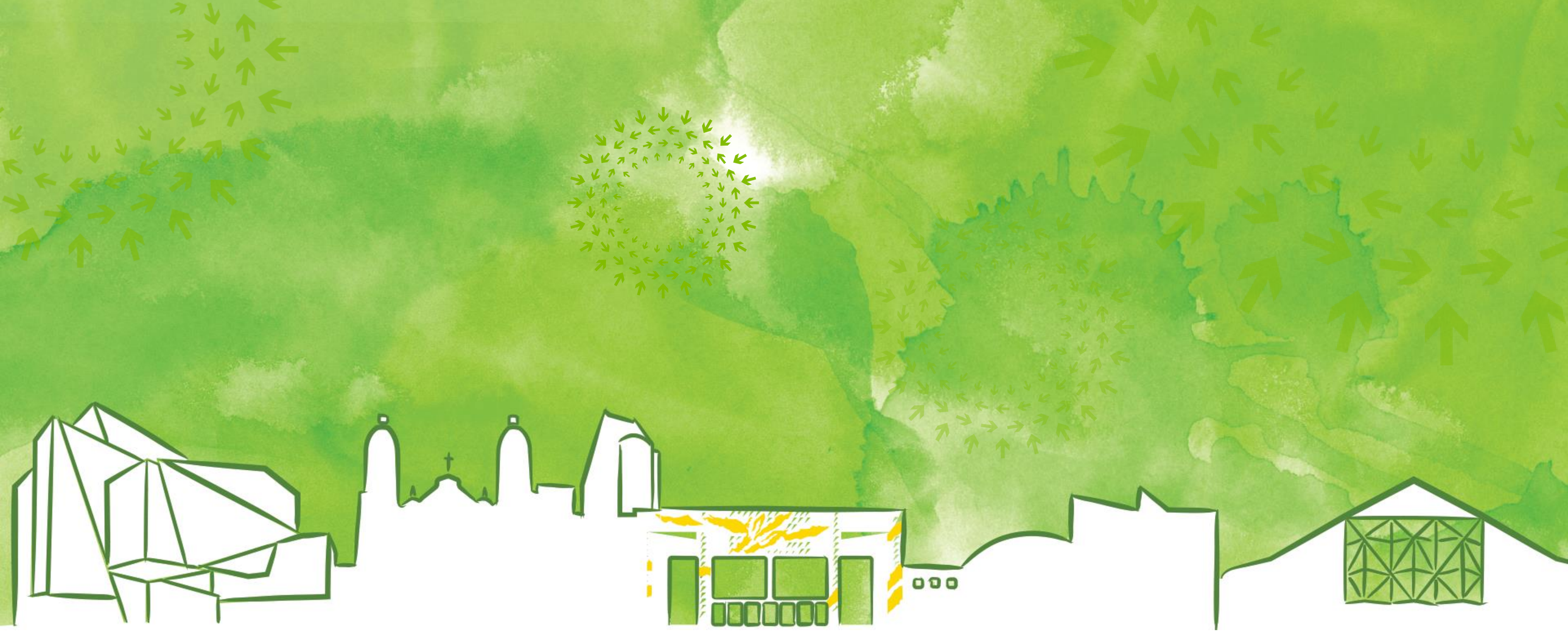


OCP Engineering Workshop **25 September 2017**| Dallas, TX

OCP Engineering Workshop – 25 September 2017 – Dallas, TX

Microsoft OpenBMC

A success story of leveraging openBMC w/ JBOD/JBOF enclosures

Ali Larijani
BMC F/W engineering manager

Microsoft OpenBMC

Agenda:

- Manageability H/W architecture in JBOD/JBOF enclosures
- OpenBMC Features for JBOD/JBOF enclosures
- OpenBMC Performance optimization
- OpenBMC Security enhancement
- OpenBMC Testing
- Summary

OPEN HARDWARE. OPEN SOFTWARE. OPEN FUTURE.

