**JURASSIC PARK TEST PROCEDURE**

IFM Chassis Console in building 4: 172.25.27.2 (pw: lab)

IFM1/2: 2011/2010

Blade1: 2004

Blade2: 2013

Blade3: 2017

Blade4: 2012

Blade5-7: 2014-2016

Blade 8: 2007

APC info:

**IP:172.25.27.4 (admin/raritan)**

**Ports are 5/10/15**

**IP:172.25.27.5 (admin/hwenglab)**

**Ports are 9/19/20**

Chamber IP: 172.25.27.25 (hwenglab, 049 (next level))

100G traffic python script is running on Ubuntu OS with pkgen from the host. Thus, M.2 boot on the blade is Ubuntu OS.

**On the IFM:**

You might need to do these extra steps before doing I/

If the board just boots up to ‘loader>’, you need to manually boot to OS:

* Find out where the ‘bzImage’ file is by listing all the file directories under ‘loader>’

loader> ***dir***

* To boot up to OS: ‘loader> ***ifm\_boot <directory>:bzImage’***

For instance, ‘loader> ***ifm\_boot sys1:bzImage’***

Or: ‘loader> ***ifm\_boot bootflash:bzImage’***

Note: Please check with DE for which bzImage\* to execute if there are more than one bzImage\*. Eg: recently on P2 or PP2 board, it’s bzImage-MR-diags

--- Log in: root/cmc

After booting to OS, you need to check if there are files (bzImge and \*.sh like diagpkg.sh, cmcapppkg.sh,…) under /flash or /bootflash. If you don’t see any files in either directory and /bootflash doesn’t exist, create /bootflash:

***mkdir /bootflash***

***mount /dev/sda4 /bootflash***

--- Get full OS support for the test after logging in IFM:

***cd /bootflash*** (or cd /flash if the files resides here instead)

***./cmcapppkg.sh***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

***Traffic Test And Concurrent Tests Running On The Host (Blade) Side:***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**I/ On the blade: Power and get the blade online.**

1/ blade-power on

2/ /etc/scripts/getip\_novlan.sh 0 // obtain the IP for the blade

3/ ifconfig eth0 // display the IP as shown in blue – it will be needed in the next step

Example:

*Thu Jan 01 01:17:12 IBMC-SLOT[] # :DEVONLY\_ENV$ ifconfig eth0*

*eth0 Link encap:Ethernet HWaddr 3E:57:31:E1:16:E9*

*inet addr:****10.18.178.246*** *Bcast:10.18.178.255 Mask:255.255.255.0*

*inet6 addr: fe80::3c57:31ff:fee1:16e9/64 Scope:Link*

*UP BROADCAST RUNNING MULTICAST MTU:1496 Metric:1*

*RX packets:1510 errors:0 dropped:0 overruns:0 frame:0*

*TX packets:1152 errors:0 dropped:0 overruns:0 carrier:0*

*collisions:0 txqueuelen:1000*

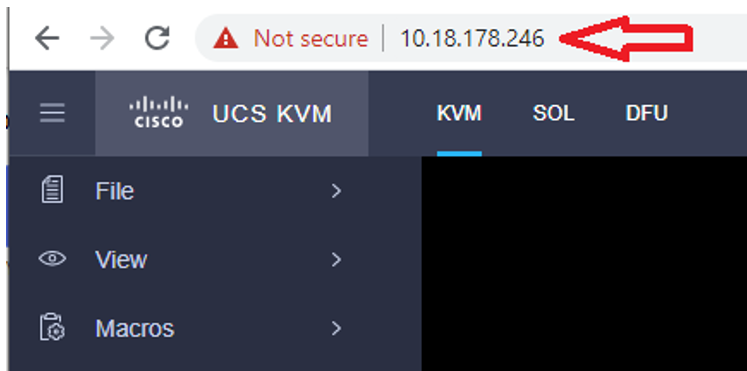
*RX bytes:131515 (128.4 KiB) TX bytes:85624 (83.6 KiB)*

*Interrupt:27*

**II/ Work with host:**

1/ Go to web browser (chrome) with the blade IP obtained from the prior step I/3.

Log in: admin/Nbv12345 except for blade8: admin21/Nbv12345



2/ Wait for booting to the Boot option on blade (if it already passed, please cycle power the blade to get back to Boot Menu option). Press Fn+F6 to select Boot Menu to break into ‘Boot Menu’.

