Topology/Numa with Qemu/Linux

Whisper informal seminar

2021

```
qemu-system-x86_64 -nographic -enable-kvm \
-m 4096 \
-drive format=raw,file=$MY_LVM2_DEBIAN_ROOT \
-nic user,hostfwd=tcp::10022-:22
-kernel $MY_KERNEL -initrd $MY_INIT_RAMFS \
-append "console=ttyS0 root=/dev/sda debug sched_debug"
```

1

```
    -smp 2
```

2

```
    -smp cpus=2,sockets=2,dies=1,cores=1,threads=1
```

3

```
    -smp cpus=2,sockets=1,dies=2,cores=1,threads=1
```

4

```
    -smp cpus=2,sockets=1,dies=1,cores=2,threads=1
```

5

```
    -smp cpus=2,sockets=1,dies=1,cores=1,threads=2
```

6

```
-smp 4 -m 512
-object memory-backend-ram,size=128M,id=m0
- ...
-numa node,cpus=0,nodeid=0,memdev=m0
- ...
-numa dist,src=0,dst=1,val=20
-numa dist,src=0,dst=2,val=30
- ...
```

User-space tools tried :

- `numactl --hardware`
- `lstopo/hwloc` (from the *Open MPI repo*)

`https://www.kernel.org/doc/html/latest/admin-guide/cputopology.html`

Sysfs file system :

- `/sys/devices/system/cpu/cpuX/topology/`

⇒ thread_siblings ⇝ core_cpus

⇒ thread_siblings_list ⇝ core_cpus_list

⇒ core_siblings ⇝ package_cpus

⇒ core_siblings_list ⇝ package_cpus_list

Not pertinent for x86_64 (CONFIG_SCHED_BOOK and CONFIG_SCHED_DRAWER) :

⇒ book_id book_siblings book_siblings_list ⇝ ∅

⇒ drawer_id drawer_siblings drawer_siblings_list ⇝ ∅

Sysfs file system :

- `/sys/devices/system/cpu/cpuX/topology/`

⇒ core_id,cpus,cpus_list

⇒ die_id,cpus,cpus_list

⇒ physical_package_id package_cpus,cpus_list (∼ socket)

https://lore.kernel.org/qemu-devel/20190620054525.37188-4-like.xu@linux.intel.com/T/

Qemu :

⇒ before 2019, cpu topology = socket/core/thread model

⇒ after 2019, cpu topology = socket/die/core/thread model

-smp 1

core	die	package
0 1 0	0 1 0	0 1 0

```
Package L#0
  NUMANode L#0 (P#0 3932MB)
  L3 L#0 (16MB) + L2 L#0 (4096KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0 + PU L#0 (P#0)
```

-smp 2 = -smp cpus=2,sockets=2,dies=1,cores=1,threads=1

core	die	package
0 1 0	0 1 0	0 1 0
0 2 1	0 2 1	1 2 1

```
NUMANode L#0 (P#0 3931MB)
Package L#0 + L3 L#0 (16MB) + L2 L#0 (4096KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0 + PU L#0 (P#0)
Package L#1 + L3 L#1 (16MB) + L2 L#1 (4096KB) + L1d L#1 (32KB) + L1i L#1 (32KB) + Core L#1 + PU L#1 (P#1)
```

```
node distances:
node   0
  0:  10
CPU0 attaching sched-domain(s):
 domain-0: span=0-1 level=DIE
  groups: 0:{ span=0 }, 1:{ span=1 }
CPU1 attaching sched-domain(s):
 domain-0: span=0-1 level=DIE
  groups: 1:{ span=1 }, 0:{ span=0 }
root domain span: 0-1 (max cpu_capacity = 1024)
rd 0-1: CPUs do not have asymmetric capacities
```

