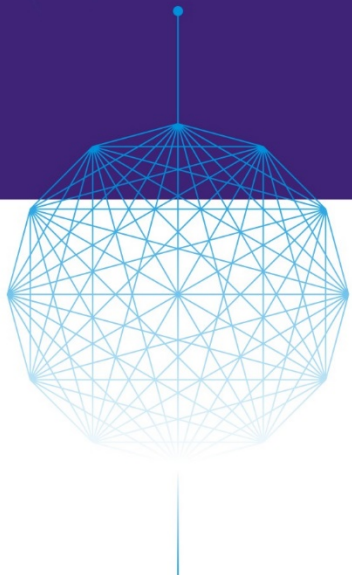







# DPDK SUMMIT CHINA 2017



主办方：

参与方： 腾讯云  ZTE  美团云  Panabit®  太一星晨  UnitedStack 联合云  云杉网络 Yunshan Networks

协办方： SDNLAB 专注网络创新技术 视频支持方： IT大咖说 网络全媒平台

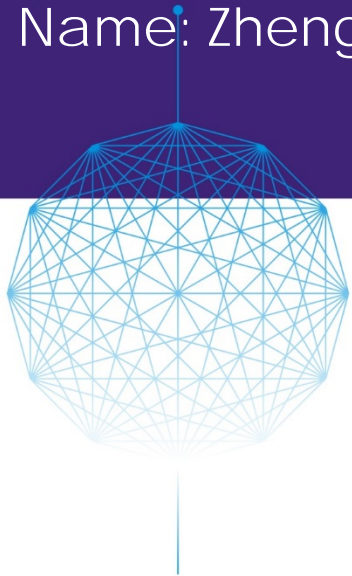



# A high-speed PMD towards LXC networking

Company: UnitedStack



Title: network virtualization engineer

Name: Zheng jie



主办方: 

参与方:  腾讯云  ZTE  美团云  Panabit®  太一星辰 Balance Your Networks   云杉网络 Yunshan Networks

协办方:  SDNLAB 专注网络创新技术 视频支持方:  IT大咖说



# Who we are

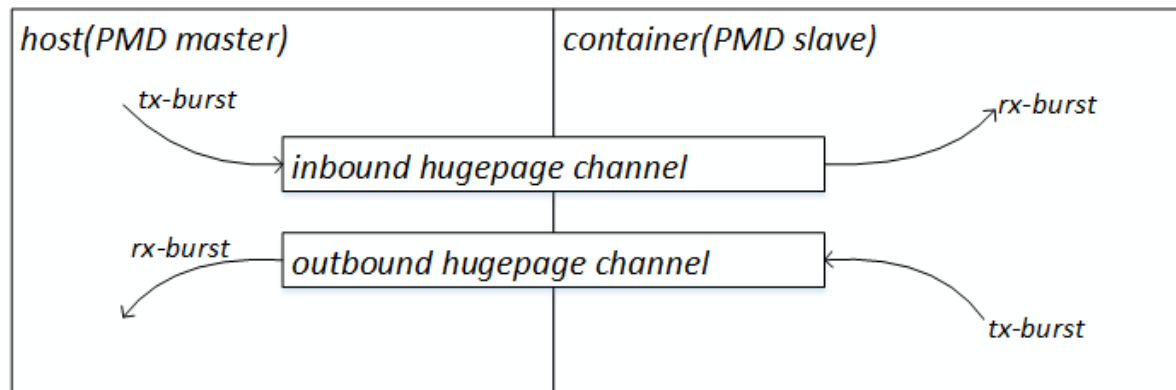
- OpenStack(Gold Member) IaaS provider
- NFV enhanced Neutron Networking
- DPDK powered applications include:
  - Distributed LoadBalancer
  - Server based Data Center fabric infrastructure
  - ... ..





## Accelerate LXC networking

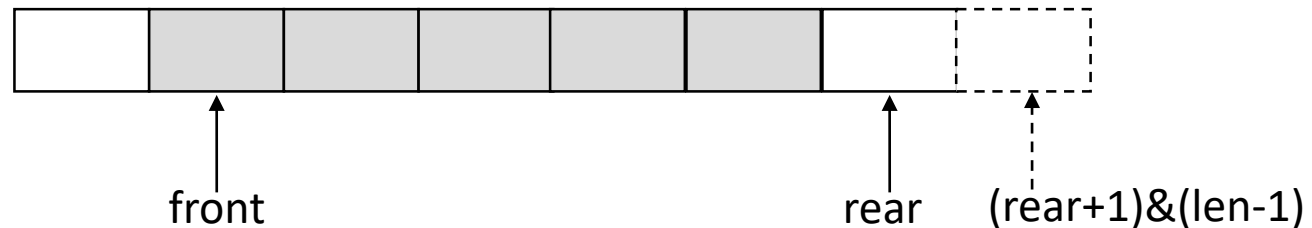
- SR-IOV for LXC
- Universal virtio PMD for LXC
- Specialized transport medium as IPC is in need