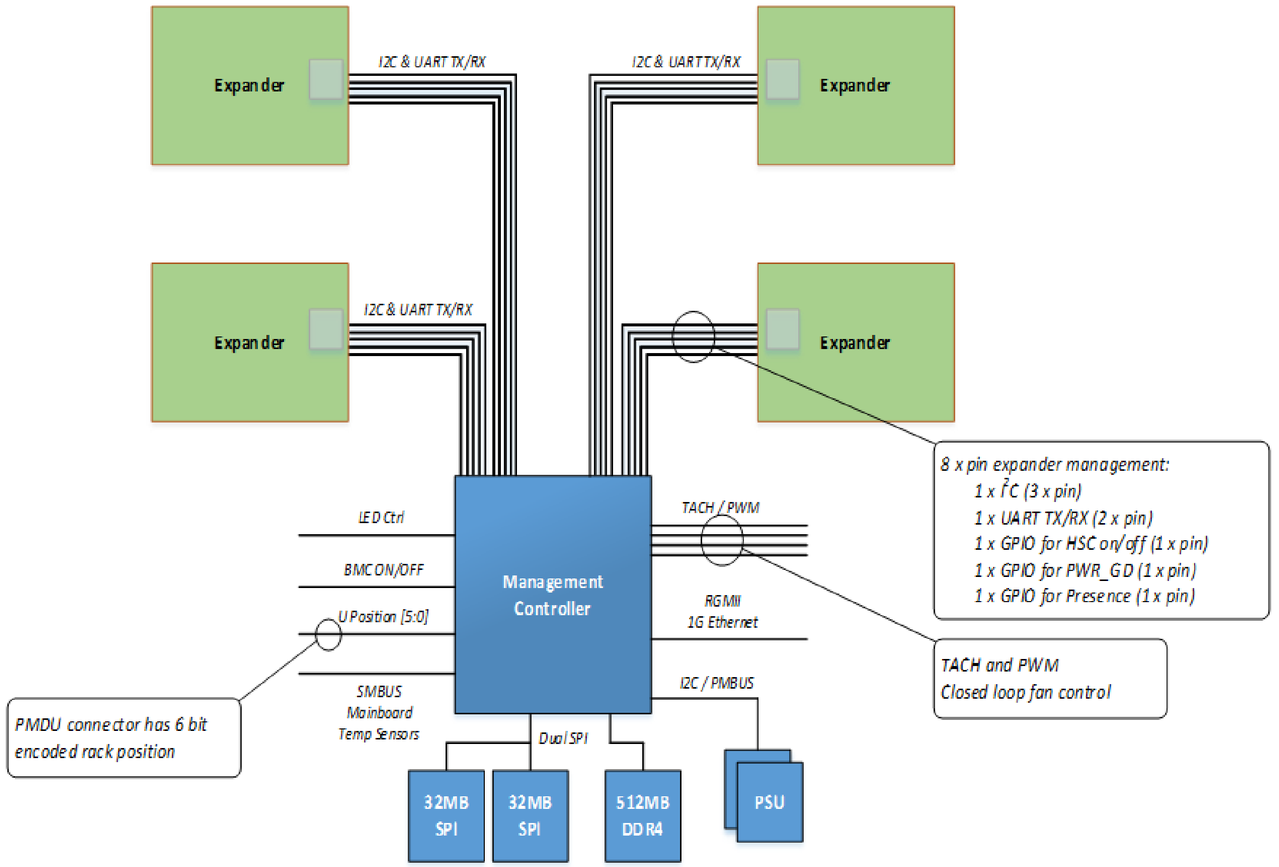
OPEN Compute Project

Microsoft OpenBMC

Manageability H/W architecture :

- BMC: AST2520
- 512MB DDR3 Memory
- x2 32MB SPI Nor Flash
- X1 1G Ethernet Interface
- Multiple I/O expanders (SAS,PCIe Switch,…)
- X2 redundant PSUs
- X12 FANs
- Upto x200 Sensors (Analog and Discrete)

Figure 1 Baseboard Management Controller (BMC)