-smp cpus=2,sockets=1,dies=2,cores=1,threads=1

core	die	package
0 1 0	0 1 0	0 1 0
0 2 1	1 2 1	0 2 1

```
NUMANode L#0 (P#0 3931MB)
Package L#0 + L3 L#0 (16MB) + L2 L#0 (4096KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0 + PU L#0 (P#0)
Package L#1 + L3 L#1 (16MB) + L2 L#1 (4096KB) + L1d L#1 (32KB) + L1i L#1 (32KB) + Core L#1 + PU L#1 (P#1)
```

```
node distances:
node   0
  0:  10
CPU0 attaching sched-domain(s):
 domain-0: span=0-1 level=DIE
  groups: 0:{ span=0 }, 1:{ span=1 }
CPU1 attaching sched-domain(s):
 domain-0: span=0-1 level=DIE
  groups: 1:{ span=1 }, 0:{ span=0 }
root domain span: 0-1 (max cpu_capacity = 1024)
rd 0-1: CPUs do not have asymmetric capacities
```

-smp cpus=2,sockets=1,dies=1,cores=2,threads=1

core	die	package
0 1 0	0 3 0-1	0 3 0-1
1 2 1	0 3 0-1	0 3 0-1

```
Package L#0
  NUMANode L#0 (P#0 3931MB)
  L3 L#0 (16MB)
    L2 L#0 (4096KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0 + PU L#0 (P#0)
    L2 L#1 (4096KB) + L1d L#1 (32KB) + L1i L#1 (32KB) + Core L#1 + PU L#1 (P#1)
```

```
node distances:
node   0
  0:  10
CPU0 attaching sched-domain(s):
 domain-0: span=0-1 level=MC
  groups: 0:{ span=0 }, 1:{ span=1 }
CPU1 attaching sched-domain(s):
 domain-0: span=0-1 level=MC
  groups: 1:{ span=1 }, 0:{ span=0 }
root domain span: 0-1 (max cpu_capacity = 1024)
rd 0-1: CPUs do not have asymmetric capacities
```

-smp cpus=2,sockets=1,dies=1,cores=1,threads=2

core	die	package
0 3 0-1	0 3 0-1	0 3 0-1
0 3 0-1	0 3 0-1	0 3 0-1

```
Package L#0
  NUMANode L#0 (P#0 3931MB)
  L3 L#0 (16MB) + L2 L#0 (4096KB) + Core L#0
    L1d L#0 (32KB) + L1i L#0 (32KB) + PU L#0 (P#0)
    L1d L#1 (32KB) + L1i L#1 (32KB) + PU L#1 (P#1)
```

```
node distances:
node   0
  0:  10
CPU0 attaching sched-domain(s):
 domain-0: span=0-1 level=SMT
  groups: 0:{ span=0 }, 1:{ span=1 }
CPU1 attaching sched-domain(s):
 domain-0: span=0-1 level=SMT
  groups: 1:{ span=1 }, 0:{ span=0 }
root domain span: 0-1 (max cpu_capacity = 1024)
rd 0-1: CPUs do not have asymmetric capacities
```

https://lkml.org/lkml/2020/2/19/133

But the topology exposed in the sysfs is not the sched_domain topology !

```
sched_debug
```

```
echo 0 > /sys/devices/system/cpu/cpu1/online
echo 1 > /sys/devices/system/cpu/cpu1/online
```

```
[  895.280035] CPU0 attaching NULL sched-domain.
[  895.281794] CPU0 attaching sched-domain(s):
[  895.283219]  domain-0: span=0-1 level=DIE
[  895.284615]   groups: 0:{ span=0 }, 1:{ span=1 }
[  895.285724] CPU1 attaching sched-domain(s):
[  895.286801]  domain-0: span=0-1 level=DIE
[  895.287749]   groups: 1:{ span=1 }, 0:{ span=0 }
[  895.288946] root domain span: 0-1 (max cpu_capacity = 1024)
[  895.290624] rd 0-1: CPUs do not have asymmetric capacities
```

