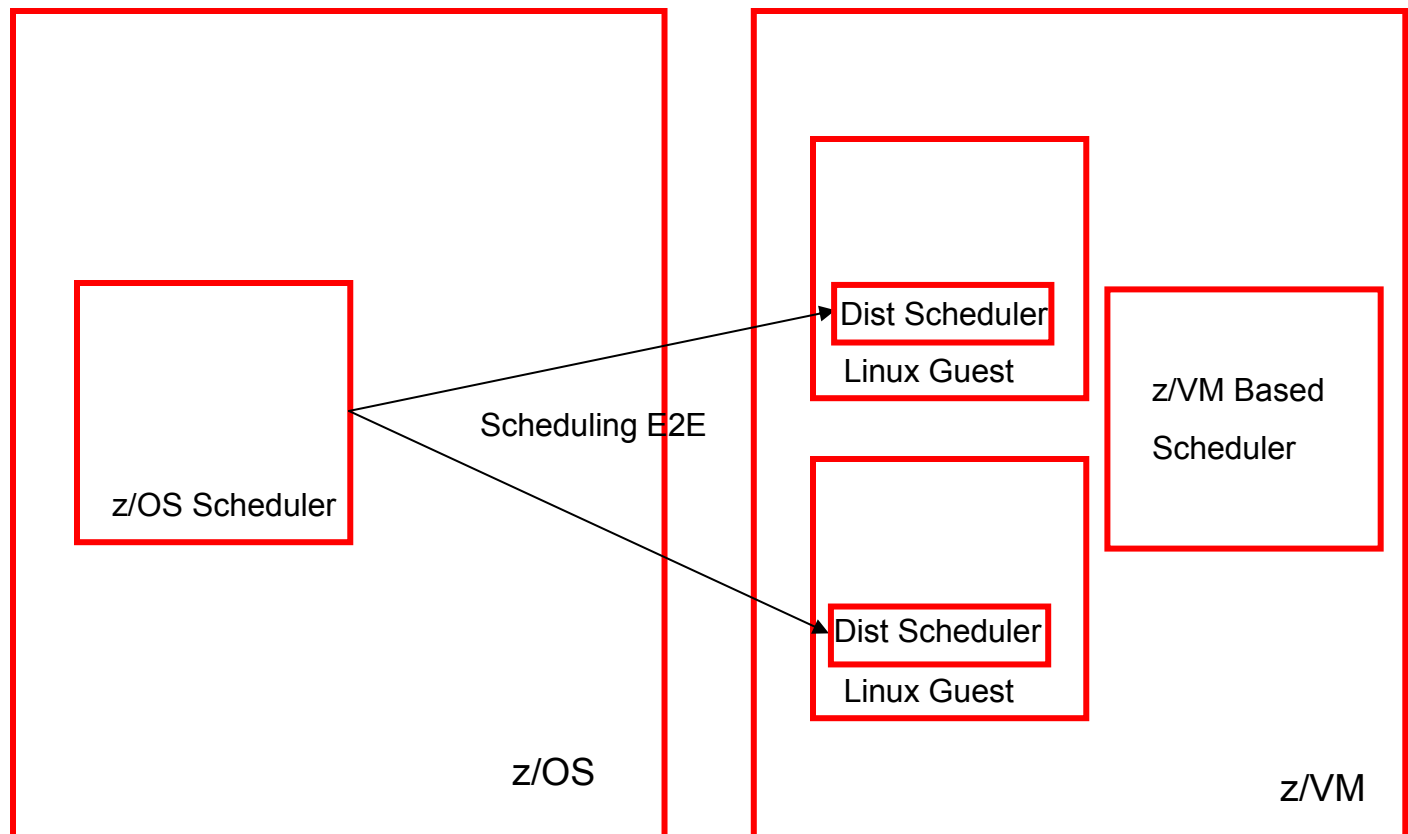
Session Abstract

Many customers today have integrated Linux for System z into their z/OS scheduling environments. They successfully manage dependencies across these two platforms as well as other non-mainframe platforms. z/VM is often omitted in scheduling environments. By integrating z/VM into your z/OS and Linux for System z scheduling environments improvements in dependency resolution may be realized by interaction with z/VM, tasks specific to z/VM may be included in the scheduling scenarios, and centralizing more management functions (including scheduling) at the z/VM level may reduce resources utilization at the Linux level. This session will introduce these concepts and discuss the potential improvements and benefits of z/VM's inclusion to the scheduling environment. The speaker will demonstrate example scenarios using Tivoli Workload Schedule End-to-End (TWS E2E) with IBM's Operations Manager for z/VM. Attendees will be able to take these examples and implement similar functionality using their choice of z/OS and Linux based schedulers as well as other z/VM Tooling (PROP, or other vendor solutions).



