Bare Metal Container

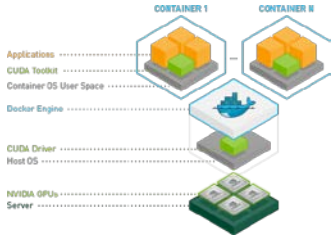
National Institute of Advanced Industrial Science and Technology(AIST)

National Institute of Advanced Industrial Science and Technology AIST

Kuniyasu Suzaki

Contents

- Background of BMC
  - Drawbacks of container, general kernel, and accounting.
- What is BMC?
- Current implementation
- Evaluation
- Extension
  - NVIDIA Docker, Moby, Intel Clear Container, etc.



moby project

Intel Clear Containers

CoreOS

snappy

RANCHEROS

DC/OS

PROJECT ATOMIC

PHOTON OS™

- Conclusions

Background of BMC 1/3

Drawback of Container

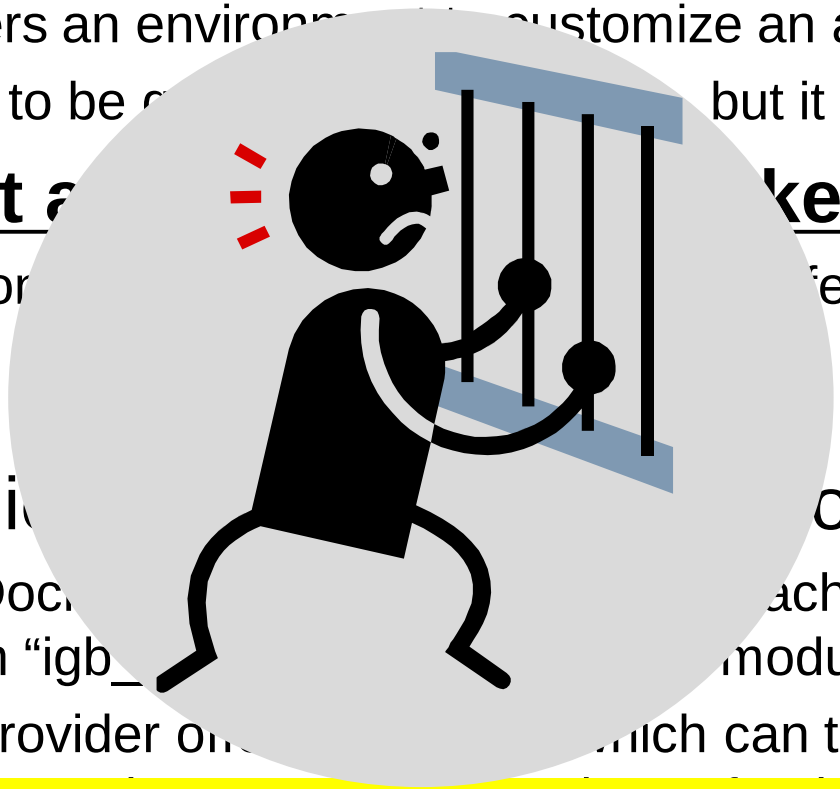
- Container technology (Docker) becomes popular.
  - Docker offers an environment to customize an application easily.
  - It looks like to be good for an application, but it is a **server centric.**
- **<u>It does not allow to change the kernel.</u>**
  - Kernel options passed through /sys are not effective.

- Some applications cannot run on Docker.
  - DPDK on Docker does not work on some machines, because it depends on “igb_uio” and “rte_kni” kernel modules.
    - Some provider offers the kernel which can treat DPDK on Docker, but it is case by case solution. It is not fundamental solution.

Background of BMC 1/3

Drawback of Container

- Container technology (Docker) becomes popular.
  - Docker offers an environ[illegible]ustomize an application easily.
  - It looks like to be [illegible] but it is a **server centric.**
- **It does not a[illegible]kernel.**
  - Kernel optior[illegible]fective.
- Some appli[illegible]ocker.
  - DPDK on Doc[illegible]achines, because it depends on "igb_[illegible]modules.
    - Some provider o[illegible]hich can treat DPDK on Docker,



