Red Hat OpenShift Container Platform on IBM Z & LinuxONE

Capacity Planning: Five Checkmarks You Don‘t Want to Miss

Danijel Soldo
Performance Chapter Lead - OpenShift on Z
_
danijel.soldo@de.ibm.com

Content

CPU Virtualization and Overcommitment Levels on IBM Z

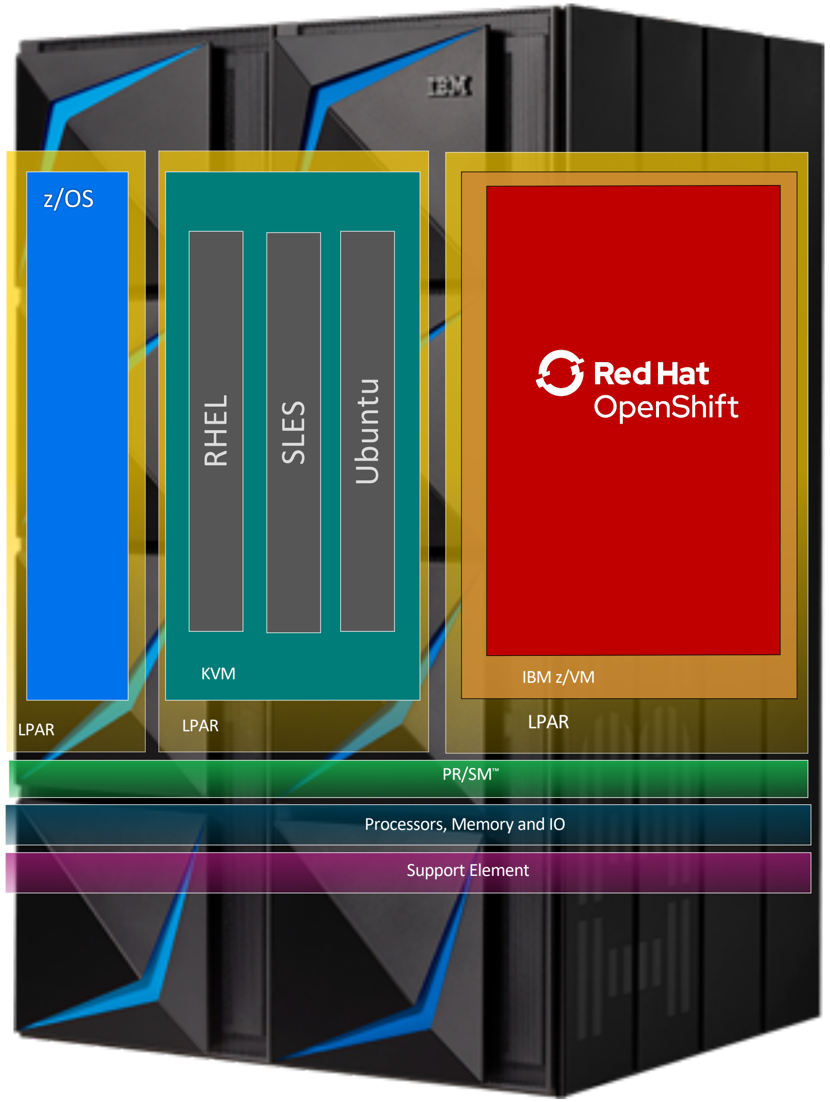
LPAR Weigths & Entitlements

CPU Polarization

Level Up – A hypervisor’s perspective

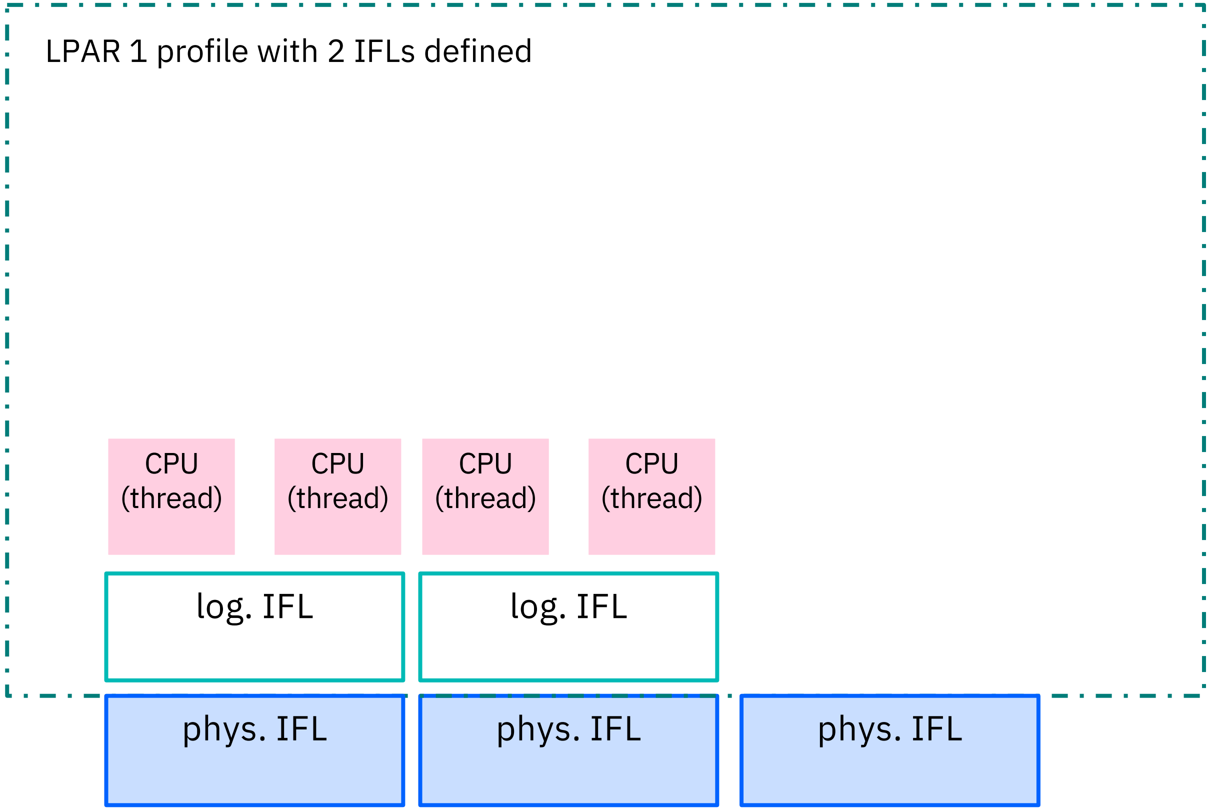
The Five Checkmarks

CPU virtualization & overcommitment levels on Z



- PR/SM grants the highest isolation level via logical partitions (LPAR)
- LPARs are as close to bare-metal as it gets
- Each virtualization layer adds a performance overhead
- Cores can be dedicated or shared
- OpenShift will co-exist and potentially share resources with other LPARs and workloads

CPU virtualization & overcommitment levels on Z



4 CPUs available (SMT-2)

2 logical IFL cores defined (PR/SM)

3 physical IFL cores (*shared*)

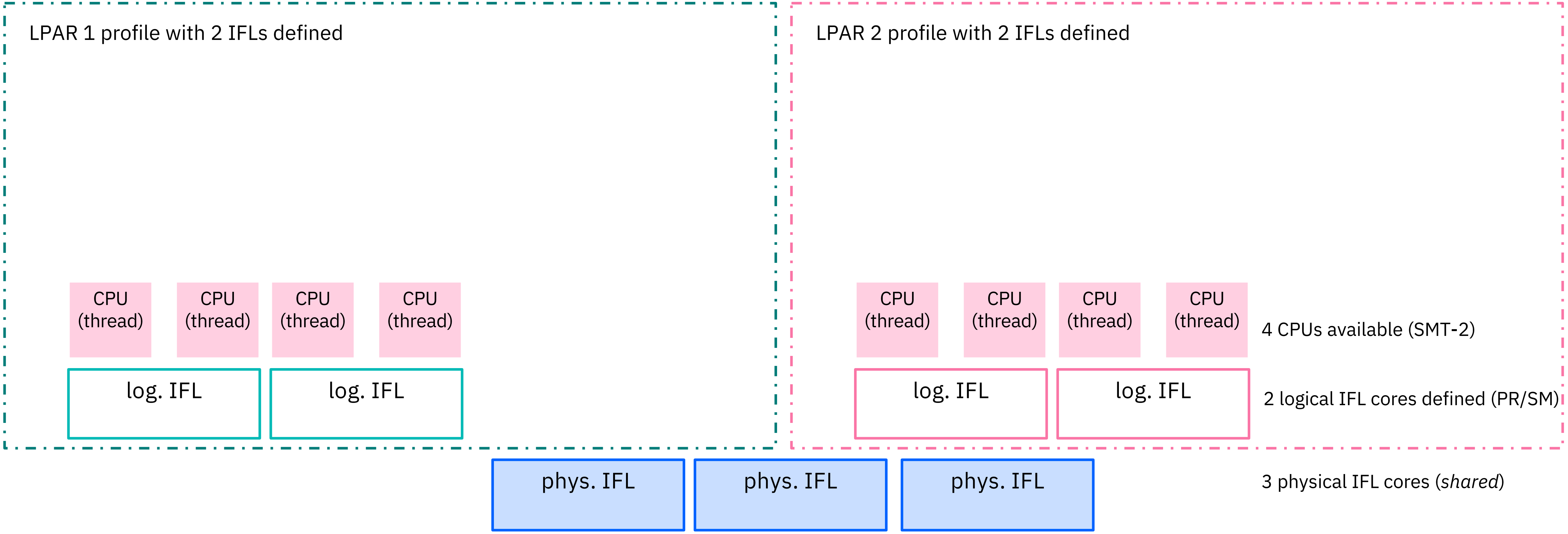
* IFL = Integrated Facility for Linux = Z phys. core

CPU virtualization & overcommitment levels on Z

Resource **overcommitment**:

LPAR shared cores (PR/SM, ratio logical to physical: 4:3)

Good practice - No LPAR can have more IFLs defined than physically available. It is recommended to evaluate the necessary capacity and avoid oversizing LPARs.