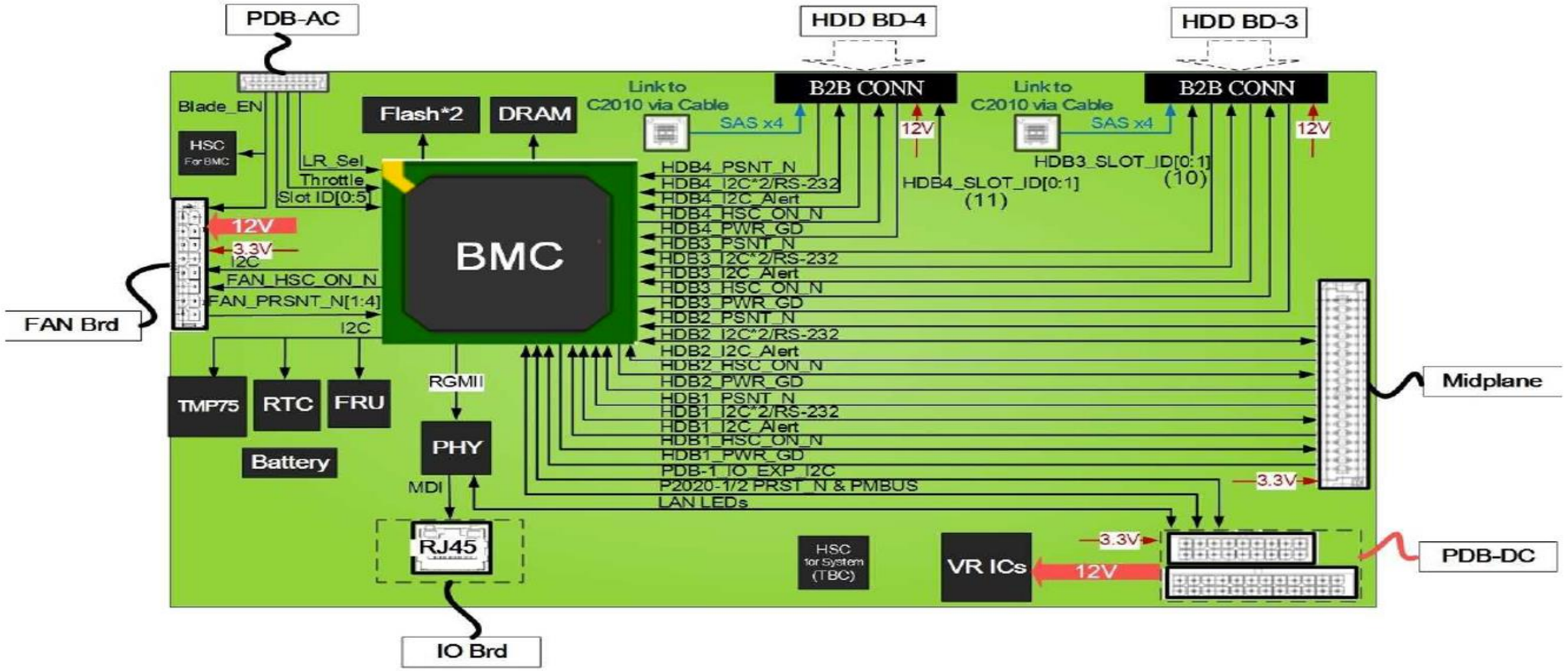


OPEN HARDWARE. OPEN SOFTWARE. OPEN FUTURE.

OPEN Compute Project

Microsoft OpenBMC

Manageability H/W architecture :



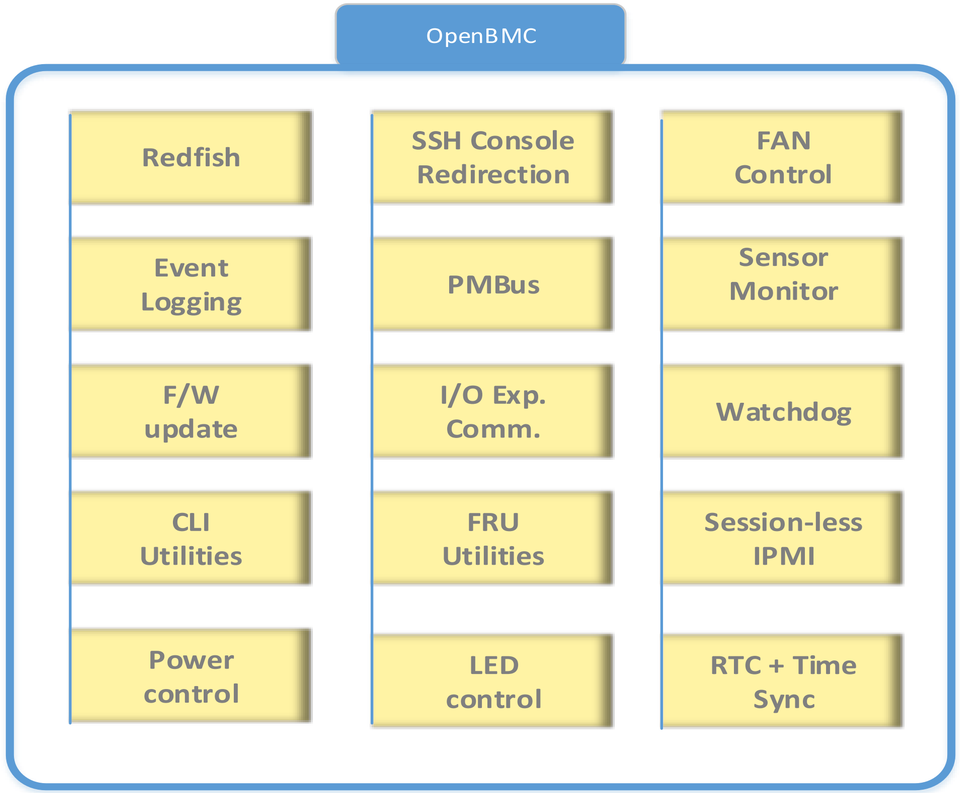
OPEN HARDWARE. OPEN SOFTWARE. OPEN FUTURE.