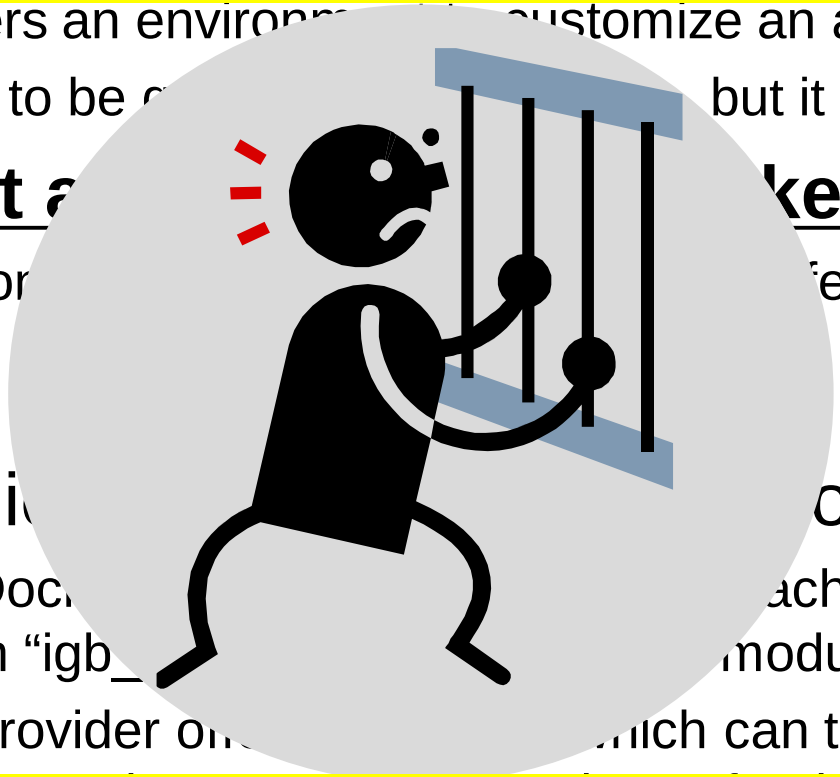
Container is a jail for a kernel optimizer.

Background of BMC 1/2

HPC users want to optimize the kernel *for their applications*. **Kernel is a servant.**
Container way is not fit for them.

- – Docker offers an environ customize an application easily.
- – It looks like to be but it is a **server centric.**

- **It does not kernel.**
  - – Kernel optior fective.



- Some appli ocker.
  - – DPDK on Doc achines, because it depends on "igb_ modules.
    - Some provider o hich can treat DPDK on Docker,

Container is a jail for a kernel optimizer.

Background of BMC 2/3

General kernel leads weak performance

- Arrakis[OSDI'14] showed that nearly 70% of network latency was spent in the network stack in a Linux kernel.
- Many DB applications (e.g., Oracle, MongoDB) reduce the performance by THP (Transparent Huge Pages) which is enabled on most Linux distributions.

ORACLE Help Center

4.11 Disabling Transparent HugePages

Oracle recommends that you disable Transparent HugePages before you start installation.

mongoDB

Transparent Huge Pages (THP) is a Linux memory management system that reduces the overhead of Translation Lookaside Buffer (TLB) lookups on machines with large amounts of memory by using larger memory pages.

However, database workloads often perform poorly with THP, because they tend to have sparse rather than contiguous memory access patterns. You should disable THP on Linux machines to ensure best performance with MongoDB.

Background of BMC 2/3

General kernel leads weak performance

- Arrakis[OSDI'14] showed that nearly 70% of network latency was spent in the network stack in a Linux kernel.
- Many DB applications (e.g., Oracle, MongoDB) reduce the performance by THP (Transparent Huge Pages) which is enabled on most Linux distributions.

ORACLE Help Center

4.11 Disabling Transparent HugePages

Oracle recommends that you disable Transparent HugePages before you start installation.

It is not fundamental solution.
HPC users want to optimize the kernel *for their applications*. **Kernel is a servant.**

However, database workloads often perform poorly with THP, because they tend to have sparse rather than contiguous memory access patterns. You should disable THP on Linux machines to ensure best performance with MongoDB.

Background of BMC 3/3

Power consumption for each application

- Current power measurement is coarse.
  - Power Usage Effectiveness: PUE only shows usage of data-center scale.
  - Current power consumption is theme for vender and administrators
- Users have no incentive for low power, even if they make a low power application.
  - Current accounting is based on time consumption.

Background of BMC 3/3

Power consumption for each application

- Current power measurement is coarse.
  - Power Usage Effectiveness: PUE only shows usage of data-center scale.
  - Current power consumption is theme for vender and administrators
- Users have no incentive for low power, even if they make a low power application.
  - Current accounting is based on time consumption.

There is no good method to measure power consumption "for an application".
No accounting which considers power consumption.

What is BMC?

- BMC(Bare-Metal Container) runs a container (Docker) image with a suitable Linux kernel on a remote physical machine.
  - Application on Container can change kernel settings and machine which fit for it. BMC extracts the full performance.
  - On BMC, the power on the machine is almost used for the application.
    - BMC tells the power usage on each machine architecture. Users can know which architecture is good for their application.

BMC offers incentives to customize kernel and select machine architecture

What is BMC?

- BMC(Bare-Metal Container) runs a container (Docker) image with a suitable Linux kernel on a remote physical machine.
  - Application on Container can change kernel settings and machine which fit for it. BMC extracts the full performance.
  - On BMC, the power on the machine is almost used for the application.
    - BMC tells the power usage on each machine architecture. Users can know which architecture is good for their application.

BMC offers incentives to customize kernel and select machine architecture

What is BMC?

- BMC(Bare-Metal Container) runs a container (Docker) image with a suitable Linux kernel on a remote physical machine.
  - Application on Container can change kernel settings and machine which fit for it. BMC extracts the full performance.
  - On BMC, the power on the machine is almost used for the application.
    - BMC tells the power usage on each machine architecture. Users can know which architecture is good for their application.