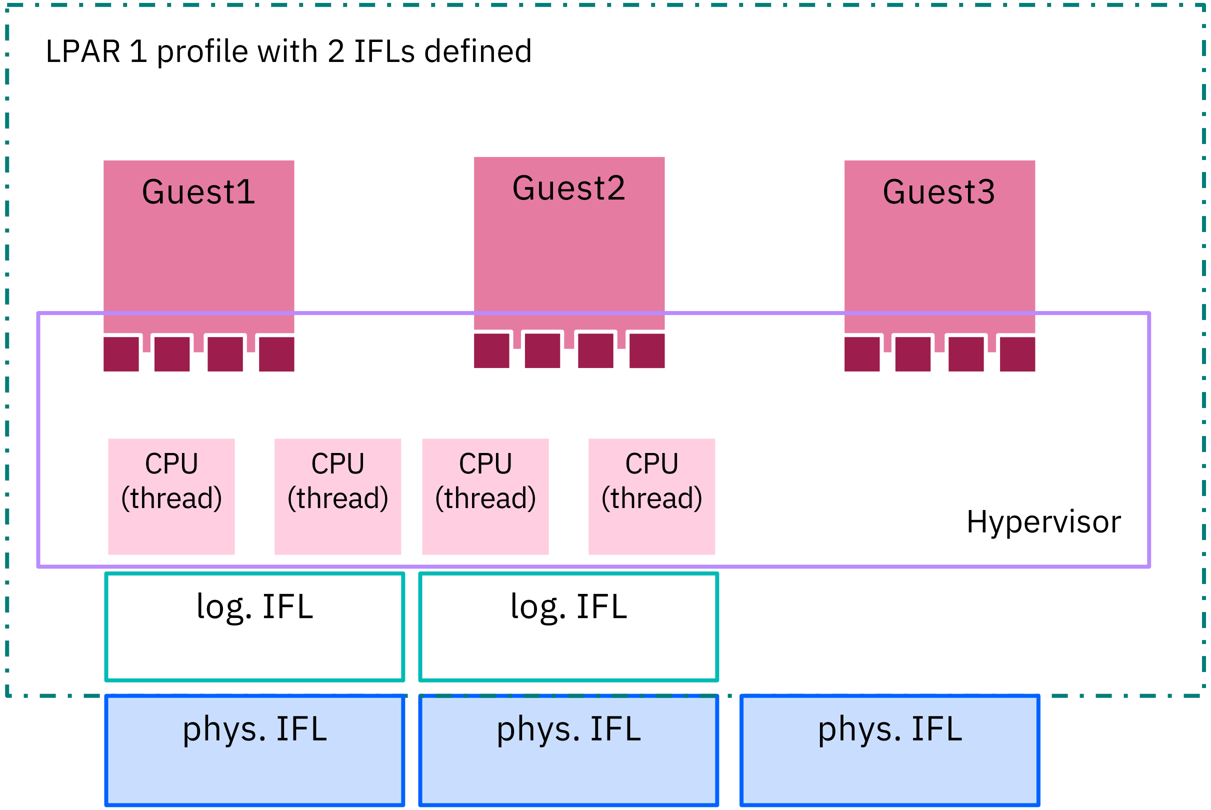
CPU virtualization & overcommitment levels on Z

Resource **overcommitment**:

Hypervisor CPU overcommitment (ratio virtual to CPU: 12:4)

Good practice – no virtual machine should have more vCPUs defined than total CPUs available.

* Hypervisor = z/VM or KVM



12 vCPUs defined (Hypervisor)

4 CPUs available (SMT-2)

2 logical IFL cores defined (PR/SM)

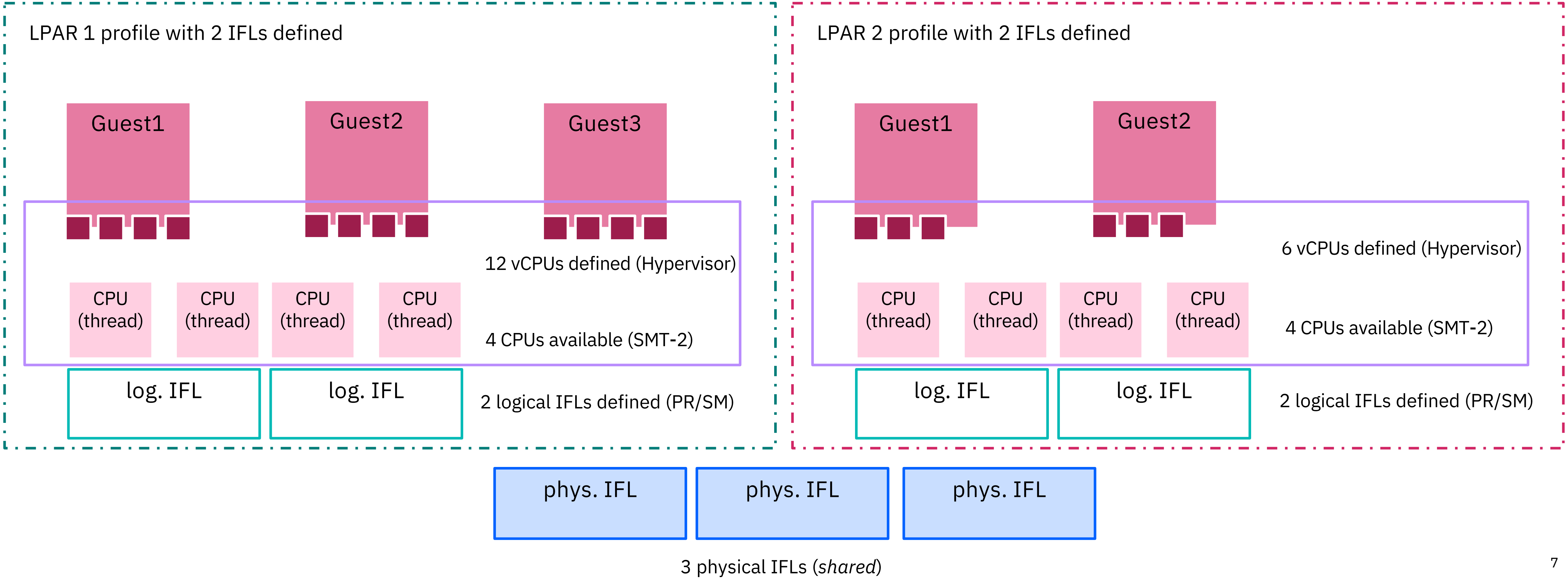
3 physical IFL cores (*shared*)

CPU virtualization & overcommitment levels on Z

Total resource **overcommitment**:

Level 1 - LPAR shared cores (PR/SM, ratio logical to physical: 4:3)
Level 2 - Hypervisor CPU overcommitment (ratio virtual to CPU: 12:4 in LPAR1; 6:4 in LPAR2)

Total: 18 vCPU (5 guests) on 3 physical IFLs



CPU virtualization & overcommitment levels on Z

Total resource **overcommitment**:

Level 1 - LPAR shared cores (PR/SM, ratio logical to physical: 4:3)
Level 2 - Hypervisor CPU overcommitment (ratio virtual to CPU: 12:4; 6:4)

Total: 18 vCPU (5 guests) on 3 physical IFLs

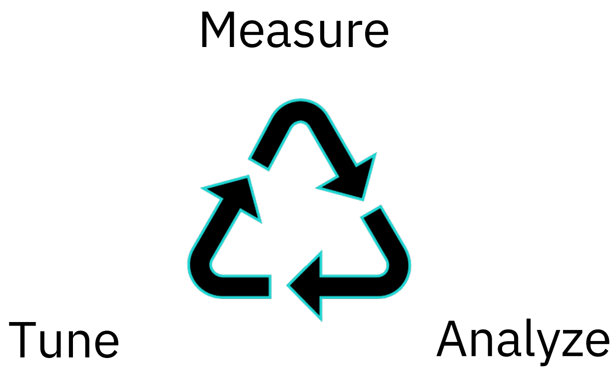
Can this even work?

Not if all vCPUs are constantly fully utilized, or if spikes happen in the exact same timeframe.

Usually a workload type which is suited for some overcommitment:

- **no constant pressure on CPU**
- **spikes are distributed in time**
- **average utilization not high**

Evaluate your workload – always a good idea.



LPAR Weights & Entitlements

Thanks to B. Wade from the z/VM performance team for putting this together:

https://www.vm.ibm.com/library/presentations/lparperf.pdf

IBM

Topics in LPAR Performance

Revision 2020-02-27.1

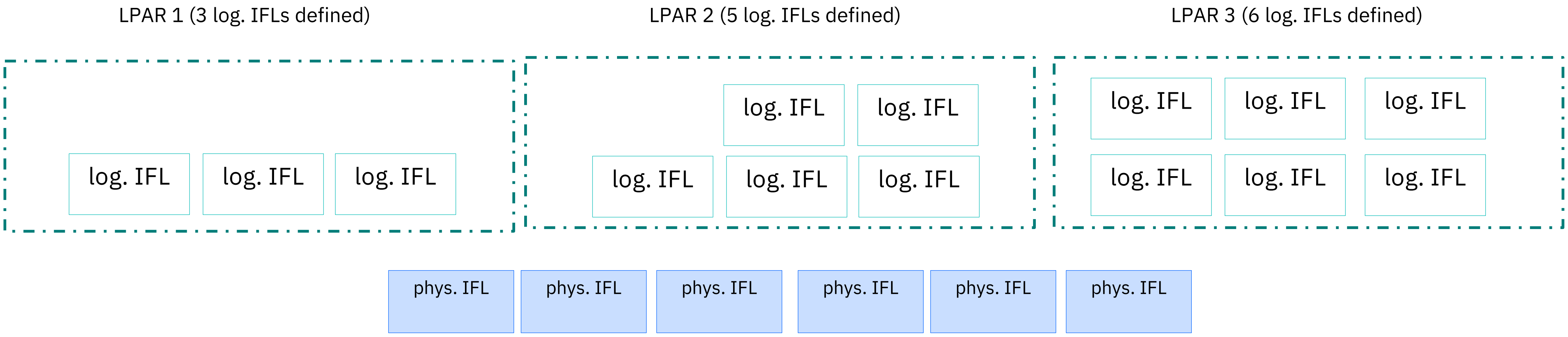
Brian K. Wade, Ph.D.
IBM z/VM Development, Endicott, NY
bkw@us.ibm.com

LPAR Weights & Entitlements

Let’s start with a simple example of 3 logical partitions (LPARs) defined, and a total of 6 physical cores (IFLs) available.

Each LPAR has a certain **amount of cores defined** (logical IFLs), and a **weight value** *which expresses the relative importance in distribution of CPU power*.

Example: 3 LPARs defined, a total of 6 **shared** physical IFLs, 14 logical IFLs defined in total.



LPAR Weights & Entitlements

Resource sharing essentials

A logical core is not a source of capacity. It is a consumer of capacity.
By increasing the LPAR size (defined IFL count), *we don‘t guarantee more power*.

LPAR Entitlement is the minimum power an LPAR can expect to get whenever needed.
Entitlements come into play only when there is not enough power to satisfy all partitions‘ demands.

LPAR Weights & Entitlements

Resource s

Checkmark

#1

Ensure to meet the minimum **physical core** requirements for the cluster setup – 6 IFL cores (SMT-2 enabled) per cluster.

OpenShift and other Kubernetes deployments bring a lot of automation and monitoring capabilities which might be CPU-intensive. Therefore, it is essential to have enough physical core capacity to back the virtualization stack.