cat /proc/schedstat

```
 version 15
timestamp 4294894831
cpu0 0 0 0 0 0 0 5032941745 1657344988 5560
domain0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
domain1 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
domain2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
cpu1 0 0 0 0 0 0 3131788587 1218124475 3860
domain0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
domain1 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
cpu2 0 0 0 0 0 0 3366853012 1069445494 3910
domain0 e 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
domain1 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
cpu3 0 0 0 0 0 0 2801910606 1279489011 3679
domain0 c 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
domain1 e 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
domain2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
  -smp 4 -m 512
  -object memory-backend-ram,size=128M,id=m0
  -object memory-backend-ram,size=128M,id=m1
  -object memory-backend-ram,size=128M,id=m2
  -object memory-backend-ram,size=128M,id=m3
  -numa node,cpus=0,nodeid=0,memdev=m0 -numa node,cpus=1,nodeid=1,memdev=m1
  -numa node,cpus=2,nodeid=2,memdev=m2 -numa node,cpus=3,nodeid=3,memdev=m3
  -numa dist,src=0,dst=1,val=20 -numa dist,src=0,dst=2,val=30
  -numa dist,src=0,dst=3,val=40 -numa dist,src=1,dst=2,val=20
  -numa dist,src=1,dst=3,val=30 -numa dist,src=2,dst=3,val=20
```

core	die	package
0 1 0	0 1 0	0 1 0
0 2 1	0 2 1	1 2 1
0 4 2	0 4 2	2 4 2
0 8 3	0 8 3	3 8 3

```
  NUMANode L#0 (P#0 125MB)
  L3 L#0 (16MB) + L2 L#0 (4096KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0 +
Package L#1
  NUMANode L#1 (P#1 126MB)
  L3 L#1 (16MB) + L2 L#1 (4096KB) + L1d L#1 (32KB) + L1i L#1 (32KB) + Core L#1 +
Package L#2
  NUMANode L#2 (P#2 126MB)
  L3 L#2 (16MB) + L2 L#2 (4096KB) + L1d L#2 (32KB) + L1i L#2 (32KB) + Core L#2 +
Package L#3
  NUMANode L#3 (P#3 96MB)
  L3 L#3 (16MB) + L2 L#3 (4096KB) + L1d L#3 (32KB) + L1i L#3 (32KB) + Core L#3 +
 node distances:
node   0   1   2   3
  0:  10  20  30  40
  1:  20  10  20  30
  2:  30  20  10  20
  3:  40  30  20  10
```

```
CPU0 attaching sched-domain(s):
 domain-0: span=0-1 level=NUMA
  groups: 0:{ span=0 }, 1:{ span=1 }
  domain-1: span=0-2 level=NUMA
   groups: 0:{ span=0-1 mask=0 cap=2048 }, 2:{ span=1-3 mask=2 cap=3072 }
ERROR: groups don't span domain->span
   domain-2: span=0-3 level=NUMA
    groups: 0:{ span=0-2 mask=0 cap=3072 }, 3:{ span=1-3 mask=3 cap=3072 }
CPU1 attaching sched-domain(s):
 domain-0: span=0-2 level=NUMA
  groups: 1:{ span=1 }, 2:{ span=2 }, 0:{ span=0 }
  domain-1: span=0-3 level=NUMA
   groups: 1:{ span=0-2 mask=1 cap=3072 }, 3:{ span=2-3 mask=3 cap=2048 }
CPU2 attaching sched-domain(s):
 domain-0: span=1-3 level=NUMA
  groups: 2:{ span=2 }, 3:{ span=3 }, 1:{ span=1 }
  domain-1: span=0-3 level=NUMA
   groups: 2:{ span=1-3 mask=2 cap=3072 }, 0:{ span=0-1 mask=0 cap=2048 }
CPU3 attaching sched-domain(s):
 domain-0: span=2-3 level=NUMA
  groups: 3:{ span=3 }, 2:{ span=2 }
  domain-1: span=1-3 level=NUMA
   groups: 3:{ span=2-3 mask=3 cap=2048 }, 1:{ span=0-2 mask=1 cap=3072 }
ERROR: groups don't span domain->span
   domain-2: span=0-3 level=NUMA
    groups: 3:{ span=1-3 mask=3 cap=3072 }, 0:{ span=0-2 mask=0 cap=3072 }
root domain span: 0-3 (max cpu_capacity = 1024)
rd 0-3: CPUs do not have asymmetric capacities
```