Text

Description automatically generated

3/ Select ‘Ubuntu’ from the boot device, then press ‘Enter’ keyboard:

Graphical user interface, application

Description automatically generated

Blade should boot up to Ubuntu OS.

OS login pw: root/Free4All

Graphical user interface, text, application

Description automatically generated

Upon logging in OS, if you’d like to get more terminals (without having the need to run multiple jobs in background, you can get as many as you want with this GUI ‘Alt-F1’, ‘Alt-F2’,…:

A screenshot of a computer

Description automatically generated

**III/ Go back to the IFM, run the following commands on both IFMs:**

1/ Issue the followings to get to DIAG mode:

Note: Please check with DE for getting an appropriate DIAG image for the test because it’s in early development and the test support has not been in the release DIAG image.

FYI. Currently, the latest DIAG image supporting FC is at (but it’s being changed time to time):

*The following support FC32 traffic with QSFP1 and QSFP2 connected back to back.*

*savbu-cmc2-lnx:/ws/datton-sjc/image/madison/fc\_snake/fc\_eth\_snake/11012023 > md5sum \*.sh*

*b67bb4e6282577a0d01c824dc1d86af5  cmcapppkg.sh*

*22d36a7911fb486fde47b06d570643ab  debugpkg.sh*

*ab0170efbc40fbf099a25a5097b06248  diagpkg.sh*

oot@ifm:~# mount /dev/sda4 /bootflash/

cd /bootflash

sh cmcapppkg.sh

sh diagpkg.sh

cd /nuova/diagnostics/

***./diagpkg.sh***

***cd /nuova/diagnostics***

***./mfgdiag board=Madison***

4/ Manually run on the system FAN to 100%

***fan tray set 100 0***

***fan tray set 100 1***

***fan tray set 100 2***

***fan tray set 100 3***

***fan ifm set 100***

***fan pifm set 100***

2/ If you want to set voltage margin on the IFM, issue the following at DIAG prompt:

* Margin low: ‘voltage margin all low’
* Margin high: ‘voltage margin all high’

**########################################################################**

**###### Instructions to run 100G FC-Eth Cable Traffic**

**######**

**######** NIF FC cable connection:

**######** Port 1 and 2: 128G QSFP optical cable are connected between these 2 ports.

**######** Current support and plan of record are having only port 1 and 2 with FC.

**######** Port 2-8: QSFP self-loop back modules.

**########################################################################**

At DIAG prompt, do this:

1/ Type ‘ctrl-z’ at DIAG prompt to edit the file: MADISON> ***ctrl-z***

2/ ***vi /asic/verif/cosim/sundown\_lib/tests/diag\_sun\_fc\_eth\_snake\_check.tcl***

3/ Modify (near the end of the file):

From:

*if {1} {*

*sleep  10*

*#pokemon*

To:

***log\_stop***

***if {0} {***

*sleep  10*

*#pokemon*

Go back to DIAG prompt: fg

Run these commands at DIAG prompt:

sp glob USE\_SERDES\_C\_API=1

sp glob LinkFlapTestLoop=1

sp glob Retry=19

**sp iomsys snakefc32eth HIFport=0-3,5,6,8,9**

sp iomsys snakefc32eth Sleep=5

sp iomsys snakefc32eth qsfp1\_2connect=1

show iomsys snakefc32eth

run iomsys snakefc32eth

**card\_port\_status**

**Note:**

A/ **sp iomsys snakefc32eth HIFport=0-3,5,6,8,9**

This command is to configure the HIF ports on the blades. The parameters in the command is for full loaded 8 blade chassis. You need to modify them to specify the blades to run the traffic in the chassis if it has less than 8 blades based on the Blade-MAC table below.

B/ **card\_port\_status**

This command lists link status (just link up, not full vnic. Vnic status needs to check at VIC prompt) on all HIFs and NIFs. You can check the link status of the HIFs on each blade and the NIF on each port based on the Blade-MAC table below.