BMC offers incentives to customize kernel and select machine architecture

What is BMC

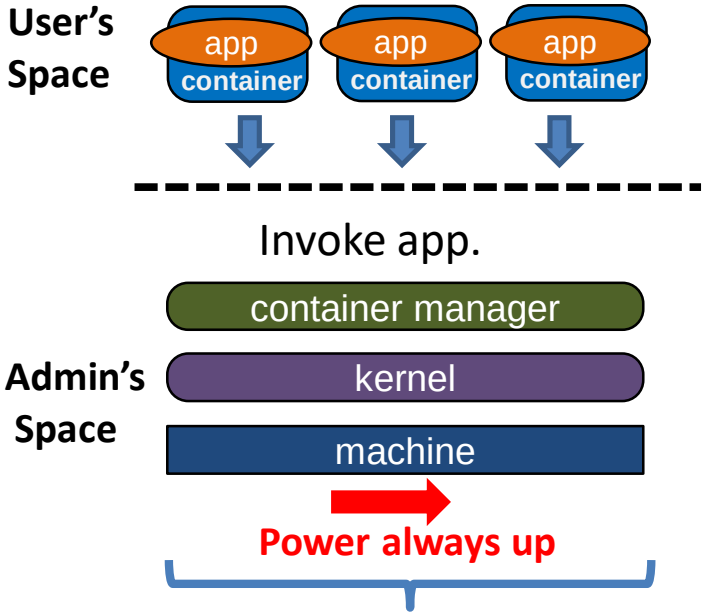
- BMC(Bare-Metal Container) run ker) image with a suitable Linux kernel al machine.
  - Application on Container tings and machine which fit for application erformance.
  - It means the powe nost used for an application.
    - BMC tells th ch machine architecture. Users can kno good for their application.

Application Centric Architecture

tives to customize kernel and
ct machine architecture

Comparison

Traditional Style (Ex: container)

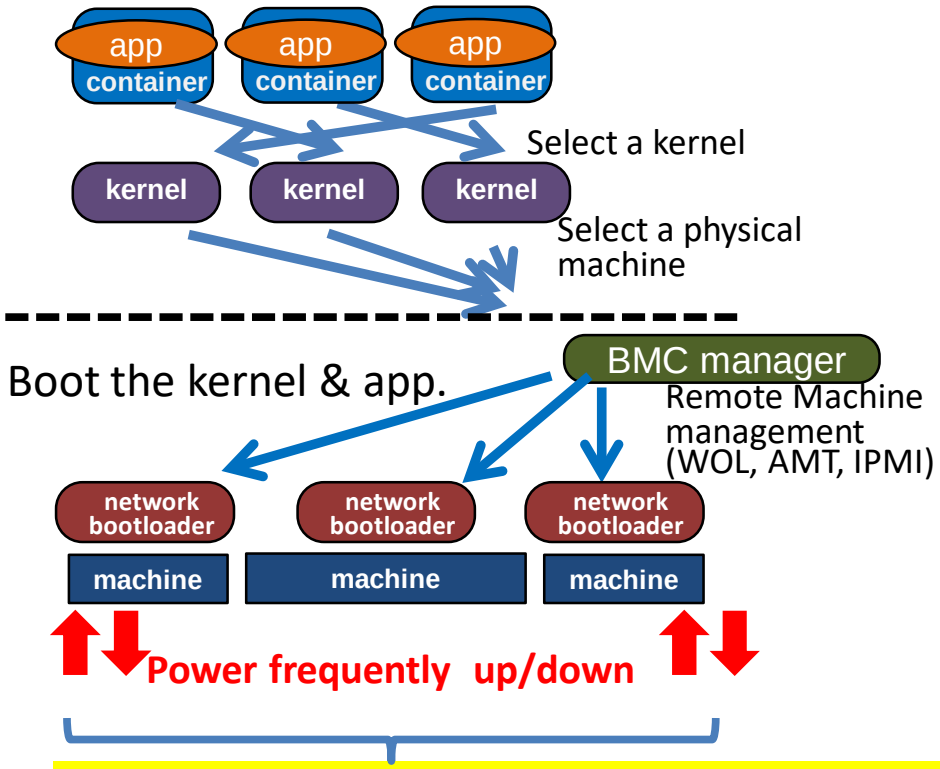


Server Centric Architecture

Pros:
- Multi Tenant
- Quick Response (No Rebooting)

Cons:
- Kernel is not replaced.