Traditional End-to-End Scheduling Techniques

What's missing?



Why Schedule on z/VM?

Integrate z/VM into your System z scheduling environments to:

- Reduce computer operator errors by automating routine tasks with timers or scheduled events and actions
- Centralizing more management functions (including scheduling) at the z/VM level may simplify management effort.
- Improvements in dependency resolution may be realized by interaction with z/VM & Linux. (ex: guest logoff)
- Scheduling and event triggered automation integrate well.
- Tasks specific to z/VM may be included in end-to-end scheduling scenarios.

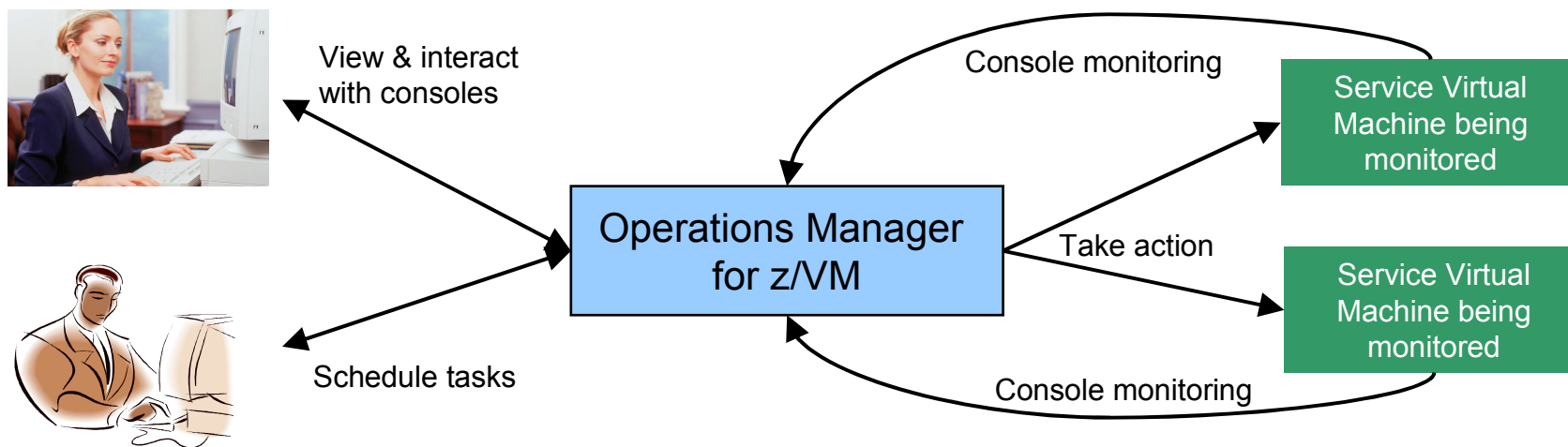
Introducing Operations Manager for z/VM

Increase productivity

- Authorized users view and interact with monitored virtual machines without logging onto them
- Multiple users view/interact with a virtual machine simultaneously

Improve system availability

- Monitor virtual machines and processes
- Take automated actions based on console messages
- Reduce problems due to operator error