OPEN Compute Project

Microsoft OpenBMC

- **OpenBMC extended features for JBOD/JBOF:**

  - ✓ *Redfish Rest interface*
  - ✓ *Unified mechanism for component F/w update using Secure File Transfer*
  - ✓ *SSH Console redirection for BMC and x4 I/O expanders*
  - ✓ *BMC-I/O Expander communication over SMBus*
  - ✓ *Expander F/W recovery via BMC*
  - ✓ *Dual Flash support for recovery*
  - ✓ *F/W signing support*
  - ✓ *External Secure boot mechanism*
  - ✓ *PSU F/W update over PMBus*
  - ✓ *NTP Time synchronization*
  - ✓ *Session less IPMI support(very lite)*



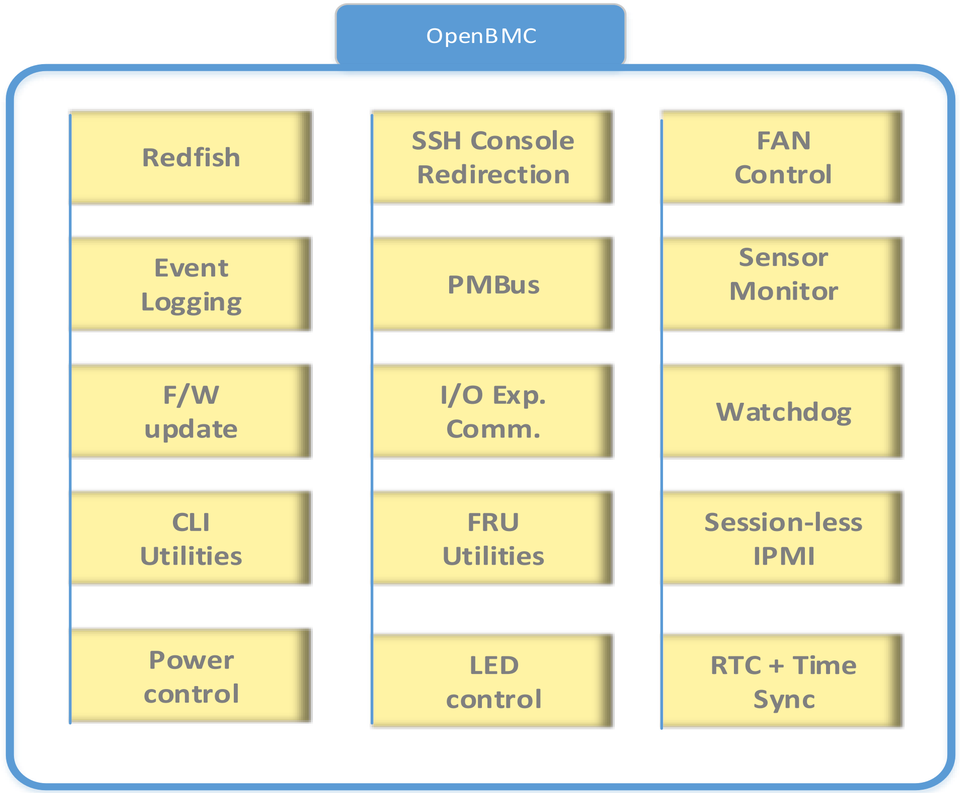
OPEN HARDWARE. OPEN SOFTWARE. OPEN FUTURE.