C/ Check for successful vnic configuration:

Get to the adapter VIC prompt: i2c\_uart -b1 -t 0x9a (to get back to BMC prompt, press ctrl-p d), then issue the command: dbgcli -s macd -e bev-ports

For example,

*# dbgcli -s macd -e bev-ports*

*Connecting to 127.0.0.1, port 41204...*

*Connected.*

*---- -- ------------ ----- --------- ---------- ----------------------- --------*

*port ch  serdes id   admin   speed     an\_mode            fec\_mode        state*

*state  cfg oper  cfg  oper        cfg        oper*

*---- -- ------------ ----- ---- ---- ---- ----- ----------- ----------- --------*

*0  0* ***12,13,14,15     E AUTO 100G AUTO   On        cl91        cl91  LINKUP***

*1  4* ***16,17,18,19     E AUTO 100G AUTO   On        cl91        cl91  LINKUP***

*2  1                  E AUTO AUTO AUTO   On        cl91        cl91  INIT*

*3  5                  E AUTO AUTO AUTO   On        cl91        cl91  INIT*

*4  0           11     E AUTO   1G  Off  Off         Off         Off  LINKUP*

*5  1                  D AUTO   1G  Off  Off         Off         Off  INIT*

*---- -- ------------ ----- ---- ---- ---- ----- ----------- ----------- --------*

*#*

D/ Blade-MAC map:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Blade | Mac | QSFP | Mac |  |
| 1 | 9 | 1 | 17 |  |
| 2 | 8 | 2 | 14 |  |
| 3 | 6 | 3 | 16 |  |
| 4 | 5 | 4 | 15 |  |
| 5 | 3 | 5 | 11 |  |
| 6 | 2 | 6 | 12 |  |
| 7 | 1 | 7 | 10 |  |
| 8 | 0 | 8 | 13 |  |  |

**4/ Run traffic on the blade in the background (will check for traffic at VIC prompt)**

i/ Run one loop to check for the setting:

***cd /home/emi\_tests/networkdiag-2.0.6***

***./vic\_traffic\_test.py -H -s***

If you see “passed” at the end of the test. Proceed to the next step

Text

Description automatically generated

ii/ Run the test with infinite option -I (capital of letter i):

***./vic\_traffic\_test.py -H -s -I***

If you’d like to run in the background and save the output,

./vic\_traffic\_test.py -H -s -I > <logfile\_name.logs> &

For example: ./vic\_traffic\_test.py -H -s -I > temp\_traffic1.logs &



**///////////////////////////////////////////////////////////////////////////////////////////**

**VI/ Check for the errors and bandwidth:**

1/ Look at the error counter printing during test on KVM host screen

2/ Go to the adapter to check for counters:

a/ Go to BMC console

b/ Get to the adapter prompt: i2c\_uart -b1 -t 0x9a (to get back to BMC prompt, press ctrl-p d)

c/ Read the counters (data transferred and errors):

dbgcli -s macd -e macstats 0

dbgcli -s macd -e macstats 1

Example:

*# dbgcli -s macd -e macstats 0*

*Connecting to 127.0.0.1, port 41204...*

*Connected.*

*DELTA TOTAL DESCRIPTION*

*3708179 6295204048 Frames Transmitted O.K*

*3708181 6295204050 Frames Transmitted All (Good/Bad Frames)*

*29695097432 50411992618560 Octets Transmitted with out error*

*29695097432 50411992618560 Octets Transmitted Total (Good/Error)*

*3708179 6295203870 Frames Transmitted Unicast*

*0 176 Frames Transmitted Multicast*

*0 4 Frames Transmitted Broadcast*

*3708178 6295203892 Frames Transmitted VLAN.*

*0 4 Frames Transmitted Length=64.*

*0 16 Frames Transmitted Length=65-127.*

*0 160 Frames Transmitted Length=128-255.*

*3708179 6295203875 Frames Transmitted Length=4096-8191.*

*3708178 6295204049 Frames Received O.K*

*29695089424 50411992690432 Octets Received in Good Frames.*

*3708178 6295204049 Frames Received All (Good/Bad Frames)*

*29695089424 50411992690432 Octets Received (Good/Bad Frames)*

*3708178 6295203881 Frames Received with Unicast Addresses.*

*0 168 Frames Received with Multicast Addresses.*

*0 2 Frames Received with Broadcast Addresses.*

*0 2 Frames Received Length=64*

*0 8 Frames Received Length=65-127*

*0 160 Frames Received Length=128-255*

*3708179 6295203888 Frames Received Length=4096-8191*

*89.377Gbps Tx Rate*

*89.377Gbps Rx Rate*

*#*

3/ On the IFM, to check errors at Sundown, issue the command at DIAG prompt to list all RX/TX transferred and errors if any:

***sun\_top\_mac\_dump\_mibs [get\_mrl\_id {sun[0]}]***

**IV/ Concurrent tests running in the background on the blade:**

1/ Go back to the KVM host (browser) and run jobs in the background:

PTU test

Memory test

Traffic test

Notes: Run these tests in background (the ‘&’ is important)