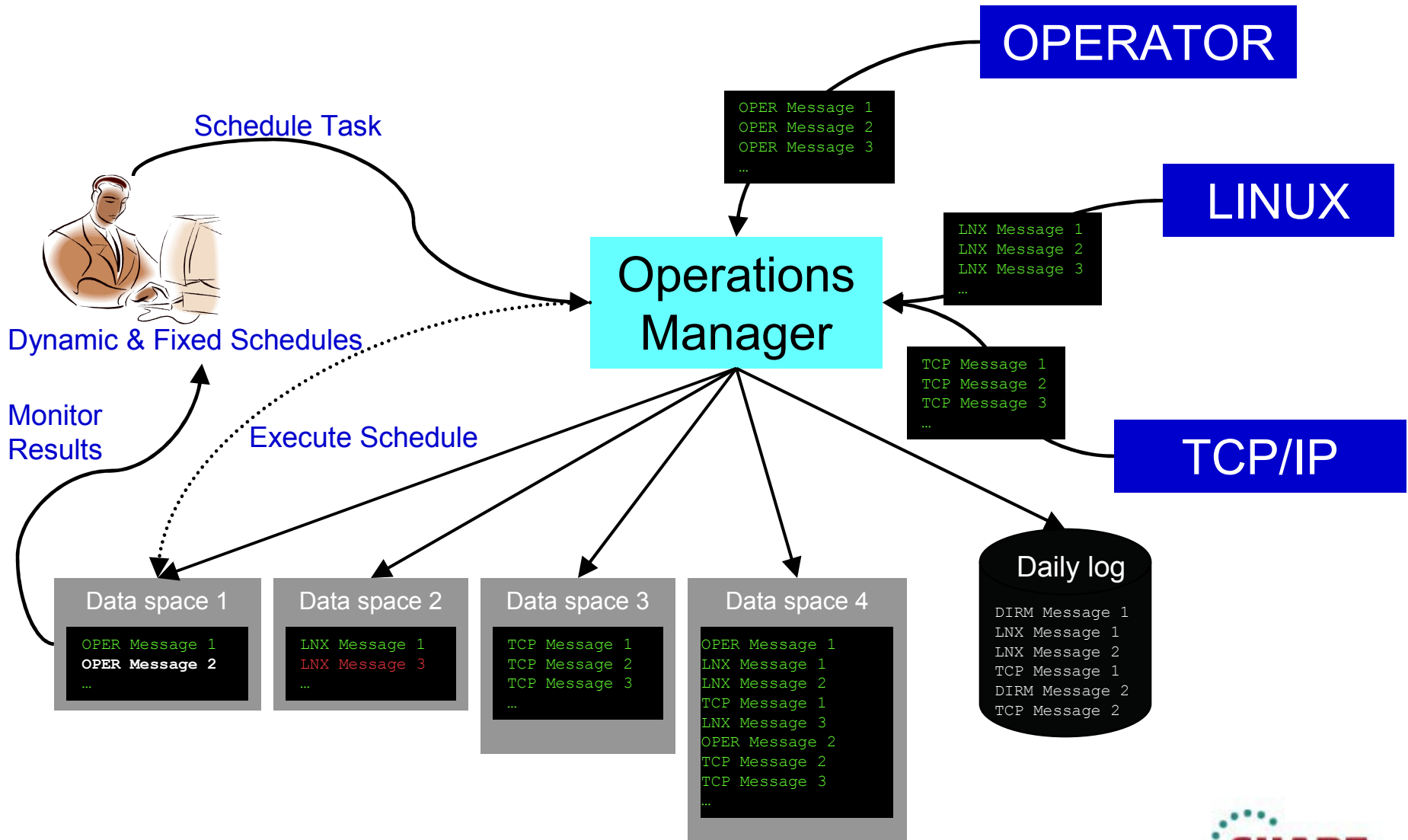
Automation

- Routine activities done more effectively with minimal operations staff
- Schedule tasks to occur on a regular basis

Integration

Fulfill take action requests from OMEGAMON XE on z/VM and Linux

Scheduling z/VM workloads



Scheduling Features of Operations Manager for z/VM

- **DEFSCHD**

- DEFSCHD NAME(*schdid*) ,ACTION(*actnid*) ,
 - WHEN(*mm*)
 - ,WHEN(*hh:mm*)
 - ,EVERY(*hh:mm*)
 - ,WHEN(*DOW-hh:mm*)
 - ,WHEN(*DD-hh:mm*)
 - ,WHEN(*MM/DD-hh:mm*)
 - ,WHEN(*NOW+hh:mm*)
 - ,WHEN(*YYYY/MM/DD-hh:mm*)

DEFSCHD Parameters

- **NAME(schdid)**
 - specifies the name of the schedule.
- **ACTION(actnid)**
 - specifies the name of the action to associate with the schedule.

DEFSCHD Parameters cont...

- **WHEN(mm)**
 - specifies the execution time is hourly. The action is scheduled at mm minutes of the hour.
- **WHEN(hh:mm)**
 - specifies the execution time is daily. The action is scheduled at hh:mm of the day. For example, 01:30 refers to 1:30 A.M., and 13:30 refers to 1:30 P.M.
- **EVERY(hh:mm)**
 - specifies the execution time is based on an interval. The interval is every hh:mm of the day starting with Operations Manager startup.

DEFSCHD Parameters cont...

- **WHEN(DOW-hh:mm)**
 - specifies the execution time is weekly. The action is scheduled on MON, TUE, WED, THU, FRI, SAT, or SUN of the week at hh:mm of the day.
- **WHEN(DD-hh:mm)**
 - specifies the execution time is monthly. The action is scheduled on DD day of the month at hh:mm of the day.