OPEN Compute Project

Microsoft OpenBMC

OpenBMC Performance optimizations:

- ✓ *Improving Redfish response time for event log and Sensor reading*
- ✓ *Supporting Concurrent redfish requests*
- ✓ *Faster F/W update*
- ✓ *Improving CPU usage and Load Average*
- ✓ *Optimizing Dbus access*
- ✓ *Improving boot time and File I/O via enabling SPI Driver for dual/QSPI modes*



Microsoft OpenBMC

Some of performance optimization efforts:

- ✓*DBus object aggregation*
- ✓*Reduce DBus interaction for Sensors retrieval over Redfish*
- ✓*Generate JSON without template*
- ✓*Reformat event log to be similar to redfish schema*
- ✓*Reduce/bypass DBus interaction for Events retrieval over Redfish*
- ✓*Replace gevent with Uswgi & GherryPy*
- ✓*Use GetManagedObjects() instead of GetAll() to get DBUS attributes*
- ✓*Sensor Monitor wakeup period increased to 1s (timeout_add_seconds() instead of gobject.timeout_add())*
- ✓*Use glib instead of gobject*
- ✓*Unused recipes Cleaned up*
- ✓*PSU FW Update & CLI tool converted from Python to C*

Microsoft OpenBMC

OpenBMC optimization results:

Operation	Original	Optimized	Original	Optimized	Notes
Boot Time to Redfish ready	105s	N/A	242Sec	84Sec	Ongoing effort to improve boot time
Read all sensors	85s (172 Sensors)	12s (172 Sensors)	22s (49 Sensor)	19s (98 Sensors)	Use GetManagedObjects() instead of GetAll() to get DBUS attributes: reduced from 85s to 33s Manually format Redfish response instead of using Python json.loads template: reduced from 33s to 12s
Dump Logs to zip file (4000 entries)	212s	12s	123s	39 Sec	Read log files directly instead of using DBUS. Change format from binary to text.
Read logs over Redfish (4000 entries)	420s	41s	24min	7min	Manually format Redfish response instead of using Python json.loads template - Using gevent webserver takes 41s - Changing to cherrypy takes 2m24s (need cherrypy to support multiple server requests)
BMC FW Update	N/A	8 minutes	6 min	4 min	Does not include pushing over SFTP (~ 2 minutes)
PSU FW Update	N/A	15 minutes	69min	4min	
CPU Utilization (Idle)	~90%	20-30%	100%	40%-100%	Replace gobject.timeout_add with glib.timeout_add_seconds to reduce CPU wakeups. Add sleep 1s for sensor polling intervals. Fix bugs that caused pmbus service to restart on sensor read error. dbus-daemon is the highest consumer at about 7-8%, followed by sensor_manager2 at about 5-7%.
Load Average	1.87	0.64	2.8	1.8	15 minute load average from uptime
	AST2520		**AST2400**		

Microsoft OpenBMC

OpenBMC Security Enhancements:

- ✓ *Security reviews conducted and security gaps identified*
- ✓ *F/W signing added*
  - ✓ *Hash : SHA256*
  - ✓ *RSA length: 2048*
- ✓ *Secure boot support added using dual Flash chip*
  - ✓ *BMC boots to primary flash after reset*
  - ✓ *Primary Flash is write protected*
  - ✓ *Uboot in primary flash verify signature of Uboot, Kernel and ROFS in secondary flash using Key stored in primary flash*
  - ✓ *After signature authenticated, Primary flash allows BMC to boot from Secondary flash.*