This is the diameter >= 3 bug

→ cf valentin schneider 2020 lpc scheduling microconference

There is a patch

→ https://lore.kernel.org/lkml/jhj4kiu4hz8.mognet@arm.com/T/

```
-smp 4 -m 512
-object memory-backend-ram,size=128M,id=m0
-object memory-backend-ram,size=128M,id=m1
-object memory-backend-ram,size=128M,id=m2
-object memory-backend-ram,size=128M,id=m3
-numa node,cpus=0,nodeid=0,memdev=m0 -numa node,cpus=1,nodeid=1,memdev=m1
-numa node,cpus=2,nodeid=2,memdev=m2 -numa node,cpus=3,nodeid=3,memdev=m3
```

```
node distances:
node   0   1   2   3
  0:  10  20  20  20
  1:  20  10  20  20
  2:  20  20  10  20
  3:  20  20  20  10
```

```
CPU0 attaching sched-domain(s):
 domain-0: span=0-3 level=NUMA
  groups: 0:{ span=0 }, 1:{ span=1 }, 2:{ span=2 }, 3:{ span=3 }
CPU1 attaching sched-domain(s):
 domain-0: span=0-3 level=NUMA
  groups: 1:{ span=1 }, 2:{ span=2 }, 3:{ span=3 }, 0:{ span=0 }
CPU2 attaching sched-domain(s):
 domain-0: span=0-3 level=NUMA
  groups: 2:{ span=2 }, 3:{ span=3 }, 0:{ span=0 }, 1:{ span=1 }
CPU3 attaching sched-domain(s):
 domain-0: span=0-3 level=NUMA
  groups: 3:{ span=3 }, 0:{ span=0 }, 1:{ span=1 }, 2:{ span=2 }
root domain span: 0-3 (max cpu_capacity = 1024)
rd 0-3: CPUs do not have asymmetric capacities
```

```
    -smp 4 -m 512
    -object memory-backend-ram,size=128M,id=m0
    -object memory-backend-ram,size=128M,id=m1
    -object memory-backend-ram,size=128M,id=m2
    -object memory-backend-ram,size=128M,id=m3
    -numa node,cpus=0,nodeid=0,memdev=m0 -numa node,cpus=1,nodeid=1,memdev=m1
    -numa node,cpus=2,nodeid=2,memdev=m2 -numa node,cpus=3,nodeid=3,memdev=m3
    -numa dist,src=0,dst=1,val=20 -numa dist,src=0,dst=2,val=30
    -numa dist,src=0,dst=3,val=30 -numa dist,src=1,dst=2,val=20
    -numa dist,src=1,dst=3,val=30 -numa dist,src=2,dst=3,val=20
```

```
node distances:
node   0   1   2   3
  0:  10  20  30  30
  1:  20  10  20  30
  2:  30  20  10  20
  3:  30  30  20  10
```

```
CPU0 attaching sched-domain(s):
 domain-0: span=0-1 level=NUMA
  groups: 0:{ span=0 }, 1:{ span=1 }
  domain-1: span=0-3 level=NUMA
   groups: 0:{ span=0-1 mask=0 cap=2048 }, 2:{ span=1-3 mask=2 cap=3072 }
CPU1 attaching sched-domain(s):
 domain-0: span=0-2 level=NUMA
  groups: 1:{ span=1 }, 2:{ span=2 }, 0:{ span=0 }
  domain-1: span=0-3 level=NUMA
   groups: 1:{ span=0-2 mask=1 cap=3072 }, 3:{ span=2-3 mask=3 cap=2048 }
CPU2 attaching sched-domain(s):
 domain-0: span=1-3 level=NUMA
  groups: 2:{ span=2 }, 3:{ span=3 }, 1:{ span=1 }
  domain-1: span=0-3 level=NUMA
   groups: 2:{ span=1-3 mask=2 cap=3072 }, 0:{ span=0-1 mask=0 cap=2048 }
CPU3 attaching sched-domain(s):
 domain-0: span=2-3 level=NUMA
  groups: 3:{ span=3 }, 2:{ span=2 }
```

```
-smp cpus=24,sockets=2,dies=1,cores=3,threads=4
```