*Note: If the specified value does not occur within a month or year, the execution time is postponed until the value is reached. For example:
WHEN(31-00:00) causes the execution to occur only in months with 31 days.
WHEN(02/29-00:00) causes the execution to occur only in leap years.*

DEFSCHD Parameters cont...

- **WHEN(MM/DD-hh:mm)**
 - specifies the execution time is yearly. The action is scheduled on MM/DD of the year at hh:mm of the day.
- **WHEN(YYYY/MM/DD-hh:mm)**
 - specifies the execution time to be once on the specified date and time. The action is not scheduled if the date entered has already occurred.
- **WHEN(NOW+hh:mm)**
 - specifies the execution time to be once, after a delay of hh:mm from now. hh:mm may range from +00:00 to +23:59. If omitted, the default is +00:00.

DEFSCHD Parameters cont...

- **PARM(parm)**
 - specifies the parameter passed to the action when it is scheduled for execution. The DEFACTN command defines the parm values. The parameter is also available to any chained action routine defined in the NEXTACTN operand on DEFACTN.
- **DAYS(MTWFSSH)**
 - specifies the days of the week (Monday-Sunday) and if the schedule is active on holidays. For the schedule to be active on a day of the week and holiday, the corresponding location must contain the asterisk (*) symbol. Issue the DEFDATE command to define holidays. The default is all days and holidays (*****).

DEFSCHD Parameters cont...

- **HOURS(h,h,..h)**
 - specifies the hours of the day the schedule is active. Any value from 0 to 23 may appear in the comma separated list. Up to 24 values may be present. The default is all hours.
- **ENV(env)**
 - specifies the execution environment of the action is overridden with the *env* supplied. The DEFACTN command provides additional information on overriding the execution environment and valid values for *env*.
- **LIMIT(count,interval)**
 - specifies the execution of the schedule's action is limited to *count* times within *interval* seconds. The values for *count* and *interval* must be from 1 to 999999.

DEFSCHD Parameters cont...

- **MAXDEFER(n)**
 - specifies the number of minutes the schedule may be delayed from beginning execution due to CONFLICT. If a schedule is not able to begin before the MAXDEFER value is reached, the schedule is advanced to the next execution time. The default is 0.
- **CONFLICT**
 - specifies up to sixteen (16) conflicting schedules. If any of the specified schedules are executing, this schedule is deferred from execution. The schedule is then checked every minute until MAXDEFER is reached.

Reduce computer operator errors by automating routine tasks with timers or scheduled events and actions

- Perform routines tasks during off-peak hours, examples:
 - Log Maintenance
 - Backup routines
 - Manage guest startup and shutdown dependencies
- Execute tasks based on:
 - Calendar requirements (holiday, workday)
 - Interval Processing
 - Event Triggers
 - Dependency on success/failure of other tasks

Operations Manager for z/VM will be used to demonstrate z/VM scheduling functionality. Other vendors provide similar solutions.

Schedule SFPURGER for execution

Purge any files of class Z

```
DEFSCHD NAME (SPLMGMT) , +
```

```
WHEN (11:00) , +
```

```
DAYS (MTWT*SSH) , +
```

```
ACTION (SFPURGER)
```

*

```
DEFACTN NAME (SFPURGER) , +
```

```
COMMAND (EXEC SFPURGER FORCE) , +
```

```
OUTPUT (LOG) , +
```

```
ENV (LVM)
```

SFPURGER OPTIONS File

Action in Operations Manager to call z/VM's SFPURGER EXEC

*

SFPURGER OPTIONS file

* Send console log to user ID TSTADMN1 at demo node

CONSOLE TSTADMN1 DEM1ZVM

* Erase LOG and RUN files that are more than 3 days old

KEEPDAY 21

* Set prime shift start and end times

PRIMSHFT 07:30:00 16:30:00

* Use defaults for the following:

* MSGTYPE SORTMOD SFPCNTL SOSCNTL SFPMOD APPEND

SFPCNTL SFPTRACY

SFPTRACY CONTROL

* Ignore any spool files found in the NSS queue (privilege class E)

QUEUE NSS ACTION IGNORE

*

* Purge any spool files found in class Z

CLASS Z ACTION PURGE

Make sure OPMGRM1 links and accesses MAINT 193 disk for access to SFPURGER functions

Centralizing more management functions (including scheduling) at the z/VM level may simplify management effort.