BMC

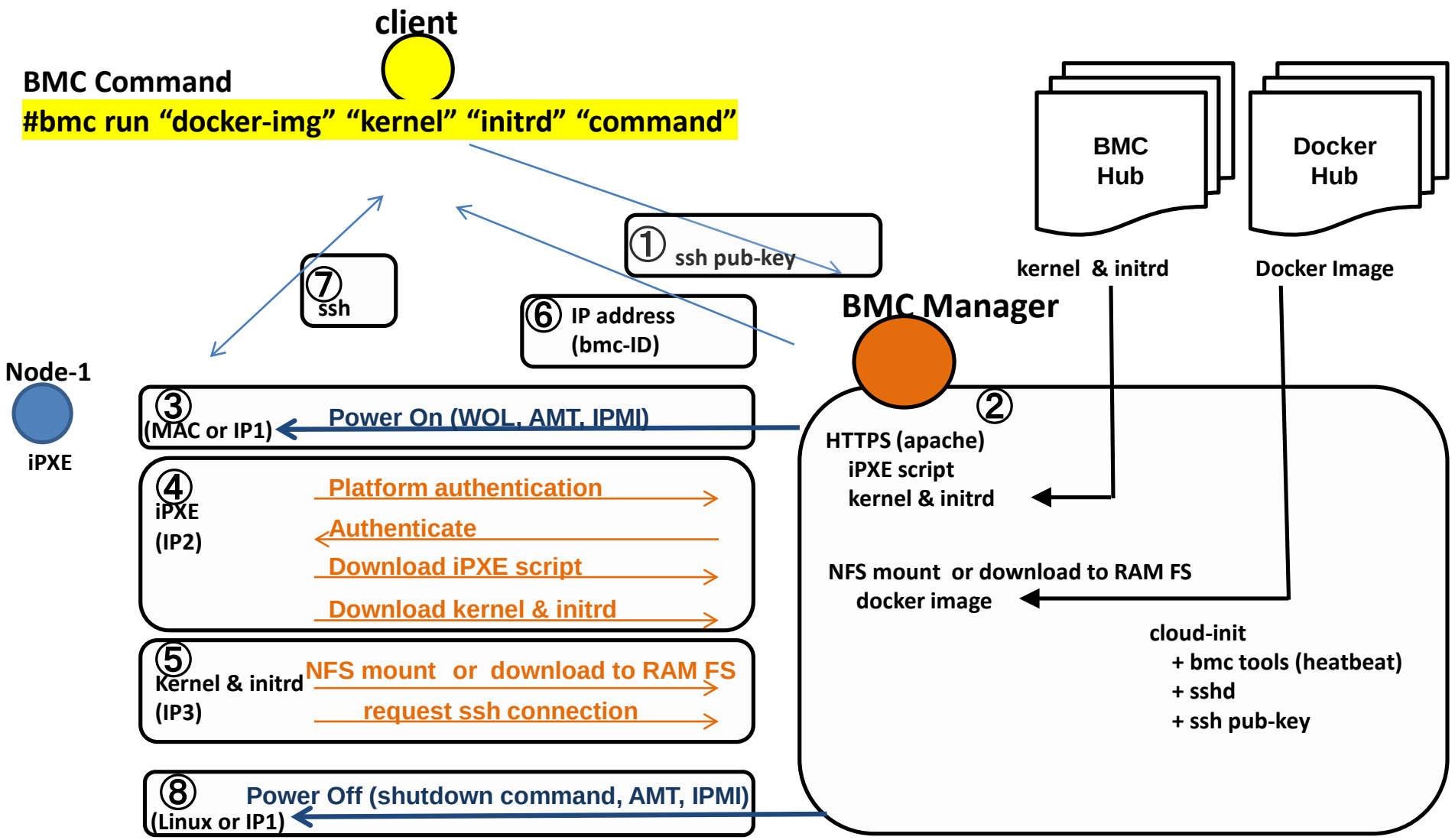


Application Centric Architecture

Pros:
- Apps can select a kernel & hardware.
- Apps occupy the machine and extract the performance.

Cons:
- Set up overhead (Rebooting)

Procedure to execute BMC command



Remote Machine Boot Procedure

1. Power-on a node machine with Remote Machine Management (WOL, Intel AMT, IPMI)
2. Network Boot Loader (iPXE)
   – Get kernel and intird from a HTTP/HTTPS server.
3. The downloaded initrd mounts a Docker image.
   - NFS mode
   - RAM FS mode
4. Boot procedure in a Docker image
   – Fortunately, Docker image keeps boot procedure.
5. SSH is connected from BMC command
   – Run an application.

Remote Machine Management

	WOL	Intel AMT	IPMI
Protocol	Magic Packet (MAC address)	HTTPS (IP address)	RMPC (IP address)
Power-On	✔	✔	✔
Power-Off	×	✔	✔
Security	×	Password	Password
Comment	Most PCs have WOL.	High level Intel machine	Server Machine (Slow BIOS)

Network Boot Loader

- PXE is the most famous, but it is limited for LAN, because it depends on “magic packet” of Layer 2.

- BMC uses iPXE which download “kernel” and “initrd” from HTTP/HTTPS.
  - iPXE is custimzed by its scripting language. BMC uses it.

```
#!ipxe
ifopen net0
set net0/ip 192.168.0.101
set net0/netmask 255.255.255.0
set net0/gateway 192.168.0.1
set dns 192.168.0.1
:loop
chain http://192.168.0.200/cgi-bin/baremetal.ipxe || goto waiting
exit
:waiting
sleep 1
goto loop
```

- The iPXE downloads kernel and initrd.