Source: https://docs.openshift.com/container-platform/4.9/installing/installing_ibm_z/installing-ibm-z.html#minimum-resource-requirements_installing-ibm-z

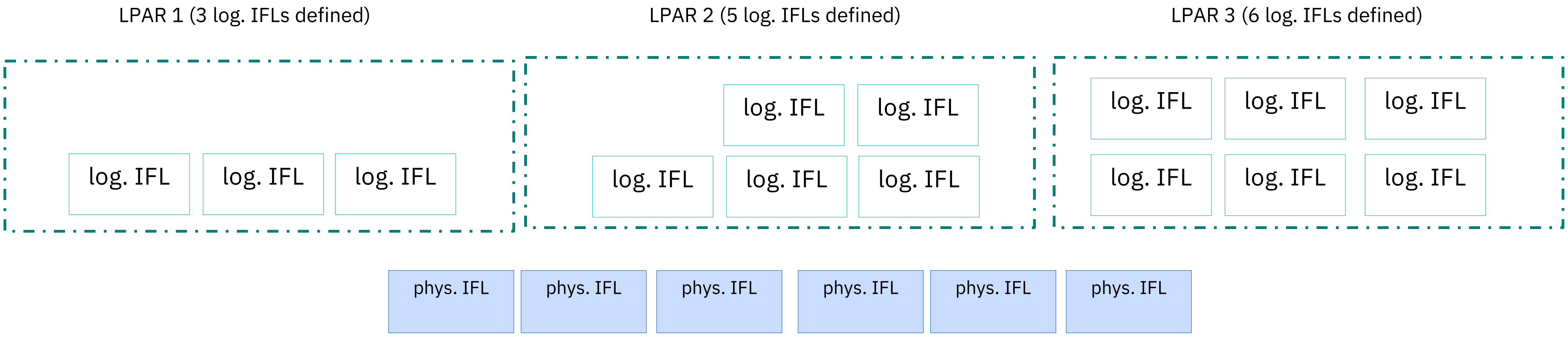
LPAR Weights & Entitlements

PR/SM calculates a table like this:

LPAR	Logical IFLs	Weight	**Entitlement**
LP1	3	10	200
LP2	5	10	200
LP3	6	10	200
SUM	14	30	600
Sum Phys. IFLs	6		

$$Entitlement = \frac{100 \; * SumPhysIFLs \; * Weight}{SumWeights}$$

Initial weight of 10 leads to an entitlement of 200 for each LPAR.
Meaning: **Each LPAR has a guaranteed capacity of 2 physical IFLs.**



Shared IFL cores (Weights are not applicable if using dedicated IFL cores)

LPAR Weights & Entitlements

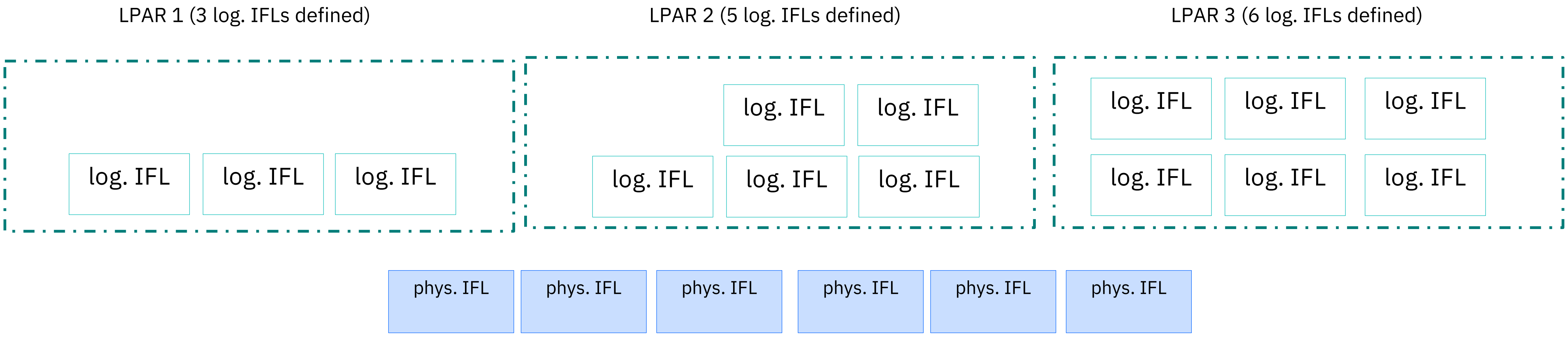
PR/SM calculates a table like this:

LPAR	Logical IFLs	Weight	**Entitlement**
LP1	3	10	200
LP2	5	10	200
LP3	6	10	200
SUM	14	30	600
Sum Phys. IFLs	6		

$$Entitlement = \frac{100 \; * SumPhysIFLs \; * Weight}{SumWeights}$$

Wait, If I give more logical IFLs to an LPAR – **it doesn‘t make any difference to the entitlement?**

It seems so.



Shared IFL cores (Weights are not applicable if using dedicated IFL cores)

LPAR Weights & Entitlements

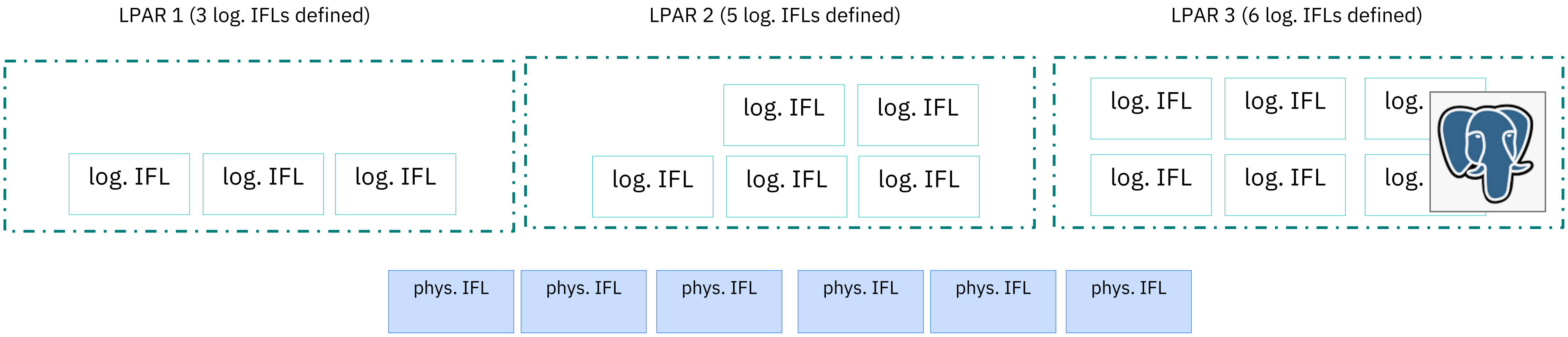
PR/SM calculates a table like this:

LPAR	Logical IFLs	Weight	**Entitlement**
LP1	3	10	200
LP2	5	10	200
LP3	6	10	200
SUM	14	30	600
Sum Phys. IFLs	6		

$$Entitlement = \frac{100 \ * SumPhysIFLs \ * Weight}{SumWeights}$$

Initial weight of 10 leads to an entitlement of 200 for each LPAR.
Meaning: Each LPAR has a guaranteed capacity of 2 physical IFLs.

What if LPAR 3 is running my database and I want it to have 3 IFLs guaranteed?



Shared IFL cores (Weights are not applicable if using dedicated IFL cores)

LPAR Weights & Entitlements

PR/SM calculates a table like this:

LPAR	Logical IFLs	Weight	**Entitlement**
LP1	3	10	100
LP2	5	20	200
LP3	6	30	300
SUM	14	60	600
Sum Phys. IFLs	6		

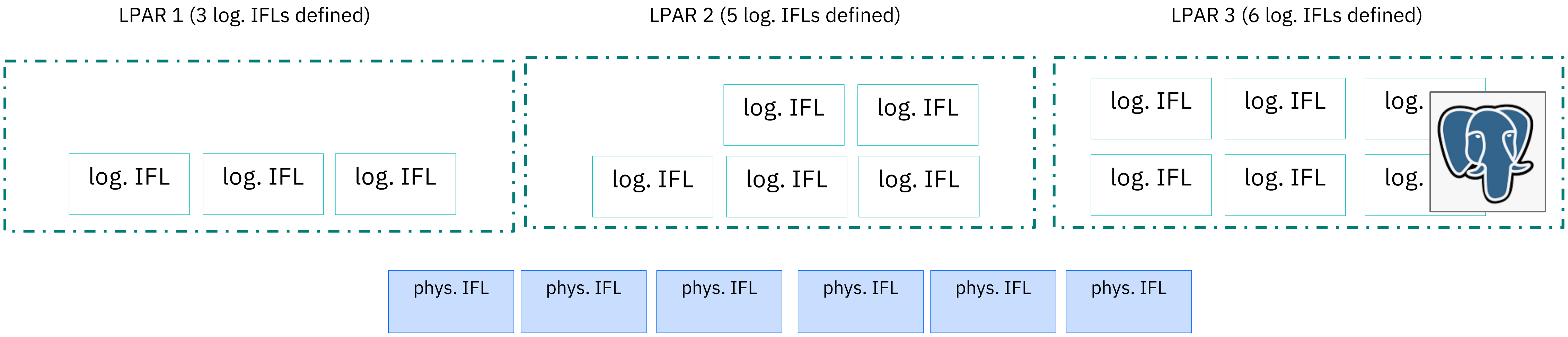
We have 3 options:

1. Add more physical IFLs
2. Dedicate 3 IFLs
3. **Give more weight to the higher priority**

Best practices:

Use weights that sum up to this: (10 x number of shared physical cores)

Now, LPAR 3 is entitled to 3 physical IFLs.



Shared IFL cores (Weights are not applicable if using dedicated IFL cores)

LPAR Weights & Entitlements

How to check your current entitlements?

z/VM Perfkit

FCX306 - Logical Partition Share Screen – LSHARACT

```
FCX306      Data for yyyy/mm/dd  Interval HH:MM:SS - HH:MM:SS    Monitor Scan

LPAR Data, Collected in Partition FCFT

   Core counts:   CP ZAAP  IFL  ICF ZIIP
      Dedicated    8    0    0    0    0
Shared physical   10    2   16    1    3
 Shared logical   79    0   35    0    0 Unused physical core(s) detected

____ .             .     .      .          .   .        . .           .      .      . .
Proc Partition  Core   Load   LPAR                                       <CoreTotal,%> Core
Type Name       Count   Max Weight Entlment Cap TypeCap GrpCapNm GrpCap   Busy Excess Conf
...  PLB1         ...   ...      0      ... ...     ... ...         ...    ...    ... .
CP   ECPX3          3   300     10     31.3 No      ... SAMPLE   1200.0    5.8     .0 o
CP   EEXT1          4   400     10     31.3 No      ... SAMPLE   1200.0    1.2     .0 o
CP   EPAT          10  1000     10     31.3 No      ... SAMPLE   1200.0    5.5     .0 o
CP   EPLX1          6   600     60    187.5 No      ... ...         ...  326.3  138.8 o
CP   EPLX2          8   800     45    140.6 No      ... ...         ...  256.0  115.4 o
CP   EPLX3          6   600     45    140.6 No      ... ...         ...  233.3   92.7 o
CP   EPRF1          4   400    DED    400.0 No      ... ...         ...  399.8     .0 -
CP   EPRF2          4   400    DED    400.0 No      ... ...         ...     .0     .0 -
CP   ESTL1          7   700     50    156.3 No      ... ...         ...    1.5     .0 o
CP   EST1           8   800     10     31.3 No      ... ...         ...    8.7     .0 o
CP   EST2           6   600     10     31.3 No      ... ...         ...    1.3     .0 o
CP   EVIC           2   200     10     31.3 No      ... ...         ...     .0     .0 o
CP   FCFT           8   800     40    125.0 No      ... ...         ...  117.4     .0 o
CP   K4             6   600     10     31.3 No      ... ...         ...    5.4     .0 o
CP   PHOS           5   500     10     31.3 No      ... ...         ...     .7     .0 o
IFL  EEXT2         16  1600     10    200.0 No      ... ...         ...    1.1     .0 o
IFL  EPLX1          3   300     60   1200.0 No      ... ...         ...    2.5     .0 u
IFL  EST3          16  1600     10    200.0 No      ... ...         ...     .0     .0 o
```

HMC

Select machine -> Operational Customization -> Change LPAR Controls

IBM **Hardware Management Console**

Home | Change LPAR Controls - ...