- No need to duplicate schedules on large number of Linux guest.
- Build one place, execute across many Linux guests.
- Cron schedules are often used in a looping fashion to “catch” an event.
- z/VM Tooling’s console and syslog management functions can be event driven, a much more efficient scheduling technique.
 - Startup and shutdown scenarios (*VMEVENT, shutdown complete msg)
 - Read-only root filesystem example (script versus syslog event)

Improvements in dependency resolution may be realized by interaction with z/VM & Linux.



```
* Start of STOPLNX configuration
* Watch for shutdown complete message on Linux guest
DEFRULE NAME(LNXDOWN),+
  MATCH(*HCPGIR450%*),+
  USER(OMEGLNX1),+
  ACTION(LNXBKUP)
* Turn off the rule in general
SUSPEND RULE(LNXDOWN)
*
DEFSCHD NAME(BACKUP),+
  WHEN(SAT-03:00),+
  ACTION(STOPLNX)
* Turn off the schedule in general
SUSPEND SCHEDULE(BACKUP)
*
* Start of guest backup scenario, resume rule for guest shutdown msg
DEFACTN NAME(STOPLNX),+
  COMMAND('RESUME RULE(LNXDOWN)'),+
  ENV(GOM),+
  NEXTACTN(STOPLNXA)
*
* Action to shut down Linux guest in prep for backup
DEFACTN NAME(STOPLNXA),+
  COMMAND(CP SIGNAL SHUTDOWN OMEGLNX1 WITHIN 90),+
  ENV(LVM)
*
...
```

Scheduling and event triggered automation integrate well.

- Dynamic scheduling: The ability to handle planned and unplanned events.
- While scheduling implies planning, timing, and forethought, it is unrealistic not to expect and prepare for dynamic changes to the schedule.
- z/VM automation tools with scheduling functionality allow for this dynamic functionality.
 - Schedule functions allow for planned events.
 - Automation functions trigger based on dynamic events.
 - Having both functions applied to z/VM and Linux together allows for more complete view of environment (ex logoff).

Example: *VMEVENT

```
DEFEMON NAME(ADMIN2),+  
  TYPE(1),+  
  USER(TSTADMN2),+  
  ACTION(AUTOLOG1)
```

*

```
DEFACTN NAME(AUTOLOG2),+  
  COMMAND(CP XAUTOLOG &3),+  
  OUTPUT(LOG),+  
  ENV(OPMGRS1)
```

Note: The &3 substitute variable is the USER ID which caused the event.

Tasks specific to z/VM may be included in end-to-end scheduling scenarios.



Full backup of Business Application and database environment.

- A Linux on System z guest hosts an application that uses a DB2 on z/OS database.
- The Linux guest is scheduled for a full backup.
- By integrating TWS E2E with Operations Manager for z/VM, it is possible to define scheduled dependent operations on z/OS, Linux on System z, and z/VM.

Operational Tasks and Dependencies

- Stop Application on Linux guest (z/VM or Linux)
- Stop DB2 on z/OS (z/OS)
- Shutdown the Linux guest (z/VM)
- Backup the Linux guest (z/VM)
- Backup DB2 (z/OS)
- Start DB2 (z/OS)
- Start Linux (z/VM)
- Start Application (z/VM or Linux)

How to Integrate TWS E2E with Operations Manager for z/VM

- RSCS
- Use FTPS to send file to CMS guest reader. With IMSG on, a console message will be generated and triggering a rule in Operations Manager.
- Use FTPS to send JCL from Operations Manager for z/VM to JES to issue SRSTAT commands to TWS E2E.
- Interface with via Linux guest to TWS E2E environment (touch file).

Note: Once the FTPS, RSCS, etc task is complete, each product functions on its own.

Operations Manager to TWS E2E

- FTPS JCL

There is a feature of FTPS on z/OS that allows JCL to be submitted to JES. Operations Manager can issue an exec that FTPs JCL to JES. This JCL will issue an SRSTAT command to TWS E2E to set the status of a Special Resource. This Special Resource is used to indicate to TWS E2E when the z/VM tasks are completed.