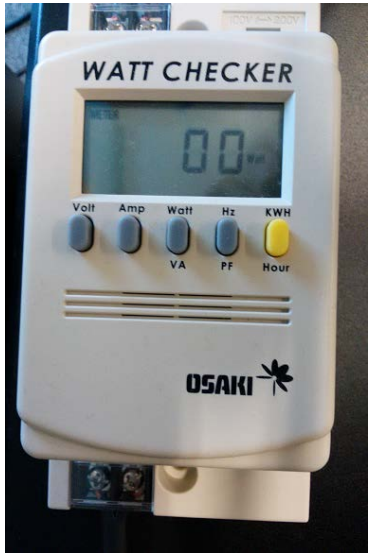
How to boot OS (Linux)

- The downloaded “initrd” is customized to mount an Docker image. It offers 2 mount methods.
  - NFS mode
    - Download necessary data only and fast boot, but it needs to download data to run applications after boot.
  - RAMFS mode
    - Download full Docker image and slow boot, but application runs fast after boot.

- Boot procedure in the Docker image.
  - An Docker image keeps boot procedure for each application because each application package designed to include them.
  - BMC utilizes these boot procedures to rum daemons, such as the SSH, because an application in the Docker image is executed by remote procedure calls from BMC manager.

Power Consumption

- Each node has power meter “WattChecker”.
- WattChecker measures power consumption from the power-on caused by WOL,AMT, or IPMI.
- BMC manager keeps the log of the power consumption.
- The log is used for power accounting.



- It is coarse, but it shows affinity between application and architecture.

Current Implementation

- Current BMC Manager is implemented with shell script.
  - 4, 500 LOC.
- Power consumed on each node is measured by WattChecker.
- We have tried several machines as BMC nodes.
  - From Atom to Xeon.
  - Application can select machine considering power consumption.

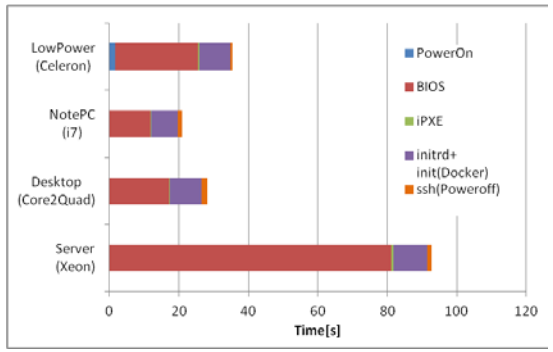
Spec of Test Machines

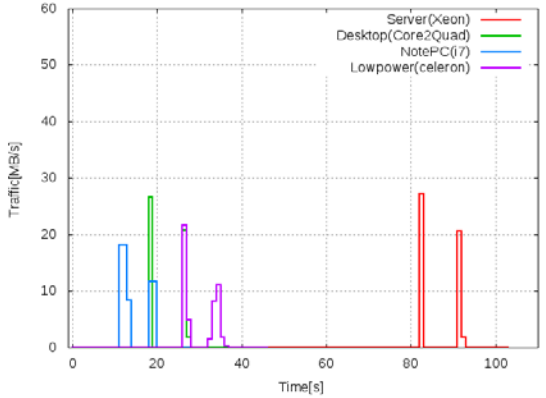
	Remote machine manage ment	CPU,Core/thread, Clock (Burst time), Power	Logical performance GFLOPS (Burst time)	Issue date	Memory	NIC (queue)
Low Power Intel NUC 5CPYH	WOL	Celeron (N3050),2/2, 1.6 (2.16)GHz,8W	6.4 (8.6)	2015	8GB	RealTek r8169 (1)
NotePC Lenovo ThinkPAD T430s	Intel AMT	i7 (3520M)2/4, 2.9(3.6)GHz, 35W	46.4 (57.6)	2012	16GB	Intel e1000 (1)
DesktopPC Dell Optiplex 960	Intel AMT	Core 2Quad (Q9400) 4 /4, 2.66GHz,95W	42.656	2008	16GB	Intel e1000 (1)
Server Dell PowerEdge T410	IPMI	Xeon (X5650) 6/12,2.66(3.06)GHz,95W	63.984 (73.44)	2010	8GB	Broadcom NeXtreme II(8)

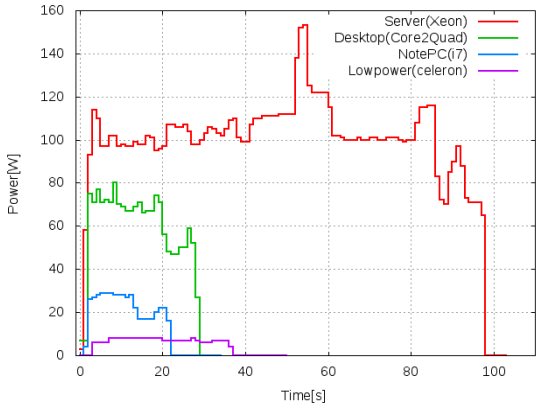
Boot performance (overhead)