## How to structure the SHM

- VECRING ---- Vectorized Ring Buffer
- Fundamental ring element (block)---- aligned cache line
- Yet single producer & single consumer queuing model
- Masked ring indicator (as with DPDK ring implementation), never wrap back



\* where len is power of 2





## Associate VECRING with mbuf

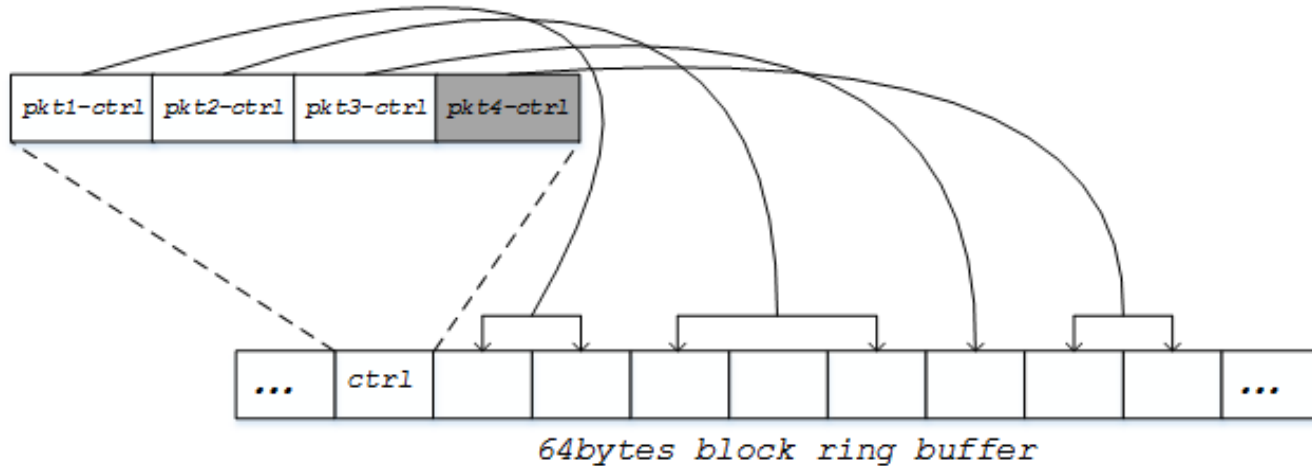
- Control block (as metadata) precede, Data blocks follow
- Control information associated with a mbuf takes 16-bytes
  - starting-index, length, whether-is-fetched, whether-is-end-of-block, etc.
- Control information(at maximum 4) can aggregate into one control block.
  - Enqueue x4
  - Enqueue x2
  - Enqueue x1





## Enqueue with x4 speed

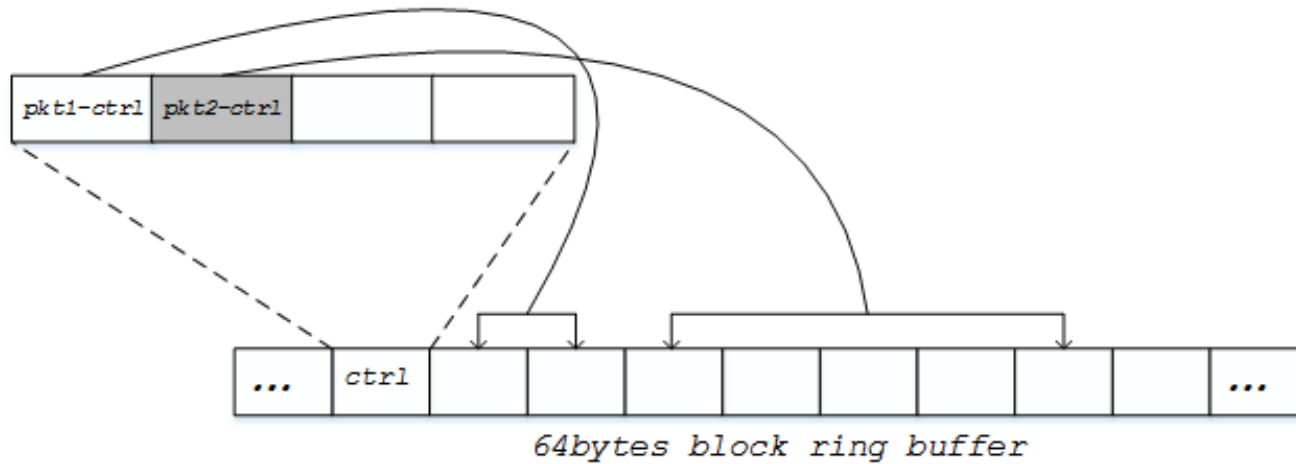
- Aggregate 4x control information into one control block