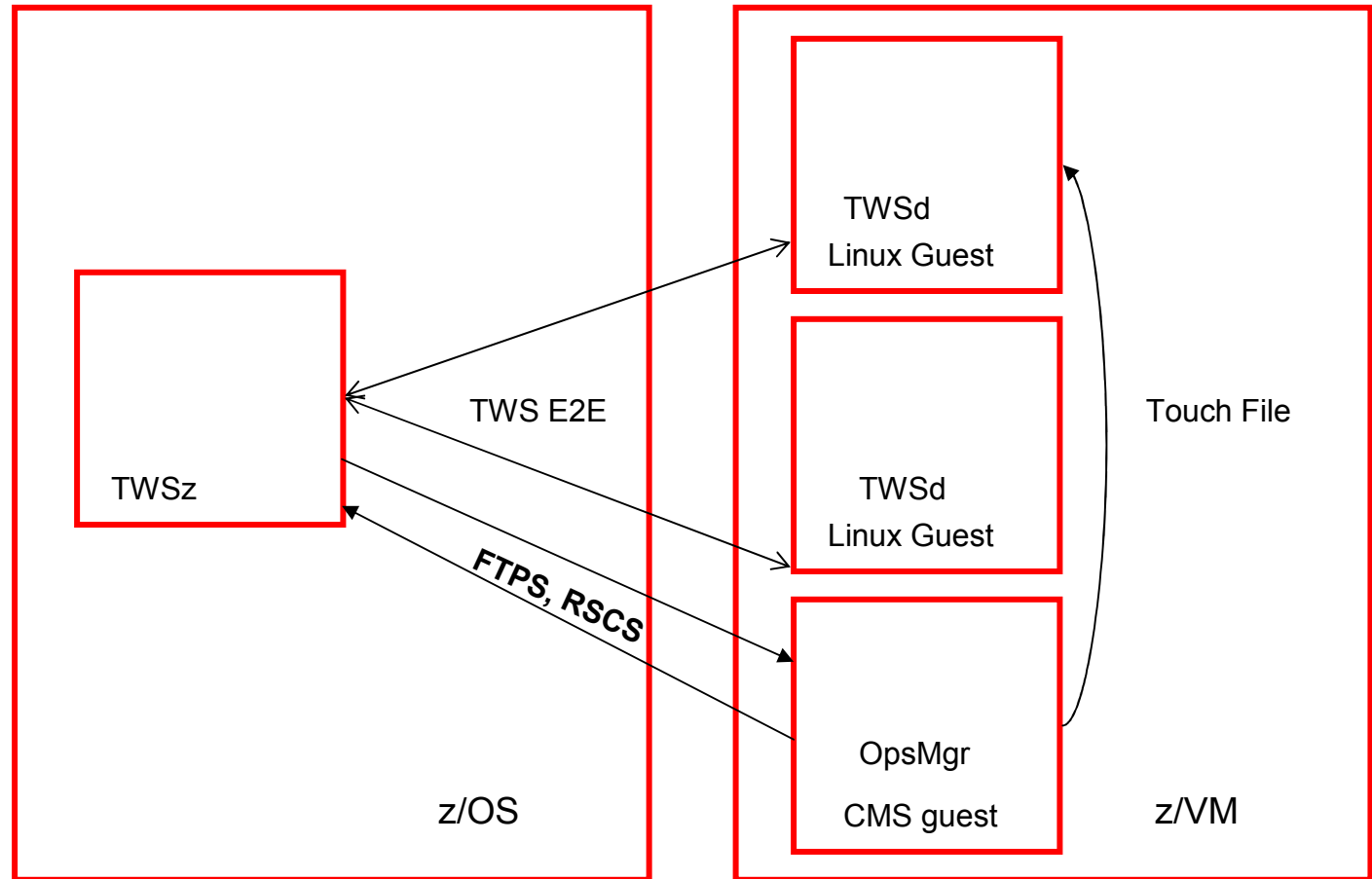
Interface Points



TWSz requests Operations Manager to exec a z/VM task.

Operations Manager provides Direct feedback to TWSz on Success or failure of the z/VM Task execution (TWS SRSTAT)

TWSz manages dependencies across z/OS, z/VM, and Linux environments.



Note: With these interface points using features of TWS E2E and Operations Manager for z/VM it is possible to include z/VM tasks in the scheduling environment.

TWS Schedule



```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
----- OPERATIONS ----- Row 1 to 8 of 8
Command ==> _ Scroll ==> PAGE

Enter/Change data in the rows, and/or enter any of the following
row commands:
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete
S - Select operation details, J - Edit JCL
Enter the TEXT command above to include operation text and number of conditions
in this list, or, enter the GRAPH command to view the list graphically.

Application          :  BACKUPWZVM          backup with z/VM

Row  cmd  Oper  Duration  Job name  Internal predecessors  Morepreds  No.of
ws  no.  HH.MM.SS  _____  _____  -IntExt-  Conds
'''  NONR  010  00.00.01  HEAD_____  _____  0  0  0
'''  L117  020  00.00.01  STOPAPPL  010  _____  0  0  0
'''  CPU1  030  00.01.00  STOPDB2_  020  _____  0  0  0
'''  CPU1  040  00.00.03  OPSMGRVM  030  _____  0  0  0
'''  CPU1  050  00.01.00  BCKUPDB2  030  _____  0  0  0
'''  CPU1  060  00.01.00  STARTDB2  050  _____  0  0  0
'''  L117  070  00.01.00  STRLAPPL  060  _____  0  0  0
'''  NONR  080  00.00.01  TAIL_____  070  _____  0  0  0
***** Bottom of data *****

MA a 02/015
Connected to remote server/host 9.82.56.124 using lu/pool HFDIP006 and port 23
    
```