Time | Network | Power

NFS

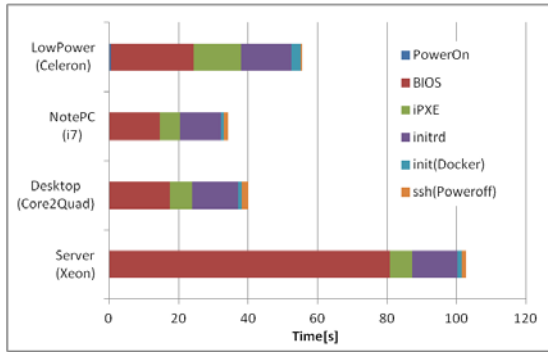


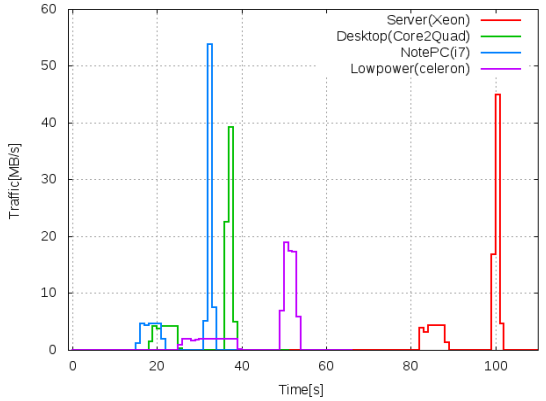


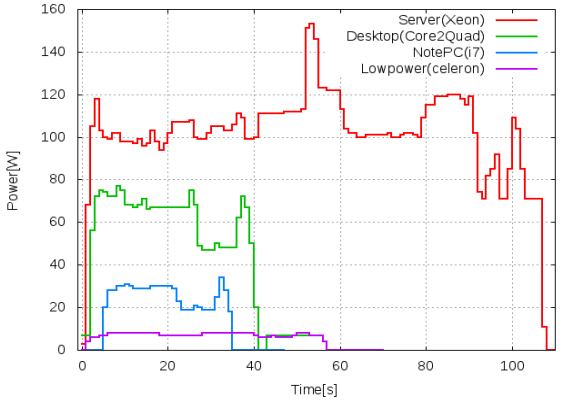


OO

RamFS







	Time(s)	Power (j)	Traffic (MB)
NFS mode			
Celeron	35.4	242	49.5
Core2 Quad	28.1	1,773	49.3
I7	20.9	481	49.1
Xeon	92.6	9,932	49.8
RAMFS mode			
Celeron	55.6	402	92.9
Core2 Quad	40.0	2,493	92.8
i7	34.3	775	92.8
Xeon	102.7	11,015	92.6

46 times

- They are BMC's overhead.
- The performance improved by optimization must surpass the overhead.

Tested Application and Optimization

Application	Optimization
Matrix Multiplication with OpenBlas	Hyper Threading off
Redis benchmark	Transparent Huge Pages off
Apache benchmark	Receive Flow Steering off

- This presentation shows the result of Matrix multiplication with/without Hyper Threading.
  - The experiment measured the time for 10 times of matrix multiplications on OpenBlas optimized for each machine.

Performance Difference

10 times of matrix multiplications [12800:12800]
on OpenBlas optimized for each machine. .

	Time (s)	Power (j)	GFLOPS	Power/(GFLOPS*time)
Celeron	12,783.8	125,084	2.99 (34.7%)	3.27
Core2Quad	1,060.2	140,346	39.8 (93.3%)	3.32
i7 HTT-on	961.4	55,315	43.8 (76.0%)	1.31
i7 HTT-off	827.1	45,364	50.9 (88.4%)	1.08
Xeon HTT-on	945.6	211,908	44.6 (60.7%)	5.02
Xeon HTT-off	698.9	151,760	60.5 (82.4%)	3.59

() shows the rate from logical performance

Celeron (Atom Core) is not cost effective CPU.

- The results show no hyper threading is better.
- Xeon shows the best performance, but i7 shows cost effective.

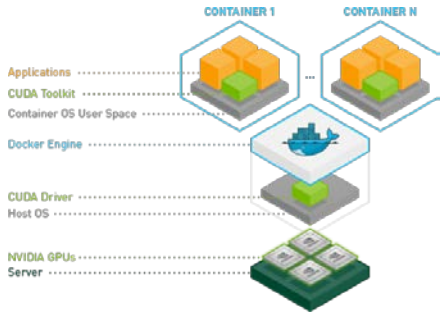
Performance improvement which compensates the boot overhead

	Boot overhead	Improvement at [6400:6400]	Improvement at [12800:12800]
Time (sec)			
i7	35.4	15.9	134.3
Xeon	108.0	29.8	246.7
Power (joule)			
i7	1,805.3	1,150	9,951
Xeon	11,274.5	6,792	60,148

- Overheads caused by booting were compensated before [12800:12800].