## Enqueue with x2 speed

- Aggregate 2x control information into one control block

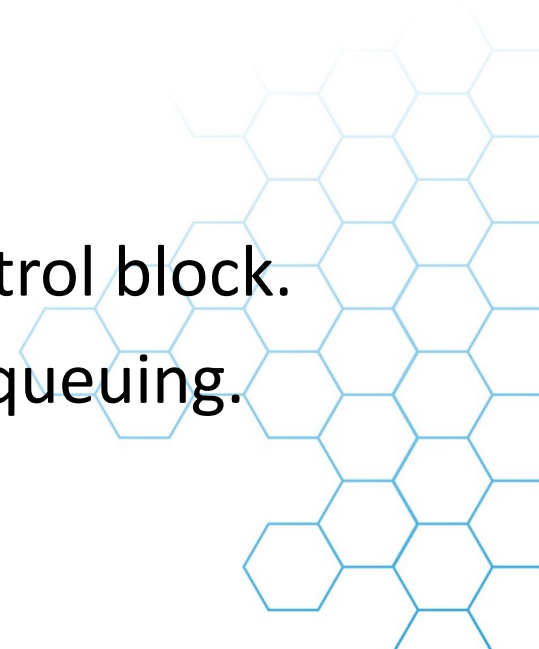






## Bulking Dequeuing

- Fetch a control block
- Walk through control information one by one until reaching end of block
  - Call `rte_pktmbuf_alloc()`
  - Copy packet payload from data blocks
  - Mark it as **fetch**
- If nothing wrong happens, proceed rear indicator to next control block.
- Else mark the control block as partially fetched, can cease dequeuing.





## How to better access memory

- Non-temporal behavior
  - Will not pollute cache layout
- CPUID supported
  - up to SSE4.2 or AVX2
  - streaming SIMD LOAD/STORE instructions
- streaming loading buffer
- Write combining



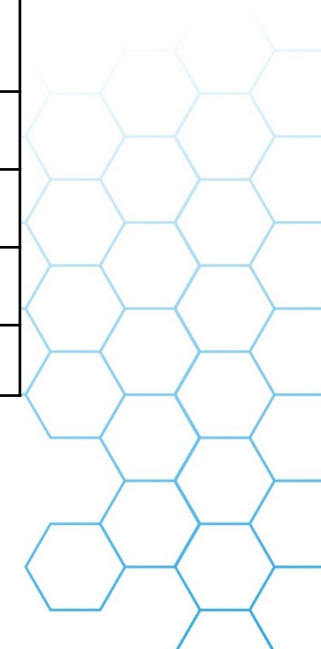


## DPDK PMD encapsulation

- vdev prefix ---- eth\_vecring
- Parameter list:

name	Mandatory	type	remark
domain	yes	string	Indicate which container it belongs to
link	yes	string	link identifier, there maybe multiple links inside a container, links are to distinguish themselves
socket	no	number	Which numa socket the link belongs to
mac	no	mac	If not provided, randomize it.
master	no	[true,false]	PMD role, default is false.
queue	no	int	The length of queue,default is DEFAULT_NR_BLOCK64

- `--vdev=eth_vecring0,domain=[string],link=[string],master=[int],mac=[mac],socket=[int]`





## Environmental pre-setup

```
#create a domain with name:demo
[root@localhost dpdk-16.07-vecring]#./vecutils.sh dom_alloc demo

#list all available domains
[root@localhost dpdk-16.07-vecring]#./vecutils.sh dom_ls
0:domain:demo huge-dir:mounted
1:domain:testcontainer huge-dir:mounted
2:domain:vnf1 huge-dir:mounted

#map the domain directories into container
#by including mapping entries in LXC container's definition file
```





## Environmental setup

```
#host side as master
```

```
[root@localhost ~]#... --vdev=eth_vecring0,domain=testcontainer,\  
link=tap456,master=true,mac=00:ec:f4:bb:d9:7f,socket=1
```

```
#container side as slave
```

```
[root@localhost ~]#... --vdev=eth_vecring0,domain=testcontainer,link=tap456
```

```
#the generated metadata and hugepage files
```