Change Logical Partition Controls - T311

Last reset profile attempted:
Input/output configuration data set (IOCDS):A0 328AT311

CPs | **IFLs** | Processor Running Time

Logical Partitions with Integrated Facility for Linux Processors

Logical Partition	Active	Defined Capacity	WLM	Current Weight	Initial Weight	Min Weight	Max Weight	Current Capping	Initial Capping	Absolute Capping	Number of Dedicated Processors	Number of Not dedicated Processors
T311LP01	Yes	0	☐	10	10			No	☐	None	0	32
T311LP02	Yes	0	☐	10	10			No	☐	None	0	4
T311LP03	Yes	0	☐	10	10			No	☐	None	0	4
T311LP04	No	0	☐	0	10			No	☐	None	0	4
T311LP05	No	0	☐	0	10			No	☐	None	0	4
T311LP06	No	0	☐	0	0			No	☐	None	1	0
T311LP07	No	0	☐	0	10			No	☐	None	0	1

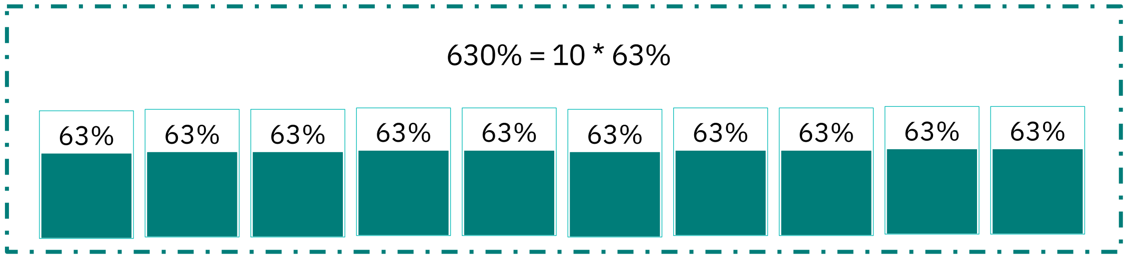
1. Only active LPARs count.
2. Weights do not apply to dedicated IFLs.
3. Separated by processor type (IFL, CP, SAP ...)
4. **Entitlements not visible** (need to calculate them on your own)

CPU Polarization
Vertical vs Horizontal

LPAR entitlement: **630%**

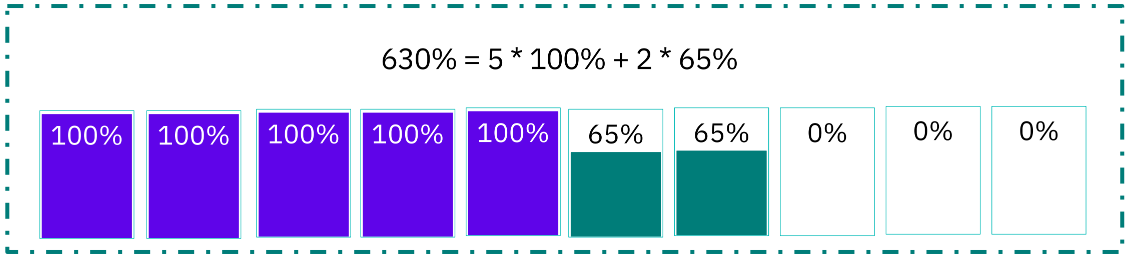
Horizontal polarization
- equal distribution over log. cores
- 63% guaranteed per core
- up to 100% possible

* disables SMT-2 capabilities under z/VM



IBM Z LPAR – 10 logical IFLs defined

Vertical polarization
- unequal distribution over log. cores
- optimal for performance reasons

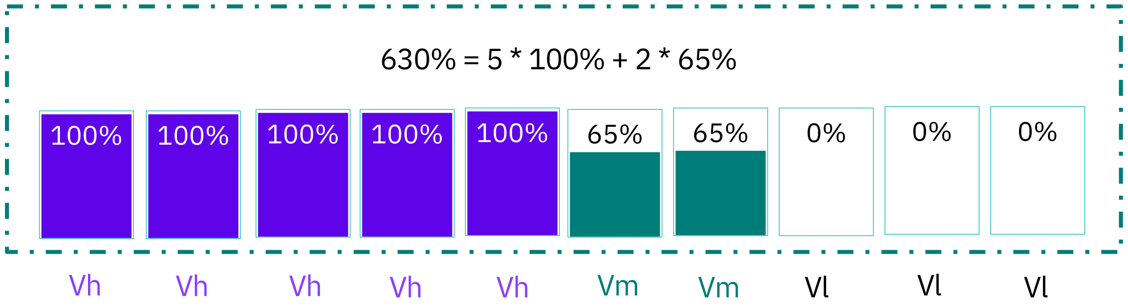


IBM Z LPAR – 10 logical IFLs defined

source: https://www.vm.ibm.com/perf/tips/zvmhd.html

CPU Polarization
Vertical vs Horizontal

Vertical polarization
- unequal distribution over log. cores
- recommended and used by default



IBM Z LPAR –
10 logical IFLs
defined

Vh – Vertical High	logical core with 100% entitlement, used exclusively
Vm – Vertical Medium	logical core with < 100% entitlement, used shared
Vl – Vertical Low	logical core with 0% entitlement, used shared

Good to know:
- The goal is to have as much Vh cores as possible
- PR/SM avoids having Vm below 50% (*630 could have been 6 * Vh + 1 * Vm@30)*

CPU Polarization
How to check it quickly – z/VM

FCX298 Logical Core Organization Log Screen – PUORGLOG

```
FCX298      CPU nnnn  SER nnnnn  Interval HH:MM:SS - HH:MM:SS    Perf. Monitor

Logical Core organization for Partition FCFT     (GDLFCFT )

Date  Time      Core Type PPD Ent. Location
04/11 14:53:55   00 CP   Vh    ... 4:2
04/11 14:53:55   01 CP   Vh    ... 4:2
04/11 14:53:55   02 CP   Vh    ... 4:2
04/11 14:53:55   03 CP   Vh    ... 4:2
04/11 14:53:55   04 CP   Vh    ... 4:2
04/11 14:53:55   05 CP   Vm    ... 4:1
04/11 14:53:55   06 CP   Vm    ... 4:1
04/11 14:53:55   07 CP   Vl    ... 4:5
04/11 14:53:55   08 CP   Vl    ... 4:5
04/11 14:53:55   09 CP   Vl    ... 4:6
04/11 14:53:55   0A CP   Vl    ... 4:6
04/11 14:53:55   0B CP   Vl    ... 4:6
04/11 14:53:55   0C ZAAP Vm    ... 3:1
04/11 14:53:55   0D ZIIP Vm    ... 3:1
04/11 14:53:55   0E IFL  Vm    ... 4:5
04/11 14:53:55   0F IFL  Vl    ... 4:5
04/11 14:53:55   10 IFL  Vl    ... 4:6
04/11 14:53:55   11 IFL  Vl    ... 4:6
```

FCX304 Processor Log Screen – PRCLOG

```
1FCX304  Run 2019/03/20 13:15:32          PRCLOG
                                          Processor Activity, by Time
 From 2019/03/11 06:52:16
 To   2019/03/11 07:12:16
 For   1200 Secs 00:20:00                 Result of 2U0C021D Run
 ____________________________________________________________________

                                               <--- Percent Busy ---->
           C                     Pct
 Interval  P                     Park
 End Time  U Type PPD Ent. DVID  Time %Susp Total  User  Syst  Emul
 >>Mean>>  0 IFL  Vh   100 0000     0   1.6  94.6  82.7  11.9  73.3
 >>Mean>>  1 IFL  Vh   100 0001     0   1.0  96.0  87.0   9.0  79.7
 >>Mean>>  2 IFL  Vh   100 0002     0    .9  96.7  88.0   8.7  82.2
 >>Mean>>  3 IFL  Vh   100 0003     0    .9  96.5  87.8   8.7  81.6
```

CPU Polarization
How to check it quickly - KVM

Default: Horizontal

```
lscpu -e
```

CPU	NODE	DRAWER	BOOK	SOCKET	CORE	L1d:L1i:L2d:L2i	ONLINE	CONFIGURED	POLARIZATION	ADDRESS
0	0	0	0	0	0	0:0:0:0	yes	yes	horizontal	0
1	0	0	0	0	0	1:1:1:1	yes	yes	horizontal	1
2	0	0	0	0	1	2:2:2:2	yes	yes	horizontal	2
3	0	0	0	0	1	3:3:3:3	yes	yes	horizontal	3
4	0	0	1	1	2	4:4:4:4	yes	yes	horizontal	4
5	0	0	1	1	2	5:5:5:5	yes	yes	horizontal	5
6	0	0	1	2	3	6:6:6:6	yes	yes	horizontal	6
7	0	0	1	2	3	7:7:7:7	yes	yes	horizontal	7
8	0	0	1	2	4	8:8:8:8	yes	yes	horizontal	8
9	0	0	1	2	4	9:9:9:9	yes	yes	horizontal	9
10	0	0	1	2	5	10:10:10:10	yes	yes	horizontal	10
11	0	0	1	2	5	11:11:11:11	yes	yes	horizontal	11
12	0	0	1	2	6	12:12:12:12	yes	yes	horizontal	12
13	0	0	1	2	6	13:13:13:13	yes	yes	horizontal	13
14	0	0	1	2	7	14:14:14:14	yes	yes	horizontal	14
15	0	0	1	2	7	15:15:15:15	yes	yes	horizontal	15
16	0	0	1	2	8	16:16:16:16	yes	yes	horizontal	16
17	0	0	1	2	8	17:17:17:17	yes	yes	horizontal	17
18	0	0	1	1	9	18:18:18:18	yes	yes	horizontal	18
19	0	0	1	1	9	19:19:19:19	yes	yes	horizontal	19
20	0	0	1	1	10	20:20:20:20	yes	yes	horizontal	20
21	0	0	1	1	10	21:21:21:21	yes	yes	horizontal	21
22	0	0	1	1	11	22:22:22:22	yes	yes	horizontal	22
23	0	0	1	1	11	23:23:23:23	yes	yes	horizontal	23
24	0	0	1	1	12	24:24:24:24	yes	yes	horizontal	24
25	0	0	1	1	12	25:25:25:25	yes	yes	horizontal	25
26	0	0	1	1	13	26:26:26:26	yes	yes	horizontal	26
27	0	0	1	1	13	27:27:27:27	yes	yes	horizontal	27
28	0	0	1	1	14	28:28:28:28	yes	yes	horizontal	28
29	0	0	1	1	14	29:29:29:29	yes	yes	horizontal	29
30	0	0	1	2	15	30:30:30:30	yes	yes	horizontal	30
31	0	0	1	2	15	31:31:31:31	yes	yes	horizontal	31

Switching to vertical with:

```
chcpu -p vertical
```

Not recommended on s390x. Might cause severe performance degradations.