OPSMGRVM: Potential integration points

- RSCS is a z/VM networking program.
 - It extends the scope of a single system to an entire network of computers and devices.
 - RSCS transfers data (as spool files) between its local system and remote devices and printers or other systems.
 - Through RSCS, users can send and receive messages, files, commands, and print and system jobs within their network.
- OPSMGRVM will execute on z/OS and either be an FTPS task.
 - **FTPS** (also known as *FTP Secure* and *FTP-SSL*) is an extension to the commonly used [File Transfer Protocol](#) (FTP) that adds support for the [Transport Layer Security](#) (TLS) and the [Secure Sockets Layer](#) (SSL) cryptographic protocols.
 - Send file to a CMS guest reader monitored by Operations Manager.
 - Monitor for console message (require IMSG ON)
- POSTZMSG
 - The **postzmsg** command sends an event to an event server.
- TWS E2E has file trigger on Linux FTA (fault tolerant agent).
 - Operations Manager can touch a file on the Linux guest with FTA
 - FTA responds to file trigger

Note: We will assume a message is sent to a monitored CMS guest to trigger z/VM Task:
MSG TWS READY FOR RES SCHED BU117S TO BEGIN.

Operations Manager

- DEFRULE NAME(BU117R),+
MATCH(*RES SCHED BU117S*),+
ACTION(RES117A), USER(OPMGRC1)
- DEFACTN NAME(RES117A),+
COMMAND(CP SIGNAL SHUTDOWN HASL117 WITHIN 90),+
ENV(LVM), NEXTACTN(LNXBKUPB)
- DEFACTN NAME(LNXBKUPB),+
COMMAND(CP MSG BKRBKUP SUBMIT HASL117), ENV(LVM)

Operation Manager cont...

- DEFRULE NAME(TWSSRR),+
MATCH(*BACKUP*SLESA117*COMPLETE*),+
ACTION(TWSSRA),+
USER(OPMGRC1)
*
- DEFACTN NAME(TWSSRA),+
COMMAND(EXEC **TWSSR**),+
OUTPUT(LOG),+
ENV(LVM)

TWSSR FTP JCL

```
/* REXX */
/* FTP JCL to ATSzOS1 to set SR VMTASKDONE to Y */
trace r
address command
Address CMS
/* parse arg ":" msgtext */
/* msgtext2 = ""msgtext "" */
/* FTP */
Queue 'sine xxxxxxxx' /* you can use anonymous */
Queue 'quote site filetype=jes'
Queue 'put sinev.jcl'
Queue 'quit'
'FTPS 9.82.56.124 (EXIT'
```

Note: FTPS syntax not complete, used as an example.

FTPS JCL

```
//SINEV JOB (0),'DS TRIG SRSTAT',CLASS=A,  
//      MSGCLASS=O,MSGLEVEL=(1,1),REGION=4096K  
//STEP1 EXEC PGM=EQQEVPGM  
//STEPLIB DD DISP=SHR,DSN=TWS.V8R5M1.SEQQLMD0  
//EQQMLIB DD DISP=SHR,DSN=TWS.V8R5M1.SEQQMSG0  
//EQQMLOG DD SYSOUT=*  
//SYSIN  DD *  
SRSTAT 'VMTASKDONE' SUBSYS(TWT8) AVAIL(YES)  
  
/*
```

Note: Use EQQEVPGM to issue the TSO commands for Tivoli Workload Scheduler for z/OS in batch.

TWS Schedule



```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
----- OPERATIONS ----- Row 1 to 8 of 8
Command ==> _ Scroll ==> PAGE

Enter/Change data in the rows, and/or enter any of the following
row commands:
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete
S - Select operation details, J - Edit JCL
Enter the TEXT command above to include operation text and number of conditions
in this list, or, enter the GRAPH command to view the list graphically.

Application          : BACKUPWZVM          backup with z/VM

Row  cmd  Oper  Duration  Job name  Internal predecessors  Morepreds  No. of
ws  no.  HH.MM.SS  _____  _____  -IntExt-  Conds
'''  NONR  010  00.00.01  HEAD_____  _____  0  0  0
'''  L117  020  00.00.01  STOPAPPL  010  _____  0  0  0
'''  CPU1  030  00.01.00  STOPDB2_  020  _____  0  0  0
'''  CPU1  040  00.00.03  OPSMGRVM  030  _____  0  0  0
'''  CPU1  050  00.01.00  BCKUPDB2  030  _____  0  0  0
'''  CPU1  060  00.01.00  STARTDB2  050  _____  0  0  0
'''  L117  070  00.01.00  STRLAPPL  060  _____  0  0  0
'''  NONR  080  00.00.01  TAIL_____  070  _____  0  0  0
***** Bottom of data *****

MA a 02/015
Connected to remote server/host 9.82.56.124 using lu/pool HFDIP006 and port 23
    
```