**A/ PTU test on the blade concurrently:**

***cd /home/emi\_test/saved\_traffic\_fi/scripts/latest ptu***

If you want to run in the background and save the outputs,

./ptu -ct 1 > <log\_file\_name> &

(for example: ./ptu -ct 1 -mon > temp\_ptu &)

You can view the <log\_file\_name> to check for errors or anything else)

**B/ Memory test on the blade concurrently:**

***cd /home/emi\_test/saved\_traffic\_fi/scripts/pmem/sysdiag/sysdiag-2.0.4***

***./pmem2run -m 0 -s 0 -l 0*** (lower case of letter ‘L’)

If you want to run in the background and save the outputs,

./pmem2run -m 0 -s 0 -l 0 > <logfile\_name> &

(for example: ./pmem2run -m 0 -s 0 -l 0 > temp\_pmem & )

Please note: -l (lower case L); don’t forget ‘&’ to run in the background.



**V/ Concurrent Tests Running On The IFM In The Background**

Get full OS support for the test after logging in IFM:

***./cmcapppkg.sh***

**1/ Noe\_valley test in OS with fio:**

/dev/sda1 or /dev/sda2 is found to be ‘safe’ partition of Noe\_valley to run fio test as it’s just for logs. However, it might be changed during the development. Please check the purpose of the content to be okay for overwriting before testing.

mount /dev/sda1 /mnt

***cd /flash/fio or /bootflash/fio***

***./fio --name=randrw --iodepth=1 --rw=randrw --bs=1m --size=10m --numjobs=2 --group\_reporting --time\_based --rwmixread=70 --do\_verify=1 --verify=md5 --verify\_fatal=1 --directory=/mnt/ --runtime=10000 > temp\_m2.log &***

**Note: The feeding parameters in pink in the command need be modified to be suitable for:**

* **The directory (disk) to be tested.**
* **Duration of the test to be run.**
* **The need to run the test in the background (&) and save the test outputs so that there are other tests can run concurrently; otherwise ‘> temp\_m2.log &’ is omitted.**

2/ IFM PTU: After booting to OS,

***cd /bootflash/denverton\_ptu\_linux\_v1.1 or /bootflash/denverton\_ptu\_linux\_v1.1***

***./uServerx64\_gen -P100 &*** (to stress CPU cores – functional test should choose this for stress)

*Or,*

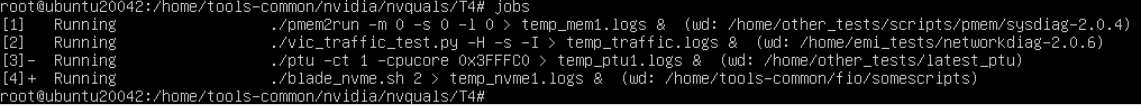
***./uServerx64\_gen -T &*** (to stress on thermal – mechanical thermal test should choose this for stress)

If you want to monitor to see the instant CPU temperatures and CPU power, you can run the monitor in the background and save the outputs in a file, for instance: temp\_ifmptu.log (unlikely do this because of its usefulness)

***./DenvertonPwrMon > temp\_ifmptu.log &***

3/ DIAG tests: The test needs to be issued after completing the configuration for traffic test in DIAG prompt. Only one DIAG can be run with the DIAG traffic test. The DIAG test must not interfere or interrupt with the traffic data flow that is SERDES, reset test,…

Note: The command ‘jobs’ would list all the jobs running in the background.



/////////////////////////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////////////////////////

**~~########################################################################~~**

**~~###### Instructions to run 100G Eth Traffic~~**

**~~###### (Previous information when FC support was not avail – it’s unlikely to run on Madison after P0)~~**

**~~########################################################################~~**

~~NIF cable connection:~~

~~100G QSFP optical/copper cables are connected between port 1-3, 2-4, 5-7, 4-8~~

~~3/ Copy and paste the following commands at the DIAG prompt (SUP10:MADISON> ) to run the configuration. There will be many messages and would take up to 4-5 minutes.~~

***~~cp /nuova/diagnostics/skagitriver\_files/roc\_fwd\_port\_pin.sr100.cfg /nuova/diagnostics/skagitriver\_files/roc\_fwd\_port\_pin.sr100.cfg.org~~***

***~~cp /nuova/diagnostics/skagitriver\_files/roc\_fwd\_port\_pin.sr-edvt.cfg /nuova/diagnostics/skagitriver\_files/roc\_fwd\_port\_pin.sr-edvt.cfg.org~~***