However, shortly switching over gives a good impression on the current entitlement.

CPU	NODE	DRAWER	BOOK	SOCKET	CORE	L1d:L1i:L2d:L2i	ONLINE	CONFIGURED	POLARIZATION	ADDRESS
0	0	0	0	0	0	0:0:0:0	yes	yes	vert-medium	0
1	0	0	0	0	0	1:1:1:1	yes	yes	vert-medium	1
2	0	0	0	0	1	2:2:2:2	yes	yes	vert-low	2
3	0	0	0	0	1	3:3:3:3	yes	yes	vert-low	3
4	0	0	1	1	2	4:4:4:4	yes	yes	vert-low	4
5	0	0	1	1	2	5:5:5:5	yes	yes	vert-low	5
6	0	0	1	1	3	6:6:6:6	yes	yes	vert-low	6
7	0	0	1	1	3	7:7:7:7	yes	yes	vert-low	7
8	0	0	1	1	4	8:8:8:8	yes	yes	vert-low	8
9	0	0	1	1	4	9:9:9:9	yes	yes	vert-low	9
10	0	0	1	1	5	10:10:10:10	yes	yes	vert-low	10
11	0	0	1	1	5	11:11:11:11	yes	yes	vert-low	11
12	0	0	1	1	6	12:12:12:12	yes	yes	vert-low	12
13	0	0	1	1	6	13:13:13:13	yes	yes	vert-low	13
14	0	0	1	1	7	14:14:14:14	yes	yes	vert-low	14
15									vert-low	15
16									vert-low	16
17									vert-low	17
18									vert-low	18
19	0	0	1	2	9	19:19:19:19	yes	yes	vert-low	19
20	0	0	1	2	10	20:20:20:20	yes	yes	vert-low	20
21	0	0	1	2	10	21:21:21:21	yes	yes	vert-low	21
22	0	0	1	2	11	22:22:22:22	yes	yes	vert-low	22
23	0	0	1	2	11	23:23:23:23	yes	yes	vert-low	23
24	0	0	1	2	12	24:24:24:24	yes	yes	vert-low	24
25	0	0	1	2	12	25:25:25:25	yes	yes	vert-low	25
26	0	0	1	2	13	26:26:26:26	yes	yes	vert-low	26
27	0	0	1	2	13	27:27:27:27	yes	yes	vert-low	27
28	0	0	1	2	14	28:28:28:28	yes	yes	vert-low	28
29	0	0	1	2	14	29:29:29:29	yes	yes	vert-low	29
30	0	0	1	2	15	30:30:30:30	yes	yes	vert-low	30
31	0	0	1	2	15	31:31:31:31	yes	yes	vert-low	31

Only 2 vert-mediums – clearly not a good example.

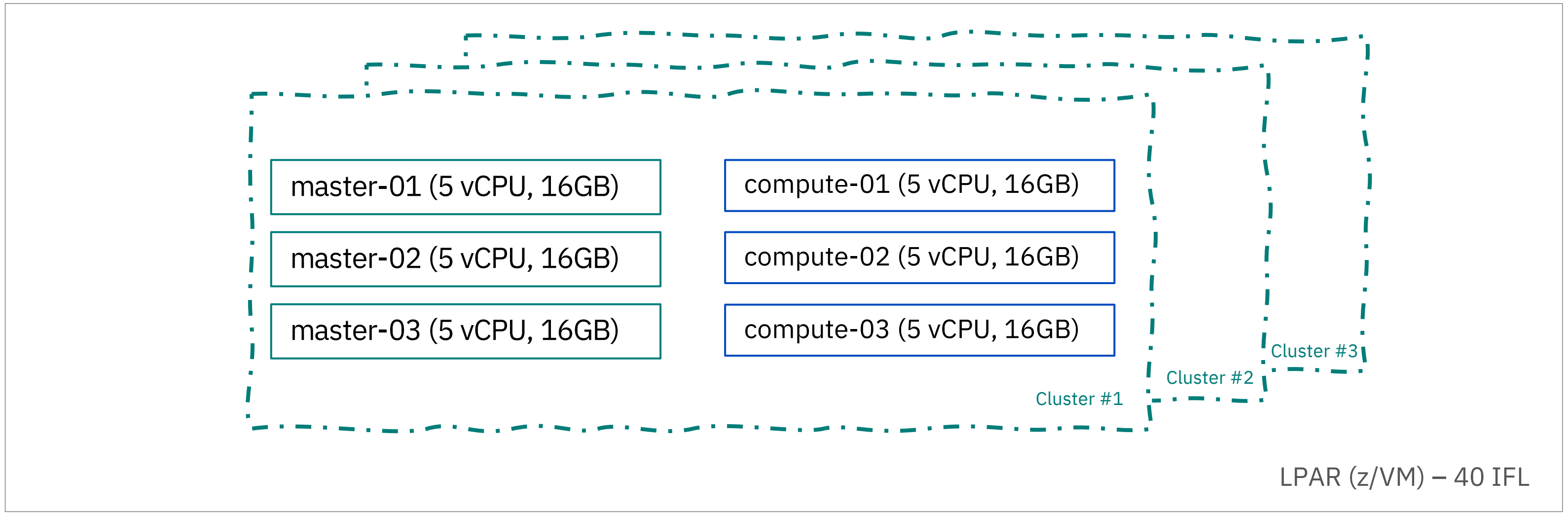
LPAR Weigths, Entitlements & CPU Polarization

How can this become a **problem** for OpenShift on Z?

Make sure your cluster survives
Real example (September '21)

Environment: 3 x OCP Cluster (6 nodes each); 40 logical IFLs defined (shared), z14

Symptoms: very slow response at the console, pods crashing (CRI-O, monitoring), high steal CPU% in the nodes



Before – Cluster #1

```
[root@bastion ~]# oc adm top node
NAME                           CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
master-01.ocp-cluster.example.tmp    4789m        106%    12768Mi         85%
master-02.ocp-cluster.example.tmp    3495m        77%     10609Mi         70%
master-03.ocp-cluster.example.tmp    3527m        78%     9969Mi          66%
compute-01.ocp-cluster.example.tmp   3343m        74%     11078Mi         46%
compute-02.ocp-cluster.example.tmp    821m         18%     3412Mi          14%
compute-03.ocp-cluster.example.tmp   2890m        64%     7397Mi          30%
```

No workload deployed.
Consuming almost 12 vCPUs just for the Control Plane?

Logging into the master nodes shows up to ~70% steal CPU%.
(OCP Monitoring adds steal% to the CPU (cores) consumption of nodes.)

Make sure your cluster survives
Real example (September ‘21)

Ok, but who/what is *stealing* the CPU?

- Cluster #2 and #3 do not run any workload and show a very similar consumption pattern
- The whole LPAR consumes little less than 10 IFLs (cluster logging stack installed)
- There is 40 IFLs defined for the LPAR, should be enough for everyone

Maybe other LPARs?

CEC Setup:

Total physical IFLs: 141 (shared)
Total logical IFLs: 1116 (*8x more*)

Make sure your cluster survives

Real example (September ‘21)

Checking the z/VM Perfkit to get an overview of all LPAR weigths and understand what entitlement does my LPAR have:

FCX306 Logical Partition Share (**ZVMHOSTLP**)

Type	Name	Count	Max	**Weight**	Entlment	Cap	TypeCap	GrpCapNm
IFL	TESTLP	40	4000	**10**	184.2	No	...	...

FCX298 Logical Core organization log

Date	Time	Core	Type	PPD	Ent.	Location
Aug 25	16:58:18	0	IFL	Vh	...	05:01
Aug 25	16:58:18	1	IFL	Vm	...	05:01
Aug 25	16:58:18	2	IFL	Vl	...	05:01
Aug 25	16:58:18	3	IFL	Vl	...	05:01
Aug 25	16:58:18	4	IFL	Vl	...	05:01
Aug 25	16:58:18	5	IFL	Vl	...	05:01
Aug 25	16:58:18	6	IFL	Vl	...	05:01
Aug 25	16:58:18	7	IFL	Vl	...	05:03
Aug 25	16:58:18	8	IFL	Vl	...	05:02
Aug 25	16:58:18	9	IFL	Vl	...	05:02
Aug 25	16:58:18	0A	IFL	Vl	...	05:02
Aug 25	16:58:18	0B	IFL	Vl	...	05:02
Aug 25	16:58:18	0C	IFL	Vl	...	05:02
Aug 25	16:58:18	0D	IFL	Vl	...	05:02
Aug 25	16:58:18	0E	IFL	Vl	...	05:02
Aug 25	16:58:18	0F	IFL	Vl	...	05:02
Aug 25	16:58:18	10	IFL	Vl	...	05:02
Aug 25	16:58:18	11	IFL	Vl	...	05:03
Aug 25	16:58:18	12	IFL	Vl	...	05:03
Aug 25	16:58:18	13	IFL	Vl	...	05:03
Aug 25	16:58:18	14	IFL	Vl	...	05:03
Aug 25	16:58:18	15	IFL	Vl	...	05:03
Aug 25	16:58:18	16	IFL	Vl	...	05:03
Aug 25	16:58:18	17	IFL	Vl	...	05:03
Aug 25	16:58:18	18	IFL	Vl	...	06:01
Aug 25	16:58:18	19	IFL	Vl	...	06:01
Aug 25	16:58:18	1A	IFL	Vl	...	06:01
Aug 25	16:58:18	1B	IFL	Vl	...	06:01
Aug 25	16:58:18	1C	IFL	Vl	...	06:01
Aug 25	16:58:18	1D	IFL	Vl	...	06:01
Aug 25	16:58:18	1E	IFL	Vl	...	06:02
Aug 25	16:58:18	1F	IFL	Vl	...	06:02
Aug 25	16:58:18	20	IFL	Vl	...	06:02
Aug 25	16:58:18	21	IFL	Vl	...	06:02
Aug 25	16:58:18	22	IFL	Vl	...	06:02
Aug 25	16:58:18	23	IFL	Vl	...	06:02
Aug 25	16:58:18	24	IFL	Vl	...	01:01
Aug 25	16:58:18	25	IFL	Vl	...	01:01
Aug 25	16:58:18	26	IFL	Vl	...	02:01
Aug 25	16:58:18	27	IFL	Vl	...	02:01

Make sure your cluster survives
Real example (September '21)

Checking the z/VM Perfkit to get an overview of all LPAR weigths and understand what entitlement does my LPAR have:

FCX306 Logical Partition Share (**ZVMHOSTLP**)

Type	Name	Count	Max	**Weight**	Entlment	Cap	TypeCap	GrpCapNm
IFL	TESTLP	40	4000	**10**	184.2	No	...	...