core	die	package
0 0x00000f 0-3	0 0x000fff 0-11	0 0x000fff 0-11
0 0x00000f 0-3	0 0x000fff 0-11	0 0x000fff 0-11
0 0x00000f 0-3	0 0x000fff 0-11	0 0x000fff 0-11
0 0x00000f 0-3	0 0x000fff 0-11	0 0x000fff 0-11
1 0x0000f0 4-7	0 0x000fff 0-11	0 0x000fff 0-11
1 0x0000f0 4-7	0 0x000fff 0-11	0 0x000fff 0-11
1 0x0000f0 4-7	0 0x000fff 0-11	0 0x000fff 0-11
1 0x0000f0 4-7	0 0x000fff 0-11	0 0x000fff 0-11
2 0x000f00 8-11	0 0x000fff 0-11	0 0x000fff 0-11
2 0x000f00 8-11	0 0x000fff 0-11	0 0x000fff 0-11
2 0x000f00 8-11	0 0x000fff 0-11	0 0x000fff 0-11
2 0x000f00 8-11	0 0x000fff 0-11	0 0x000fff 0-11
0 0x00f000 12-15	0 0xfff000 12-23	1 0xfff000 12-23
0 0x00f000 12-15	0 0xfff000 12-23	1 0xfff000 12-23
0 0x00f000 12-15	0 0xfff000 12-23	1 0xfff000 12-23
0 0x00f000 12-15	0 0xfff000 12-23	1 0xfff000 12-23
1 0x0f0000 16-19	0 0xfff000 12-23	1 0xfff000 12-23
1 0x0f0000 16-19	0 0xfff000 12-23	1 0xfff000 12-23
1 0x0f0000 16-19	0 0xfff000 12-23	1 0xfff000 12-23
1 0x0f0000 16-19	0 0xfff000 12-23	1 0xfff000 12-23
2 0xf00000 20-23	0 0xfff000 12-23	1 0xfff000 12-23
2 0xf00000 20-23	0 0xfff000 12-23	1 0xfff000 12-23
2 0xf00000 20-23	0 0xfff000 12-23	1 0xfff000 12-23
2 0xf00000 20-23	0 0xfff000 12-23	1 0xfff000 12-23

```
CPU23 attaching sched-domain(s):
 domain-0: span=20-23 level=SMT
  groups: 23:{ span=23 }, 20:{ span=20 }, 21:{ span=21 }, 22:{ span=22 }
  domain-1: span=12-23 level=MC
   groups: 20:{ span=20-23 cap=4096 }, 12:{ span=12-15 cap=4096 }, 16:{ span=16-19 cap=4096 }
   domain-2: span=0-23 level=DIE
    groups: 12:{ span=12-23 cap=12288 }, 0:{ span=0-11 cap=12288 }
root domain span: 0-23 (max cpu_capacity = 1024)
rd 0-23: CPUs do not have asymmetric capacities
```

```
cpu23 0 0 0 0 0 0 3426411969 359485251 288
domain0 f00000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
domain1 fff000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
domain2 ffffff 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
    -smp cpus=32,sockets=4,dies=2,cores=2,threads=2
    -m 512 \
    -object memory-backend-ram,size=128M,id=m0 \
    -object memory-backend-ram,size=128M,id=m1 \
    -object memory-backend-ram,size=128M,id=m2 \
    -object memory-backend-ram,size=128M,id=m3 \
    -numa node,cpus=0-7,nodeid=0,memdev=m0 \