BMC Extension

- NVIDIA-Docker
- Moby
- Intel Clear Containers
- OSes for Container



moby project

Intel Clear Containers

CoreOS

snappy

RANCHEROS

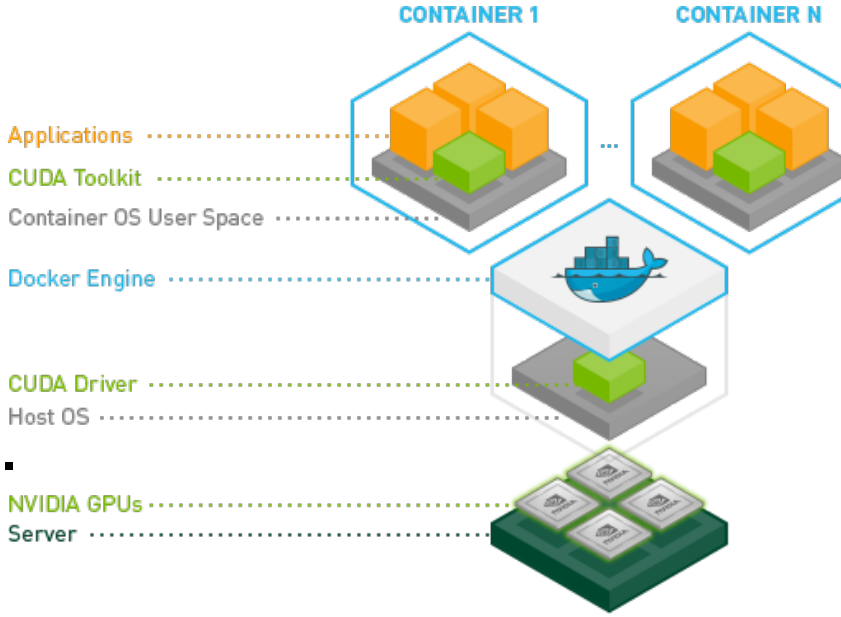
DC/OS

PROJECT ATOMIC

PHOTON OS™

Extension for “NVIDIA-Docker”

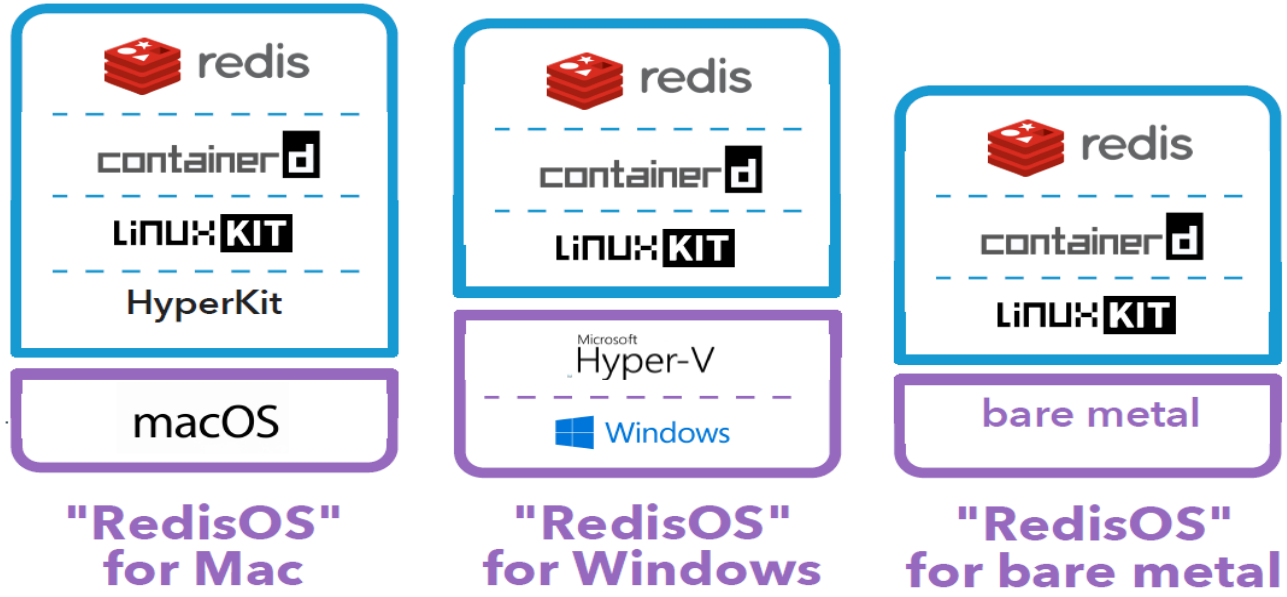
- NVIDIA-Docker runs CUDA applications.
  - It manages CUDA and driver of NVIDIA GPU.
    - A suitable CUDA is added Docker image automatically.
    - So, users do not need to install CUDA.
  - Users don’t need to care about CUDA version.