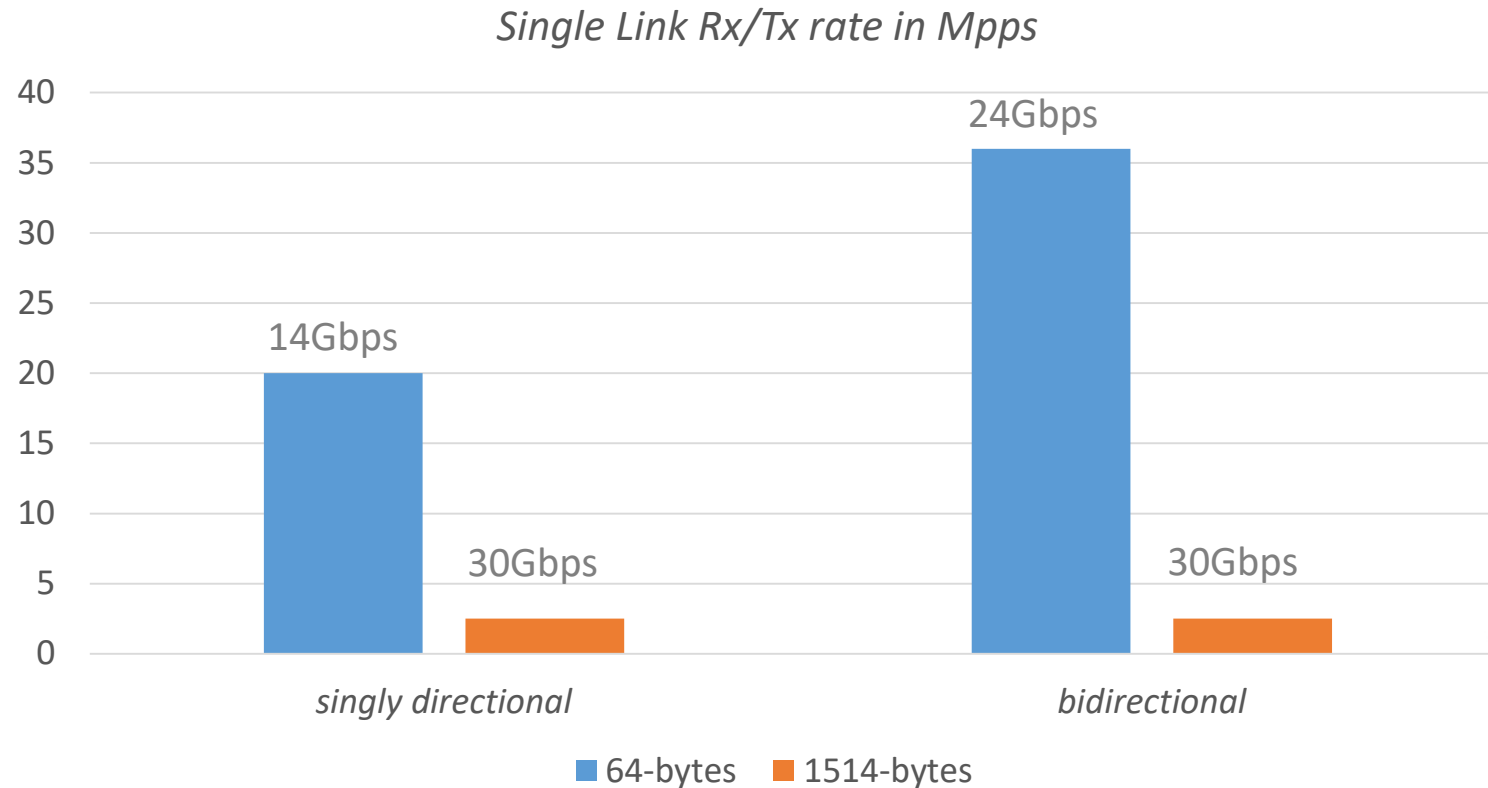
```
[root@localhost testcontainer]# tree
```

```
.  
├── huge  
│   ├── vecring-tap456.inbound-0  
│   ├── vecring-tap456.inbound-1  
│   ├── vecring-tap456.outbound-0  
│   └── vecring-tap456.outbound-1  
└── tap456.metadata
```