STRTAPPL SR VMTASKDONE



```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
----- SPECIAL RESOURCES ----- Row 1 to 1 of 1
Command ==> _                      Scroll ==> PAGE

Enter/Change data in the rows, and/or enter any of the following
row commands:
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete

Operation          : L117 070          start linux application

Row  Special                               Qty  Shr  Keep On  Avail on
cmd  Resource                               Ex   Error Complete
' ' ' VMTASKDONE                           1    S    _      _
***** Bottom of data *****

```

MA a 02/015

Connected to remote server/host 9.82.56.124 using lu/pool HFDIP006 and port 23

Operations Manager for z/VM as the Master Scheduler

Why Operations Manager as the Master Scheduler?

- z/VM may own the backup schedule for all Linux guest systems.
- Startup and Shutdown procedures may be owned by z/VM and be the trigger for other schedules.
- Other reasons including politics...

Operations Manager as Master Scheduler Scenario

- The same scenario as before, however, Operations Manager will trigger the scheduled tasks.

Operations Manager Schedule

- DEFSCHD NAME(LNXBKUP), ACTION(BKUPLNX), WHEN(FRI-02:00)
- DEFACTN NAME(BKUPLNX), COMMAND(STOPAPPL), NEXTACTN(BKUPLN2)
- DEFACTN NAME(BKUPLN2), COMMAND(**TWSSR2**), NEXTACTN(BKUPLN3)
- DEFACTN NAME(BKUPLN3), COMMAND(STOPLNX), NEXTACTN(BKUPLN4)
- DEFACTN NAME(BKUPLN4), COMMAND(BACKLNX), NEXTACTN(BKUPLN5)
- DEFACTN NAME(BKUPLN5), COMMAND(STRTLNX)

FTPS JCL

```
//SINET JOB (0),'DS TRIG SRSTAT',CLASS=A,  
//      MSGCLASS=O,MSGLEVEL=(1,1),REGION=4096K  
//STEP1  EXEC PGM=EQQEVPGM  
//STEPLIB DD DISP=SHR,DSN=TWS.V8R5M1.SEQQLMD0  
//EQQMLIB DD DISP=SHR,DSN=TWS.V8R5M1.SEQQMSG0  
//EQQMLOG DD SYSOUT=*  
//SYSIN  DD *  
SRSTAT 'DB2BKUP' SUBSYS(TWT8) AVAIL(YES)  
  
/*
```

Note: Use EQQEVPGM to issue the TSO commands for Tivoli Workload Scheduler for z/OS in batch.

Other potential Interface Points Beyond FTPS JCL

- TWS E2E issue console messages on Linux to trigger action in Operations Manager.
- Event Trigger tracking: Operations Manager can create a file on z/OS via FTP versus FTP JCL. TWSz triggers on file creation with ETT.
- SYSLOG messages on Linux. Operations Manager can read Unix/Linux SYSLOG messages, including remote systems, in the same way it monitors console messages.
- RSCS connection between z/VM and z/OS.

References

- FTP JCL: http://www.ehow.com/how_5213702_ftp-jcl-mainframe.html
- REXEC: <http://www.mkssoftware.com/docs/man1/rexec.1.asp>
- Operations Manager for z/VM
<http://www.ibm.com/software/sysmgmt/zvm/operations>
- Automatically Logging on a User at Linux System Boot time for Console Management
<http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101634>
- Routing Linux SYSLOG data to IBM Operations Manager for z/VM
<http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101379>

Questions?



E
101

धन्यवाद

Hindi

多謝

Traditional Chinese

감사합니다

Korean

Спасибо

Russian

Gracias

Spanish

شكراً

Arabic

Thank You

English

Obrigado

Brazilian Portuguese

Grazie

Italian

Danke

German

多谢

Simplified Chinese

Merci

French

நன்றி

Tamil

ありがとうございました

Japanese

ขอบคุณ

Thai