    -numa node,cpus=8-15,nodeid=1,memdev=m1 \
    -numa node,cpus=16-23,nodeid=2,memdev=m2 \
    -numa node,cpus=24-31,nodeid=3,memdev=m3 \
    -numa dist,src=0,dst=1,val=20 -numa dist,src=0,dst=2,val=30 \
    -numa dist,src=0,dst=3,val=30 -numa dist,src=1,dst=2,val=20 \
    -numa dist,src=1,dst=3,val=30 -numa dist,src=2,dst=3,val=20 \
```

```
  Package L#0
    NUMANode L#0 (P#0 95MB)
    Die L#0 + L3 L#0 (16MB)
      L2 L#0 (4096KB) + Core L#0
        L1d L#0 (32KB) + L1i L#0 (32KB) + PU L#0 (P#0)
        L1d L#1 (32KB) + L1i L#1 (32KB) + PU L#1 (P#1)
      L2 L#1 (4096KB) + Core L#1
        L1d L#2 (32KB) + L1i L#2 (32KB) + PU L#2 (P#2)
        L1d L#3 (32KB) + L1i L#3 (32KB) + PU L#3 (P#3)
    Die L#1 + L3 L#1 (16MB)
      L2 L#2 (4096KB) + Core L#2
        L1d L#4 (32KB) + L1i L#4 (32KB) + PU L#4 (P#4)
        L1d L#5 (32KB) + L1i L#5 (32KB) + PU L#5 (P#5)
      L2 L#3 (4096KB) + Core L#3
        L1d L#6 (32KB) + L1i L#6 (32KB) + PU L#6 (P#6)
        L1d L#7 (32KB) + L1i L#7 (32KB) + PU L#7 (P#7)
```

```
CPU0 attaching sched-domain(s):
 domain-0: span=0-1 level=SMT
  groups: 0:{ span=0 }, 1:{ span=1 }
  domain-1: span=0-3 level=MC
   groups: 0:{ span=0-1 cap=2048 }, 2:{ span=2-3 cap=2048 }
   domain-2: span=0-7 level=DIE
    groups: 0:{ span=0-3 cap=4096 }, 4:{ span=4-7 cap=4096 }
    domain-3: span=0-15 level=NUMA
     groups: 0:{ span=0-7 cap=8192 }, 8:{ span=8-15 cap=8192 }
     domain-4: span=0-31 level=NUMA
      groups: 0:{ span=0-15 mask=0-7 cap=16384 }, 16:{ span=8-31 mask=16-23 cap=24576 }
```

- 0 has L1i and L1d (SMT)
- 0-1 share L2 (MC)
- 0-3 share L3 (DIE)
- 0-7 share 128M of the 512M of RAM (NUMA)
- 0-7 is closer to 8-15 than 16-23 or 24-31 so there is another level of NUMA
- if we had put -numa dist,src=1,dst=3,val=40, we would have a third level of NUMA

Why3 : domain.mlw

```
type flag = DOMAIN_SMT | DOMAIN_CACHE | DOMAIN_NUMA
```

⇝

- Simultaneous multithreading (SMT) (share L2)
- Multi-Core Cache (MC) (share L3)
- Package (DIE) (don't share L3)

```
type group = list int
type domain = {
  dcores  : group;
  groups : list group;
  flag   : flag;
}
```

WIP : comments and example on linux source code structure :

- sd->groups
- sd->flags
- sd->parent et sd->child
- group->sgc->id
- group->sgc->capacity