The *whole LPAR* is guaranteed to get 1.84 IFLs! If it wants more, it needs to compete against other LPARs on the machine.

Running 3 OCP clusters, it would be necessary to see **at least 6 IFLs guaranteed (entitlement > 600)**.

FCX298 Logical Core organization log

Date	Time	Core	Type	PPD	Ent.	Location
Aug 25	16:58:18	0	IFL	**Vh**	...	05:01
Aug 25	16:58:18	1	IFL	**Vm**	...	05:01
Aug 25	16:58:18	2	IFL	**Vl**	...	05:01
Aug 25	16:58:18	3	IFL	**Vl**	...	05:01
Aug 25	16:58:18	4	IFL	**Vl**	...	05:01
Aug 25	16:58:18	5	IFL	**Vl**	...	05:01
Aug 25	16:58:18	6	IFL	**Vl**	...	05:01
Aug 25	16:58:18	7	IFL	**Vl**	...	05:03
Aug 25	16:58:18	8	IFL	**Vl**	...	05:02
Aug 25	16:58:18	9	IFL	**Vl**	...	05:02
Aug 25	16:58:18	15	IFL	**Vl**	...	05:03
Aug 25	16:58:18	16	IFL	**Vl**	...	05:03
Aug 25	16:58:18	17	IFL	**Vl**	...	05:03
Aug 25	16:58:18	18	IFL	**Vl**	...	06:01
Aug 25	16:58:18	19	IFL	**Vl**	...	06:01
Aug 25	16:58:18	1A	IFL	**Vl**	...	06:01
Aug 25	16:58:18	1B	IFL	**Vl**	...	06:01
Aug 25	16:58:18	1C	IFL	Vl	...	06:01
Aug 25	16:58:18	1D	IFL	Vl	...	06:01
Aug 25	16:58:18	1E	IFL	Vl	...	06:02
Aug 25	16:58:18	1F	IFL	Vl	...	06:02
Aug 25	16:58:18	20	IFL	Vl	...	06:02
Aug 25	16:58:18	21	IFL	Vl	...	06:02
Aug 25	16:58:18	22	IFL	Vl	...	06:02
Aug 25	16:58:18	23	IFL	Vl	...	06:02
Aug 25	16:58:18	24	IFL	Vl	...	01:01
Aug 25	16:58:18	25	IFL	Vl	...	01:01
Aug 25	16:58:18	26	IFL	Vl	...	02:01
Aug 25	16:58:18	27	IFL	Vl	...	02:01

1 * Vh @ 100

1 * Vm @ 84

38 * Vl @ 0

Make sure your cluster survives
Real example (September ‘21)

Let‘s quickly increase the weight to 100.

LPAR Entitlement jumped to **1489 -> almost 15 IFLs guaranteed.** Checking the cluster now:

Before

```
[root@bastion ~]# oc adm top node
NAME                               CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
master-01.ocp-cluster.example.tmp   4789m       106%    12768Mi         85%
master-02.ocp-cluster.example.tmp   3495m       77%     10609Mi         70%
master-03.ocp-cluster.example.tmp   3527m       78%     9969Mi          66%
compute-01.ocp-cluster.example.tmp  3343m       74%     11078Mi         46%
compute-02.ocp-cluster.example.tmp   821m        18%     3412Mi          14%
compute-03.ocp-cluster.example.tmp  2890m       64%     7397Mi          30%
```

After

```
[root@bastion ~]# oc adm top node
NAME                               CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
master-01.ocp-cluster.example.tmp   1520m       33%    8677Mi          57%
master-02.ocp-cluster.example.tmp   1079m       23%    7773Mi          51%
master-03.ocp-cluster.example.tmp    736m       16%    6484Mi          43%
compute-01.ocp-cluster.example.tmp  1425m       31%    13945Mi         58%
compute-02.ocp-cluster.example.tmp  1237m       27%    11438Mi         47%
compute-03.ocp-cluster.example.tmp   954m       21%    11257Mi         46%
```

Result:

- 10x less steal CPU%
- 4x reduced control plane consumption
- no pods crashing, smooth response times

Make sure your cluster survives
Real example (September ‘21)

However, this is still not optimal. The difference between logical IFLs and entitlement should not be this big (40 vs 15).

We need to evaluate the workload and discuss the sizing.

For OpenShift on Z, it is essential to understand the minimum required resources for vital cluster operations:

at least 2 IFL of capacity should be guaranteed per cluster.

For a single-LPAR cluster we should ensure an entitlement of **at least 200**.

For multi-LPAR clusters, entitlements per LPAR should be **at least 100.**

What is consuming resources if no workload is deployed?
Replay available externally: OpenShift on Z - CPU Consumption Demystified

Make sure your cluster survives

Real exan

However, th

We need to

For OpenSh

at least 2 IF

For a single-

For multi-LF

Checkmark

#2

Ensure **each cluster is entitled to get sufficient capacity** for system operations – at least 2 IFL cores per LPAR in a single-LPAR cluster, or at least 1 IFL core per LPAR in a multi-LPAR deployment.

Resource sharing is one of the key strengths of the platform – make sure to understand the principles of LPAR weights, entitlements and CPU polarization. If the LPAR is not able to get enough computing resources because of low entitlement – **no tuning on higher levels will help.**

Level Up – A Hypervisor’s Perspective

There’s never enough virtualization.

Level Up – z/VM perspective

With z/VM, a virtual machine receives its proportion of processor time according to its SHARE setting.

When demand for CPU resources is larger than available resources, virtual machines will have to wait to get their share of the CPU.
There are two types of shares:

- Absolute Share
- Relative Share

With relative shares, we express the importance of a virtual machine in relation to the others. The idea is very similar to the concept of LPAR weights.

To display a virtual machine's current share setting, enter:

```
# query share userid
```

To assign a normal relative share of 300 to the virtual machine of user USER1, enter:

```
# set share user1 relative 300
```