Microsoft OpenBMC

OpenBMC testing using PytestUtil

- An Open source tools created by Microsoft to support comprehensive verifications of BMC functionality across platforms
- A growing community as being adopted/contributed by Microsoft Partners
- Test framework can be extended using XML or Python to enable new test scenarios

```
<?xml version="1.0" encoding="utf-8"?>
<TestList duration="60:00:00" iterations="1">
 <test name="VerifyDefaultGetBiosConfig" delay="0" iterations="1"/>
 <test name="VerifyDefaultGetDefaultPowerLimit" delay="0" iterations="1"/>
 <test name="VerifyDefaultGetEnergyStorage" delay="0" iterations="1"/>
 <test name="VerifyDefaultGetNvDimmTrigger" delay="0" iterations="1"/>
 <test name="VerifyDefaultGetPowerLimit" delay="0" iterations="1"/>
 <test name="VerifyDefaultGetPsuAlert" delay="0" iterations="1"/>
 <test name="VerifyDefaultSetPowerRestorePolicy" delay="0" iterations="1"/>
 <test name="VerifyGetDeviceId" delay="0" iterations="1"/>
 <test name="VerifySetProcessorInfo" delay="0" iterations="1"/>
 <test name="VerifyGetProcessorInfo" delay="0" iterations="1"/>
 <test name="VerifySetMemoryInfo" delay="0" iterations="1"/>
 <test name="VerifyGetMemoryInfo" delay="0" iterations="1"/>
```

OPEN HARDWARE. OPEN SOFTWARE. OPEN FUTURE.

OPEN Compute Project

Microsoft OpenBMC

OpenBMC testing using PytestUtil

Verbose Logging

```
2017-08-05 06:48:23,046 - RestApiCall: GET: url = https://192.168.0.9:443/redfish/v1/Chas
2017-08-05 06:48:24,530 - J2010RestConcurrentStressTest.Execute (threadId 2): GET REST AP
    "@Redfish.Copyright": "Copyright 2014-2016 Distributed Management Task Force, Inc. (D
    "@odata.context": "/redfish/v1/$metadata#Chassis",
    "@odata.id": "/redfish/v1/Chassis/System/MainBoard",
    "@odata.type": "#Chassis.v1_3_0.Chassis",
    "Id": "MainBoard",
    "Name": "Management MainBoard",
    "ChassisType": "Module",
    "Manufacturer": "Microsoft",
    "Model": "J201A0",
    "SerialNumber": "BT870800027",
    "PartNumber": "M1020583-901",
    "PowerState": "On",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "Links": {
        "ManagedBy": [
            {
                "@odata.id": "/redfish/v1/Managers/System"
            }
        ],
        "ManagersInChassis": [
            {
                "@odata.id": "/redfish/v1/Managers/System"
            }
        ]
    },
    "@Message.ExtendedInfo": [
        {
            "Oem": {
                "Microsoft": {
                    "@odata.type": "#Ocs.v1_0_0.Message",
                    "CompletionCode": "Success"
                }
            }
        }
    ]
}
```

Hierarchical report generation

	Cycle 0	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7	Cycle 8
Time Duration	0.110541	0.111002	0.111555	0.108894	0.11248	0.111784	0.11316	0.113241	0.11424
GetMACAddressTest	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED
BmcDecompressionTime	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED	PASSED
NegativeRestApiTests	FAILED	FAILED	FAILED	FAILED	FAILED	FAILED	FAILED	FAILED	FAILED
GPIOTest	FAILED	FAILED	FAILED	FAILE					
BMCConsoleRedirectionTest	PASSED	PASSED	PASSED	PASSE					
ExpanderConsoleRedirectionTest	PASSED	PASSED	PASSED	PASSE					
CheckFolderSize	PASSED	PASSED	PASSED	PASSE					
J2010RestFunctionalTest	PASSED	PASSED	PASSED	PASSE					
AcPowerCycleStressTest	PASSED	PASSED	PASSED	PASSE					
MixedTrafficStressTest	PASSED	PASSED	PASSED	PASSE					
J2010RestConcurrentStressTest	PASSED	PASSED	PASSED	PASSE					
FruReadWriteTest	PASSED	PASSED	PASSED	PASSE					
ExpanderBoardPowerControlTest	PASSED	PASSED	PASSED	PASSE					
VerifyGetChanAuthCapabilities	PASSED	PASSED	PASSED	PASSE					
BmcFwUpdateTest	PASSED	PASSED	PASSED	PASSE					
PyTestUtil	v1.12								
BMC Version	J2010.BC.01.09.00								