## Single Link rx/tx rate

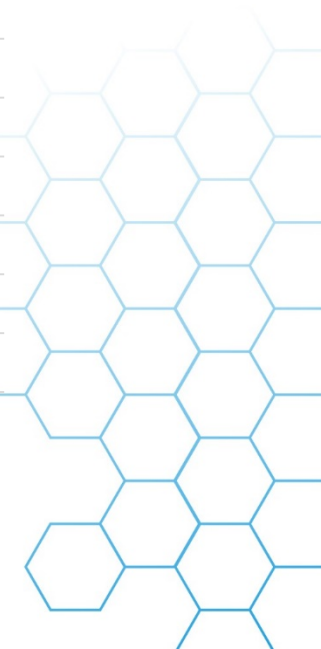
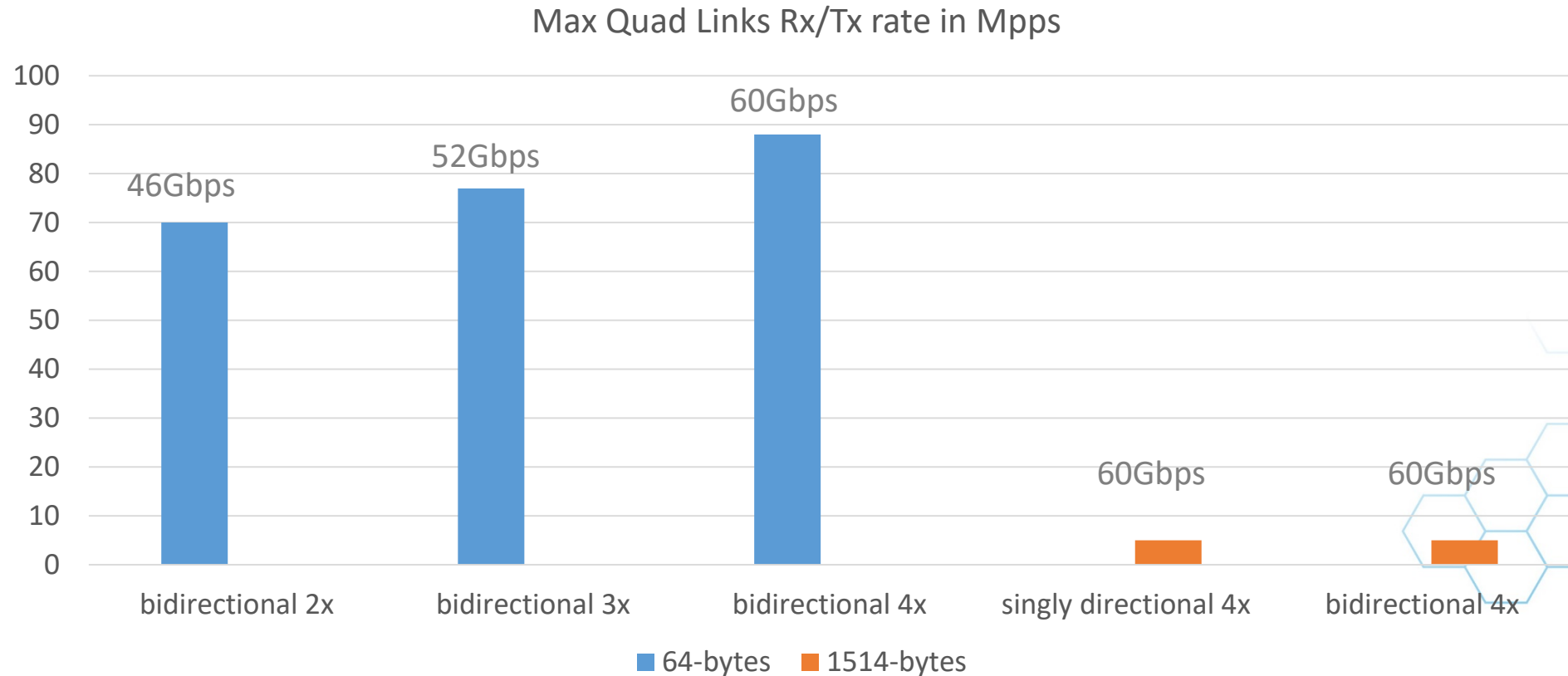


Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz with 20M L3 cache





## Max Quad Links rx/tx rate





## Summary

- Scales with number of links, but not linearly, and constrained by memory bandwidth.
- Two times of memory copy involved, DPDK multi-processes model eliminates it(at the expense of resource segregation).
- Tested with LXC, it should also work with other containers.
- Other virtual device PMD is supposed to meet the same challenges.







# Thanks!!



欢迎关注**DPDK**开源社区

[zhengjie@unitedstack.com](mailto:zhengjie@unitedstack.com)

<https://github.com/chillancezen/dpdk-16.07-vecring>

