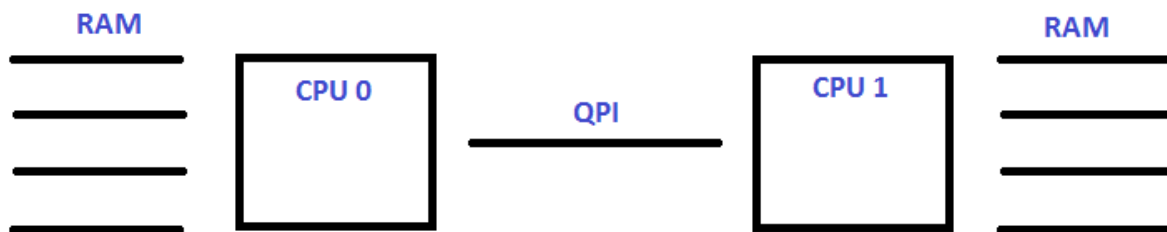# NUMA Cores

NUMA (non-uniform memory access) is a method of configuring a cluster of microprocessors in a multiprocessing system so that they can share memory locally, improving performance and the ability of the system to be expanded. NUMA is used in a symmetric multiprocessing (SMP) system. An SMP system is a "tightly-coupled," "share everything" system in which multiple processors working under a single operating system access each other's memory over a common bus or "interconnect" path. Ordinarily, a limitation of SMP is that as microprocessors are added, the shared bus or data path get overloaded and becomes a performance bottleneck. NUMA adds an intermediate level of memory shared among a few microprocessors so that all data accesses don't have to travel on the main bus.

The G6 has two CPUs. Each CPU has 12 Cores (each core has hyper-threading) so from the OS looks like there are 48 CPUs (2×12 Cores=24, 24 hyper threads (2 per) = 48)

The G7 has two CPUs. Each CPU has 18 Cores, so from the OS it looks like there are 72 CPUs (2×18 Cores=36, 36 hyper threads (2 per) = 72)

Each CPU has its own memory, RAM banks on the motherboard are linked to each CPU (but that memory can be accessed by the other CPU(s) but this is access is slower.

In normal operation, the MFEL is configured such that NUMA is enabled in the BIOS and in the advanced parameters. This restricts a service to accessing a single NUMA core (so a service will not take cores from both CPUs, only from one).

Because in this configuration, a single service uses a single processor, if you have one service, only one processor is used. The servers always have two processors. So if you can do one service, it follows that you can do two. If you can do three services, then you can do four.

It is important to note, that if you have an ABR or multi-video service, that you can get to the point where a processor is unable to handle that much work, and you will get a Max Performance exceeded error because the CPU has too much load. Because of the NUMA configuration, the second CPU is doing practically no work at all (a few OS tasks maybe).

When you look on the MFEL, the CPU will report only around 50%, this is confusing because people don't understand why Max Performance has been reached at only 50% CPU. For services that have a lot of video components, like ABR services, we can allow via an Advanced Parameter, the service to share processors, thus resolving this performance issue.

The reason we use the NUMA in this way, is that we want to ensure that a service is using the same CPU and memory resource. As allowing a unit with a lot of single video services access to share CPUs can lead to a loss of efficiency (as we know if a CPU starts accessing the memory of another CPU, this is slower).