2017-08-05 00:01:09,630 - RestApiCall: HTTP GET Request https://1
2017-08-05 00:01:10,505 - RestApiCall: HTTP GET Request
https://192.168.0.9:443/redfish/v1/Chassis/System/MainBoardsKo1Me
Uo8Eq9Xv3Dl9Lo4Wo9Ez6Kw2Jj2He9Mc8Lh2Vr0Ys1Ly2Qo7Vc9Fz2Tt01
Kd2Wv2Tf1Qf4Kz9Bd5Dq0Bk9Ur6Pr4Si9Nd2Ad1Mq4Go2Jy7Xh8Ao3K
2017-08-05 00:01:11,723 - RestApiCall: HTTP GET Request https://19
2017-08-05 00:01:12,755 - RestApiCall: HTTP GET Request
https://192.168.0.9:443/redfish/v1/Chassis/System/MainboardPn9Gt
5Cb5Se5Vy3Cr5Vb9Cm3Di3Gm2Hc7Mv6Cw4Hr1Fk4Po8Sk9Pa2Wi0Ms
2017-08-05 00:01:14,114 - RestApiCall: HTTP GET Request https://19
2017-08-05 00:01:15,207 - RestApiCall: HTTP GET Request
https://192.168.0.9:443/redfish/v1/Chassis/System/StorageEnclosu
y8No9Yj3Tr1Tz0Mm1Oi7Og1Tf8Pb6Ga5Nt2Km2Bo4Gm2Iv8Vy1Bz2Tv9
5Nz2Ih0Vc0Qo4Gj1Qm2Mr2Gs0Cr9Kf9Zx6Ny7Bi4We8Cz4Gl0Bx8Hx1Mh8
2017-08-05 00:01:16,239 - RestApiCall: HTTP GET Request https://19
2017-08-05 00:01:17,052 - RestApiCall: HTTP GET Request
https://192.168.0.9:443/redfish/v1/Chassis/systemLe8Sf6Tg4Rw1Al5C
Uh5Sa2Gl0Wx1Nv3Jw7Yb3Bm3Hb5Xa4Ko3Tu2Wk7Ta9Ar0Qm3Ji4Ye6W
2017-08-05 00:01:17,802 - RestApiCall: HTTP GET Request https://
2017-08-05 00:01:18,614 - RestApiCall: HTTP GET Request
https://192.168.0.9:443/redfish/v1/Chassis/System/fruDf3Gz0Ui4Vg5N
1Yd0Bd9Rp5Eq7Jj1Yl0Tn2Wa4Nh3Ku4Lb8Gg1Sg5Lh0Rj1Na8Ao2Ye9S
2017-08-05 00:01:19,505 - RestApiCall: HTTP GET Request https://
2017-08-05 00:01:20,427 - RestApiCall: HTTP GET Request

OPEN HARDWARE. OPEN SOFTWARE. OPEN FUTURE.

OPEN Compute Project

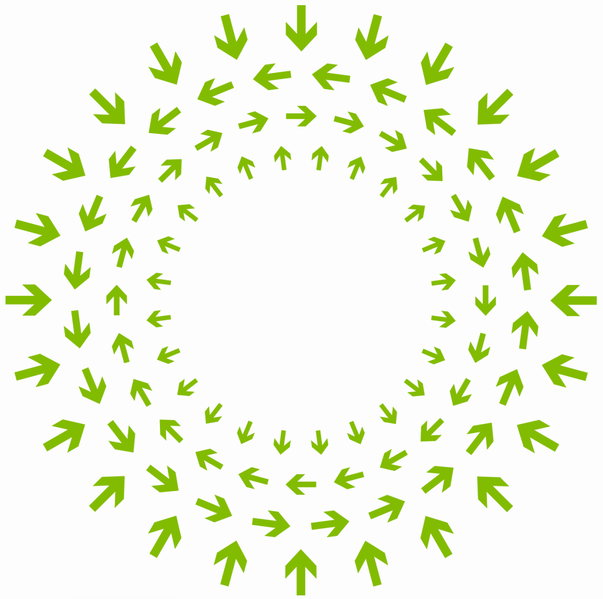
Microsoft OpenBMC

Summary:

- Feature set ,Performance ,Availability and Security are cornerstones of H/W manageability at cloud scale.

- OpenBMC has good potential to grow into a robust manageability solution if community :
  - Can establish a traceability between manageability requirement and design/implementation
  - Can establish a development model that effectively addresses key areas of manageability at cloud scale
  - Can leverage lesson-learnt and wide industry expertise to improve functionalities

OPEN Compute Project



OPEN

Compute Project