- BMC is now customizing to add CUDA which matches to NVIDIA-driver.
  - The target is TensorFlow on NVIDIA-Docker.
    - https://hub.docker.com/r/tensorflow/

Moby

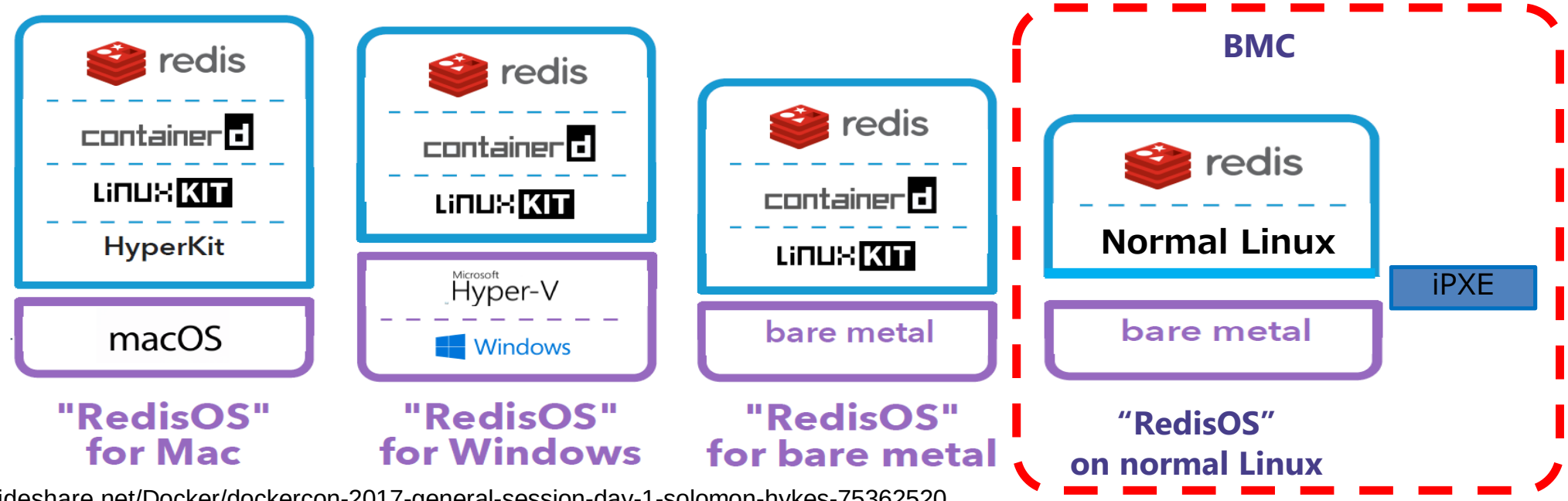
- Moby is a framework to assemble container systems.
- Moby allows to run “redis” container without Docker!
  - Why do we need “containerd” and “LinuxKit”?
  - Redis should run on native Linux kernel?



https://www.slideshare.net/Docker/dockercon-2017-general-session-day-1-solomon-hykes-75362520

Moby

- Moby is a framework to assemble container systems.
- Moby allows to run “redis” container without Dock!
  - Why do we need “containerd” and “LinuxKit”?
  - Redis should run on native Linux kernel?



https://www.slideshare.net/Docker/dockercon-2017-general-session-day-1-solomon-hykes-75362520

Intel Clear Container

- Intel Clear Container is a counter-thesis of container. It encourages to use virtualization (Intel VT).
  - Intel insists that Intel VT and Linux's boot (< 200ms!) are fast. Why don't you use virtualization? Virtualization offers stronger isolation.

- BMC' s offers counter-thesis against Intel Clear Container.
  - If Linux's boot is fast, why don't you use native Linux?
  - BMC offers more flexibility of kernel customization.

Current BMC target

- Many Docker OSes are proposed.
  - CoreOS
  - RancherOS
  - Snappy Ubuntu Core
  - RedHat Project Atomic
  - Mesosphere DC/OS
  - VMware Photon
- BMC can use these Linux kernels.
  - BMC only requires kernel and container image.

Related works

- Triton [Joyent's product]
  - Triton = Docker + SmartOS.
    - In order to optimize, user needs to customize SmarOS.
- LinuxBIOS/BProc Cluster[HPCS'02]
  - Testbed for kernel test. It is not so easy to implement because it requires to replace BIOS.
- SLURM[ICDCN'14]
  - Measure power consumption for an application. It depends on function to measure power (Intel RAPL: Running Average Power Limit, or CRAY machine).

Conclusions

- BMC (Bare-Metal Container) runs a container (Docker) image with a suitable Linux kernel on a remote physical machine.
- The overhead of BMC was compensated by the improved performance of applications.

- Official HP: http://www.itri.aist.go.jp/cpc/research/bmc/
- Docker Image for BMC manager: https://hub.docker.com/r/baremetalcontainer/
- Source Code: https://github.com/baremetalcontainer