***~~cp /nuova/diagnostics/skagitriver\_files/roc\_fwd\_port\_pin.sr-edvt.cfg  /nuova/diagnostics/skagitriver\_files/roc\_fwd\_port\_pin.sr100.cfg~~***

***~~sp glob LinkFlapTestLoop=1~~***

***~~sp glob Retry=19~~***

***~~krphy\_init 0 slot 1 -100~~***

***~~krphy\_init 0 slot 2 -100~~***

***~~krphy\_init 0 slot 3 -100~~***

***~~krphy\_init 0 slot 4 -100~~***

***~~krphy\_init 0 slot 5 -100~~***

***~~krphy\_init 0 slot 6 -100~~***

***~~krphy\_init 0 slot 7 -100~~***

***~~krphy\_init 0 slot 8 -100~~***

~~2/ Check if the corresponding links on the blades and QSFP ports we installed are up. In this case, blade 2, 4 and QSFP port 2, 4 (and I highlighted in blue for you to easy to see) with the command:~~

~~card\_port\_status~~

~~The Port/QSFP has the following map to MAC (data path):~~

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ~~Source~~ | |  |  |  | ~~Destination~~ | | |
| ~~Blade~~ | ~~Mac~~ | ~~QSFP~~ | ~~Mac~~ |  | ~~QSFP~~ | ~~Mac~~ | ~~BLD~~ | ~~Mac~~ |
| ~~1~~ | ~~9~~ | ~~1~~ | ~~17~~ |  | ~~3~~ | ~~16~~ | ~~1~~ | ~~9~~ |
| ~~2~~ | ~~8~~ | ~~2~~ | ~~14~~ |  | ~~4~~ | ~~15~~ | ~~2~~ | ~~8~~ |
| ~~3~~ | ~~6~~ | ~~3~~ | ~~16~~ |  | ~~1~~ | ~~17~~ | ~~3~~ | ~~6~~ |
| ~~4~~ | ~~5~~ | ~~4~~ | ~~15~~ |  | ~~2~~ | ~~14~~ | ~~4~~ | ~~5~~ |
| ~~5~~ | ~~3~~ | ~~5~~ | ~~11~~ |  | ~~7~~ | ~~10~~ | ~~5~~ | ~~3~~ |
| ~~6~~ | ~~2~~ | ~~6~~ | ~~12~~ |  | ~~8~~ | ~~13~~ | ~~6~~ | ~~2~~ |
| ~~7~~ | ~~1~~ | ~~7~~ | ~~10~~ |  | ~~5~~ | ~~11~~ | ~~7~~ | ~~1~~ |
| ~~8~~ | ~~0~~ | ~~8~~ | ~~13~~ |  | ~~6~~ | ~~12~~ | ~~8~~ | ~~0~~ |

~~In words we can say~~

~~Blade 1  à QSFP 1 à QSFP 3 à Blade 1~~

~~Blade 2  à QSFP 2 à QSFP 4 à Blade 2~~

~~Blade 3  à QSFP 3 à QSFP 1 à Blade 3~~

~~Blade 4  à QSFP 3 à QSFP 2 à Blade 2~~

~~……~~

~~Cable connection:~~

|  |  |  |
| --- | --- | --- |
| ~~QSFP 1  ßà  QSFP3~~ |  | ~~QSFP 5  ßà QSFP 7~~ |
| ~~QSFP 2  ßà QSFP4~~ |  | ~~QSFP 6  ßà QSFP 8~~ |

New 2024

root@ifm:~# mount /dev/sda4 /bootflash/

mount: /dev/sda4 is already mounted or /bootflash busy

/dev/sda4 is already mounted on /bootflash

root@ifm:~# cd /bootflash/

root@ifm:/bootflash# sh cmcapppkg.sh

root@ifm:/bootflash# sh diagpkg.fc16.sh

root@ifm:/bootflash# cd /nuova/diagnostics/

root@ifm:/nuova/diagnostics# ./mfgdiag

Use this sequence:

sp glob USE\_SERDES\_C\_API=1

sp glob LinkFlapTestLoop=1

sp glob Retry=19

**sp iomsys snakefc32eth HIFport=0-3,5,6,8,9**

sp iomsys snakefc32eth qsfps\_connect\_map=3

sp iomsys snakefc32eth Sleep=10

show iomsys snakefc32eth

run iomsys snakefc32eth

cd con