Level Up – z/VM perspective

How to check it quickly – relative shares z/VM

User Configuration Screen (**FCX226**) shows virtual machine configuration information for each user

```
FCX226        CPU nnnn  SER nnnnn        Status   HH:MM:SS        Perf. Monitor
________      .   .    .    .   .       .     .         . .    .     .    .    .    .     .     .    .
                                   No   Atta-
                   Mach Flg Qck MDC  ched   Stor Reserved <---- Virt. CPUs ----> <---- Share ------> CPU
 Userid       SVM Mode ReO DSP Fair XSTOR   Size    Pages Type Aff Def. Ded. Stop Share Limit MaxSh. Pool
 CFT2ND       No  ESA  Off Off No       0   256M        0 CP   Off    3    0    0   100 ...    ...   ...
 CFT2NDA      No  EME  Off Off No       0  1024M        0 CP   On     1    0    0   100 ...    ...   GROUP2ND
 DTCVSW1      Yes ESA  Off On  No       0    32M        0 CP   On     1    0    0   100 ...    ...   GROUP2ND
 GCS          No  ESA  Off Off No       0    16M        0 CP   On     1    0    0   100 ...    ...   ...
 MISCSERV     No  ESA  Off Off No       0    64M        0 CP   On     1    0    0   100 ...    ...   ...
 MONWRITE     No  ESA  Off On  No       0     4M      512 CP   On     1    0    0  3.0% Hard   6.0%  ...
 RSTL3        No  ESA  Off On  No       0   512M        0 CP   On     1    0    0  3.0% Soft   6.0%  ...
 RXAGENT1     Yes ESA  Off On  No       0    32M        0 CP   On     1    0    0   100 ...    ...   ...
 SFSFCFT      Yes XC   Off On  Yes      0    64M     8192 CP   On     1    0    0  1500 ...    ...   ...
 VMNFS        Yes ESA  Off On  No       0    64M        0 CP   On     1    0    0   100 ...    ...   ...
 VMSERVP      Yes XC   Off On  Yes      0    64M        0 CP   On     1    0    0  1500 ...    ...   SFSGROUP
 VMSERVR      Yes ESA  Off On  No       0    32M        0 CP   On     1    0    0  1500 ...    ...   SFSGROUP
 VMSERVS      Yes XC   Off On  Yes      0    64M        0 CP   On     1    0    0  1500 ...    ...   SFSGROUP
 VMSERVU      Yes XC   Off On  Yes      0    32M        0 CP   On     1    0    0  1500 ...    ...   SFSGROUP
 YVETTE       No  ESA  Off Off No       0    17M        0 CP   On     1    0    0   100 ...    ...   ...
 2AXTEST      No  ESA  Off Off No       0   128M        0 CP   On     1    0    0   100 ...    ...   ...

 Select a user for user details
 Command ===> _
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

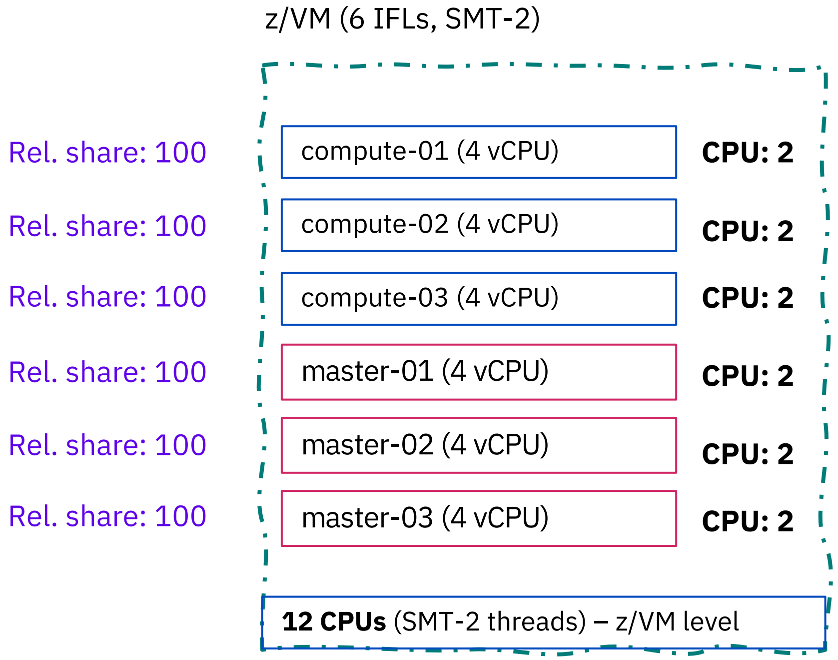
source: Performance Toolkit Reference

Level Up – z/VM perspective

What does it mean for OpenShift on Z?

Default SHARE settings are:

- 100 per guest (normal share)
- nolimit (max share)



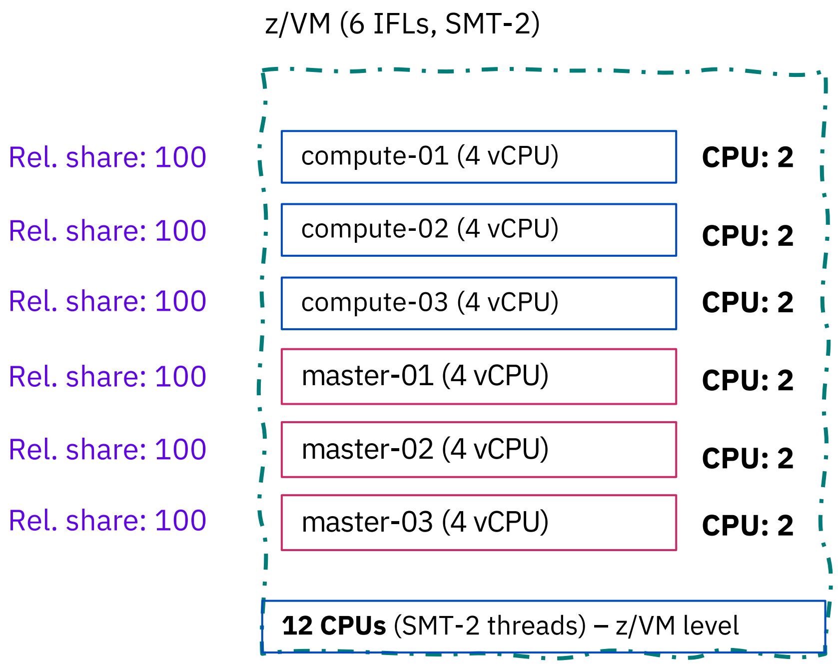
All nodes are of the same size, the defaults are pretty fine.
Each guest is considered to be able to get 2 vCPUs mapped to the existing SMT-2 threads.

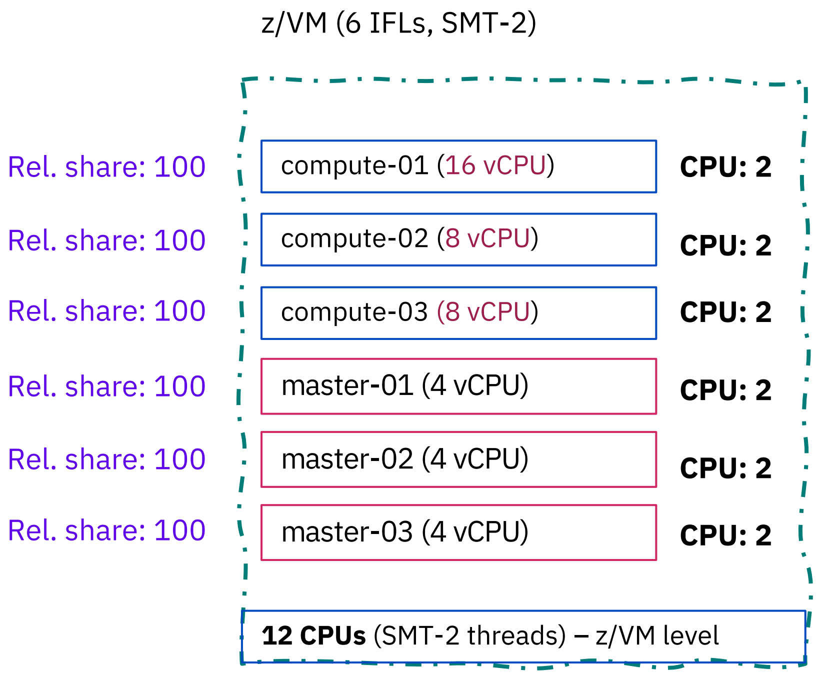
Level Up – z/VM perspective

What does it mean for OpenShift on Z?

Default SHARE settings are:

- 100 per guest (normal share)
- nolimit (max share)





All nodes are of the same size, the defaults are pretty fine.
Each guest is considered to be able to get 2 vCPUs mapped to the existing SMT-2 threads.

*If we resize the guests, but forget to update the shares – **the VM entitlement doesn't follow**.*

Level Up – z/VM perspective

What does it mean for OpenShift on Z?

Default SHARE settings are:

- 100 per guest (normal share)
- nolimit (max share)

z/VM (6 IFLs, SMT-2)

Rel. share	Guest	CPU
Rel. share: 100	compute-01 (4 vCPU)	**CPU: 2**
Rel. share: 100	compute-02 (4 vCPU)	**CPU: 2**
Rel. share: 100	compute-03 (4 vCPU)	**CPU: 2**
Rel. share: 100	master-01 (4 vCPU)	**CPU: 2**
Rel. share: 100	master-02 (4 vCPU)	**CPU: 2**
Rel. share: 100	master-03 (4 vCPU)	**CPU: 2**

12 CPUs (SMT-2 threads) – z/VM level

z/VM (6 IFLs, SMT-2)

Rel. share	Guest	CPU
Rel. share: 1600	compute-01 (16 vCPU)	**CPU: 4,36**
Rel. share: 800	compute-02 (8 vCPU)	**CPU: 2,18**
Rel. share: 800	compute-03 (8 vCPU)	**CPU: 2,18**
Rel. share: 400	master-01 (4 vCPU)	**CPU: 1,09**
Rel. share: 400	master-02 (4 vCPU)	**CPU: 1,09**
Rel. share: 400	master-03 (4 vCPU)	**CPU: 1,09**

12 CPUs (SMT-2 threads) – z/VM level

All nodes are of the same size, the defaults are pretty fine.
Each guest is considered to be able to get 2 vCPUs mapped to the existing SMT-2 threads.

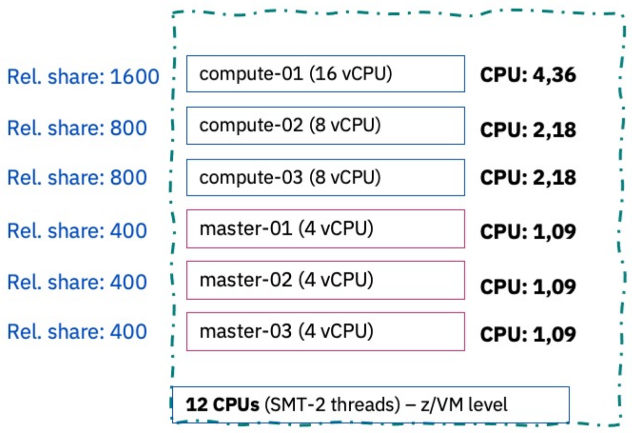
If we resize the guests, but forget to update the shares – the VM entitlement doesn't follow.

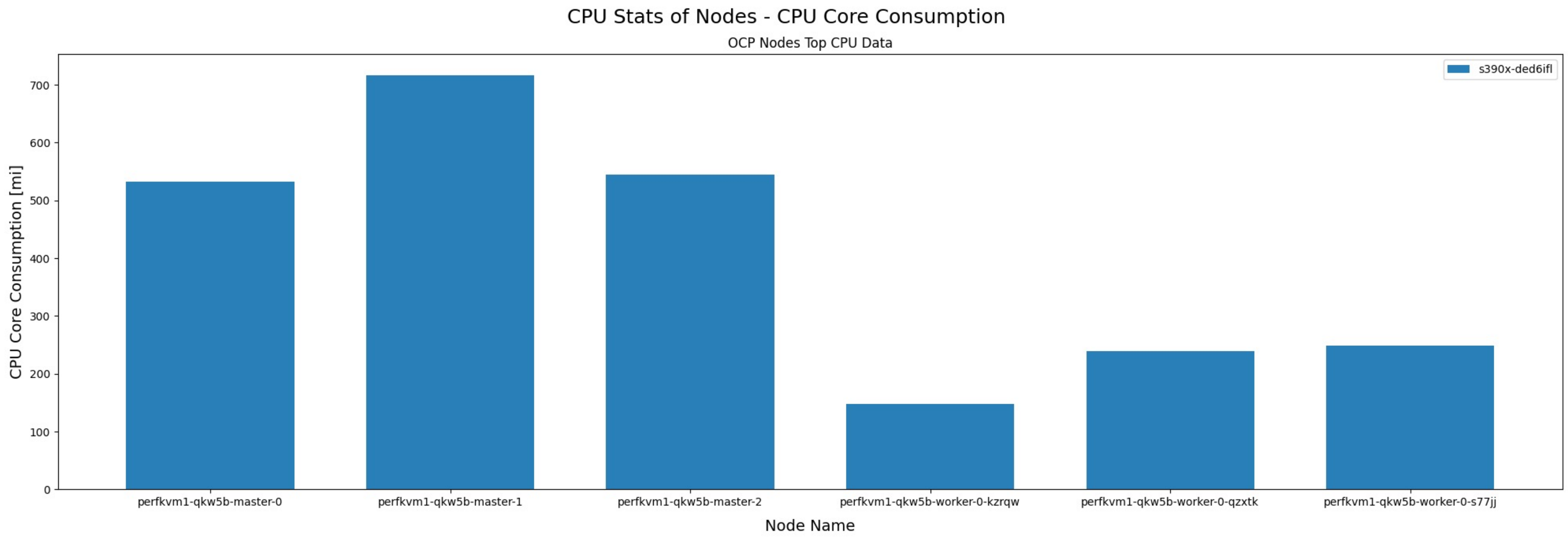
Best practice: Rel. SHARE = vCPU Count * 100

Level Up – z/VM perspective

Is it enough to have 1,09 / 4 vCPU guaranteed for the control plane nodes?

Let's check the average vCPU consumption of the control plane nodes for a cluster in steady state.





0.5 – 0.7 vCPU in average per control plane node (OCP v4.9 on KVM, 6 nodes in total)

Recommendation: By setting the z/VM (or KVM) Shares, make sure that the vCPU entitlement of the control plane nodes guarantees at least **1 vCPU per node** for a small cluster*.

* Be aware that the minimum value may vary with cluster size

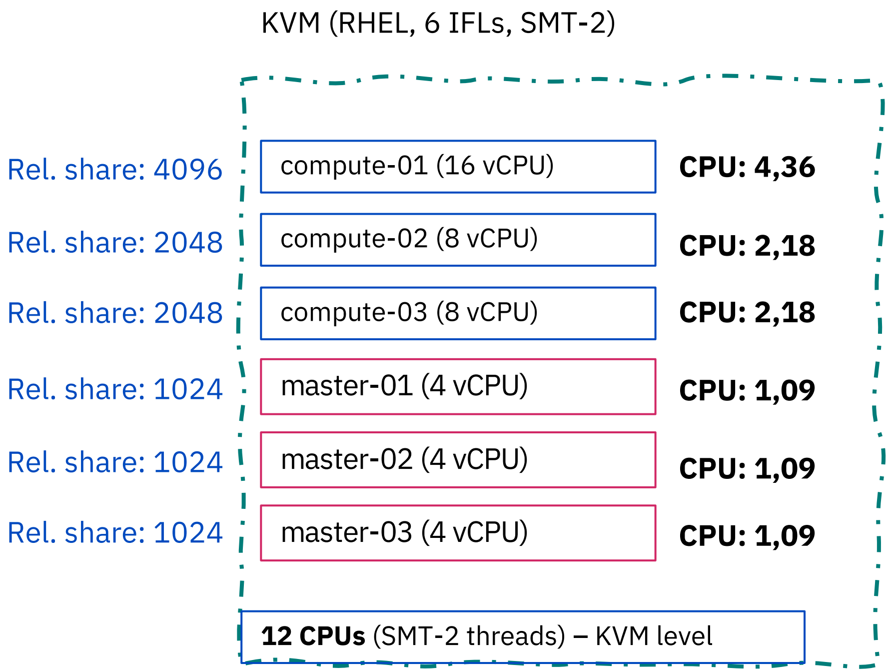
Level Up – KVM perspective

KVM CPU shares - Implemented by the Linux Scheduler and cgroups

Optionally specifies the initial CPU weight. The default is 1024.
Can be modified by changing the Domain XML:

```
<cputune>
    <shares>2048</shares>
</cputune>
```

Valid values are in the natural numbers between 2 and 262144.



Recommendation: By setting the KVM Shares, make sure that the vCPU entitlement of the master nodes guarantees at least **1 vCPU per node**.

Read more: [KVM Virtual Server Management]

Level Up – KVM perspective

KVM CPU sha

Optionally spe
Can be modifi

```
<cputune>
    <shares
</cputune>
```

Valid values a

Checkmark

#3

Ensure the control plane nodes are entitled to get a sufficient share of vCPU capacity – at least 1 vCPU per node for a small cluster*.

The control plane is the most critical part of the cluster. Make sure it doesn't starve due to high resource sharing.

* Be aware that the minimum value may vary with cluster size

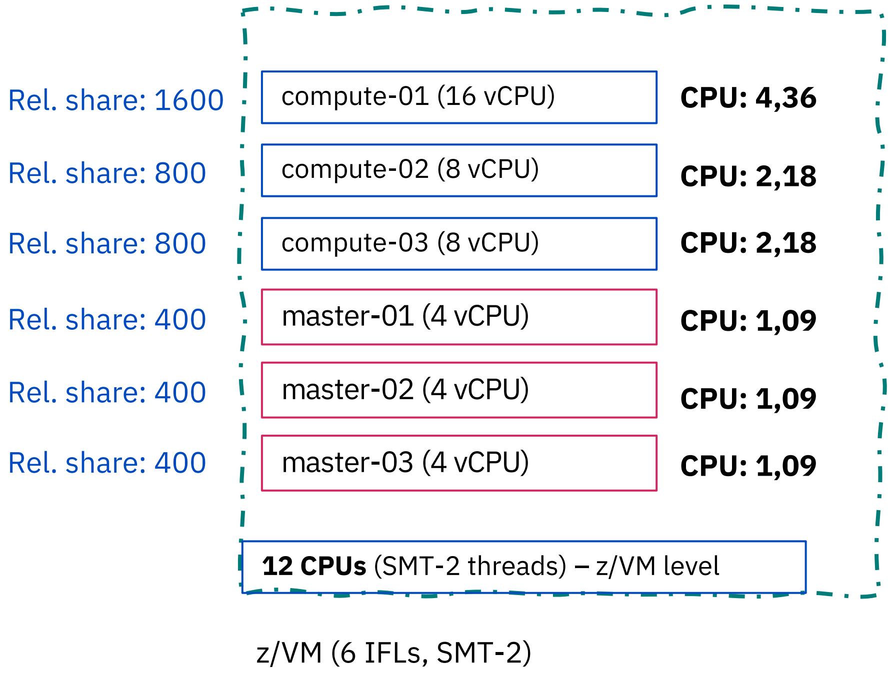
Rec e.

Read more: [KVM Virtual Server Management]

What if …

the compute plane starts to take off!

Is my control plane in danger?



What if …

the compute plane starts to take off!

Is my control plane in danger?

Let's put some load on the cluster.

Rel. share: 1600	compute-01 (16 vCPU)	CPU: 4,36
Rel. share: 800	compute-02 (8 vCPU)	CPU: 2,18
Rel. share: 800	compute-03 (8 vCPU)	CPU: 2,18
Rel. share: 400	master-01 (4 vCPU)	CPU: 1,09
Rel. share: 400	master-02 (4 vCPU)	CPU: 1,09
Rel. share: 400	master-03 (4 vCPU)	CPU: 1,09

12 CPUs (SMT-2 threads) – z/VM level

z/VM (6 IFLs, SMT-2)

14 vCPU (90%)

2 vCPU (26%)

4 vCPU (53%)

What if …

the control plane doesn't get enough capacity for critical operations?

OpenShift as a Kubernetes platform has mechanisms implemented to prevent it:

- The OpenShift scheduler always leaves 0.5 vCPU unallocatable on each node, for system operations. E.g. for a worker node of 4 vCPUs, only 3.5 vCPU worth of pod CPU requests is schedulable.
- User workload is not schedulable on the control plane – it is necessary to give it enough capacity to operate.

- Check the sizing recommendations for Control plane and infrastructure nodes – it scales with the cluster size.
- Resource starvation of the control plane will impact the overall cluster performance.

But, there are things which are out of control for the cluster:

- oversized nodes & CPU overcommitment
- lack of physical resources
- too low entitlement in shared environments
- noisy neighbours: VMs or LPARs

What if …

the contro

OpenShift a

- The Ope
  worker r
- User wo

- Check th
- Resourc

But, there a

- oversize
- lack of p
- too low e
- noisy ne

An “obvious” enemy:
oversized guests

Low amount of guaranteed vCPU share for the control plane.

Expected OCP symptoms:

- **increased response times**
- **increased steal CPU%**
- **possible ETCD leader changes**

Rel. share: 1200	compute-01 (12 vCPU)	**CPU: 3**
Rel. share: 1200	compute-02 (12 vCPU)	**CPU: 3**
Rel. share: 1200	compute-03 (12 vCPU)	**CPU: 3**
Rel. share: 400	master-01 (4 vCPU)	**CPU: 1**
Rel. share: 400	master-02 (4 vCPU)	**CPU: 1**
Rel. share: 400	master-03 (4 vCPU)	**CPU: 1**

12 CPUs (SMT-2 threads) – z/VM level

z/VM (6 IFLs, SMT-2)

What if …

the contro

OpenShift a

- The Ope
  worker r
- User wo

- Check th
- Resourc

But, there a

- oversize
- lack of p
- too low e
- noisy ne

Checkmark

#4

Monitor the workload and keep the size of VMs reasonable – **there is no benefit of having oversized guests**, only additional overhead and costs.

Virtual CPUs which are not utilized will still trigger the hypervisor signalling that they are available.

What if …

the contro

OpenShift a

- The Ope
  worker r
- User wo

- Check th
- Resourc

But, there a

- oversize
- lack of p
- too low
- noisy ne

Most common enemy:
the Noisy Neighbour

Rel. share: 5000	prod-01 (8 vCPU)	**CPU: 4,16**
Rel. share: 5000	prod-02 (8 vCPU)	**CPU: 4,16**
Rel. share: 1600	compute-01 (16 vCPU)	**CPU: 1,33**
Rel. share: 800	compute-02 (8 vCPU)	**CPU: 0,67**
Rel. share: 800	compute-03 (8 vCPU)	**CPU: 0,67**
Rel. share: 400	master-01 (4 vCPU)	**CPU: 0,33**
Rel. share: 400	master-02 (4 vCPU)	**CPU: 0,33**
Rel. share: 400	master-03 (4 vCPU)	**CPU: 0,33**

12 CPUs (SMT-2 threads) – z/VM level

z/VM (6 IFLs, SMT-2)

“Big turbo important production server”

14 vCPU (90%)

2 vCPU (26%)

4 vCPU (53%)

Expected OCP symptoms:

- **high response times**
- **high steal CPU%**
- **ETCD leader changes**
- **pods crashing/not deploying**

What if …

the contro

OpenShift a

- The Ope
  worker r
- User wo

- Check th
- Resourc

But, there a

- oversize
- lack of p
- too low
- noisy ne

Checkmark

#5

Understand the overall environment – keep the full picture in mind.

Neighbouring VMs and LPARs **do impact** the OpenShift environment if using shared IFL cores.

The Five Checkmarks You Don't Want to Miss

1. Ensure to meet the **minimum physical core requirements** for the cluster setup – 6 IFL cores (SMT-2 enabled) per cluster.

2. Ensure each cluster is **entitled to get sufficient capacity** for system operations – at least 2 IFL cores per LPAR in a single-LPAR cluster, or at least 1 IFL core per LPAR in a multi-LPAR deployment.

3. Ensure the control plane nodes are entitled to get a **sufficient share of vCPU capacity** – at least 1 vCPU per node for a small cluster.

4. Monitor the workload and **keep the size of VMs reasonable** – there is no benefit of having oversized guests, only additional overhead and costs.

5. Understand the overall environment – keep the full picture in mind. **Neighbouring VMs and LPARs do impact** the OpenShift environment if using shared IFL cores.

Resources

ibm.com/docs – OCP on Z Performance

docs.openshift.com – IBM Z & LinuxONE Host Recommendations

z/VM Performance education LPAR Topics

Thank You

Danijel Soldo
Performance Lead – OpenShift on Z & LinuxONE
IBM R&D Germany

—
danijel.soldo@de.ibm.com

Notices and disclaimers

-
- **U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts.
  In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."
- **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those
- customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers continued

- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**
- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.
- IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml

Pictures references

Postgres logo: https://upload.wikimedia.org/wikipedia/commons/2/29/Postgresql_elephant.svg
NodeJS logo: https://www.subpng.com/png-uf1mgi/download.html
Prometheus logo: https://upload.wikimedia.org/wikipedia/commons/3/38/Prometheus_software_logo.svg

Vocabulary

The machine is equipped with *physical cores.*
– They come in different *types:* a physical IFL core, a physical CP core ... (What your specific machine has depends upon what you bought)
– Each physical core contains two *processors* or *CPUs.*

The difference between *core* and *processor* is absolutely vital in the SMT world

A logical partition that has opted-in for SMT is equipped with *logical cores.*
– In an SMT-1 LPAR, each logical core contains one logical processor (or logical CPU)
– In an SMT-2 LPAR, the logical IFL cores have two processors each and the rest have one processor each
– PR/SM dispatches the LPAR’s logical cores on physical cores

Source:
https://www.vm.ibm.com/library/presentations/lparperf.pdf