April 01, 2019

# PROVIDING COST EFFECTIVE QUORUM CONSISTENCY FOR VERY SMALL CLOUD OR EDGE COMPUTE ENVIRONMENTS

Ian Wells

Kyle Mestery



# PROVIDING COST EFFECTIVE QUORUM CONSISTENCY FOR VERY SMALL CLOUD OR EDGE COMPUTE ENVIRONMENTS

AUTHORS:
Ian Wells
Kyle Mestery

## ABSTRACT

Techniques are described for using free space in a server chassis to house an additional node of low-powered compute resources in order to enable three-node (3-node) quorum-consistent systems where three full-size servers are not installed.

## DETAILED DESCRIPTION

For quorum consistency (e.g., for avoiding possible split-brain effects by using majority-wins consensus), a number of nodes are needed where a subset can be a majority. Basically, an odd number of nodes, and at least three, are needed for quorum consistency. However, in really small cloud compute environments or edge compute environments (e.g., branch offices, network operator central offices, etc.), three servers may not necessarily be present. For example, only two (2) servers (nodes) may be present, while three nodes are needed for a quorum.

Figure 1, below illustrates a typical 3-server compute environment.

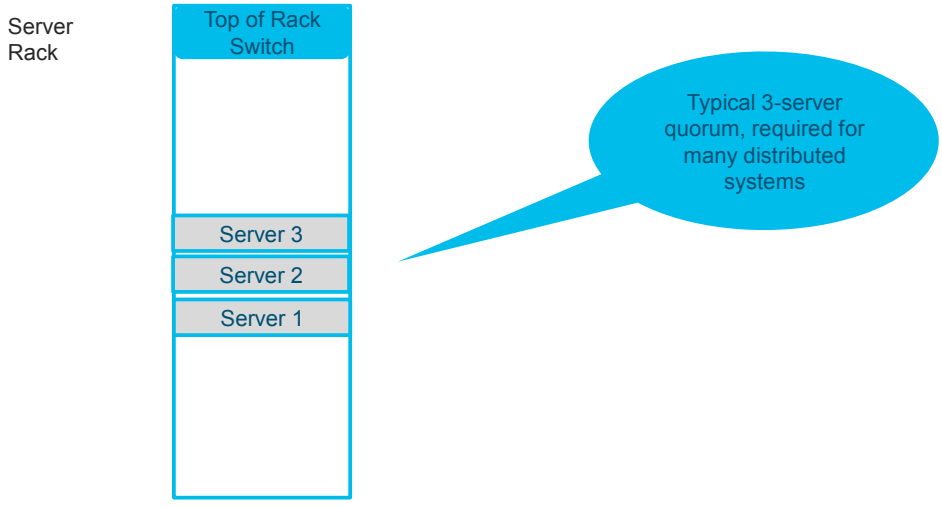


*Figure 1*

While adding a third, full-size server to a 2-server system might be a potential solution for achieving quorum consistency, adding a third server causes issues with physical cost, space, and power. For example, the server will take another rack unit (RU) The power is also problematic, especially since edge sites rented from incumbent network operators typically have 4 kilowatt (kW) racks. Thus, the costs incurred by a third server that will not be doing useful work in a cloud/edge environment is prohibitive to implement just for the redundancy aspect.

This proposal consists of novel hardware that can be used to achieve single-fault tolerant quorums in 2-node systems. In particular, this proposal involves using an additional server that occupies the form factor of a Peripheral Component Interconnect (PCI) slot but does not interface in any way with its hosting server. Figure 2 illustrates example details associated with a system in which a 3-server in a 2-server quorum can be achieved.

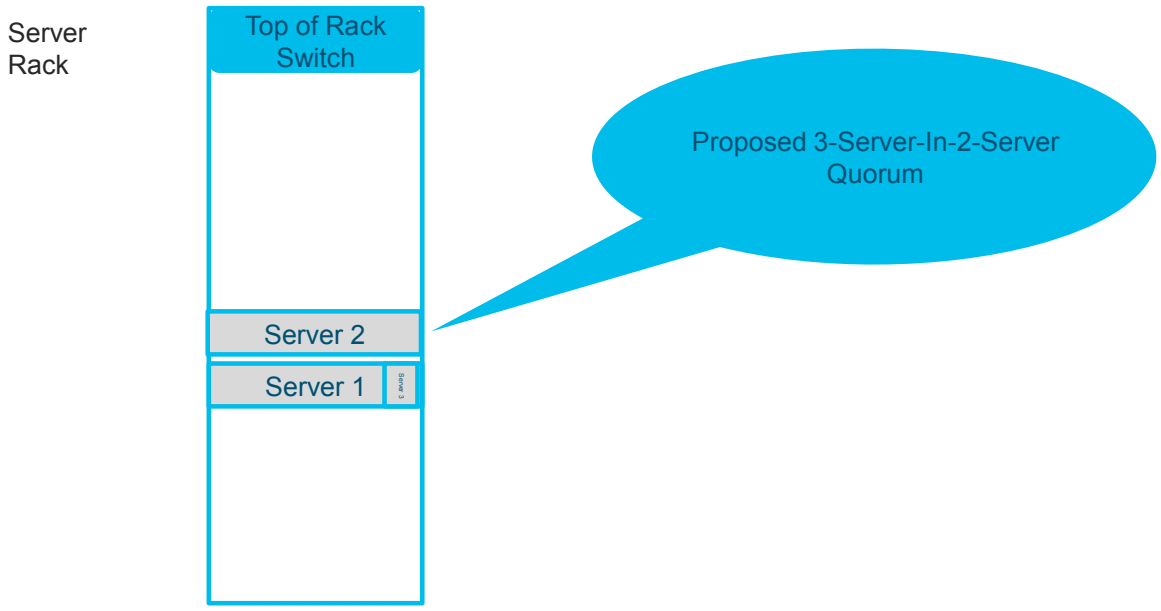


*Figure 2*

For the system proposed herein, power is not drawn from the server and no wiring is attached to the server's PCI bus; the location is purely a matter of convenient physical space. Instead, the backplate for the host server is used to provide power, networking, and out-of-band (OOB) management (OOBM) (e.g., Network Interface Controller (NIC))

connections for the third server. Figure 3 illustrates example details associated with the system proposed herein.

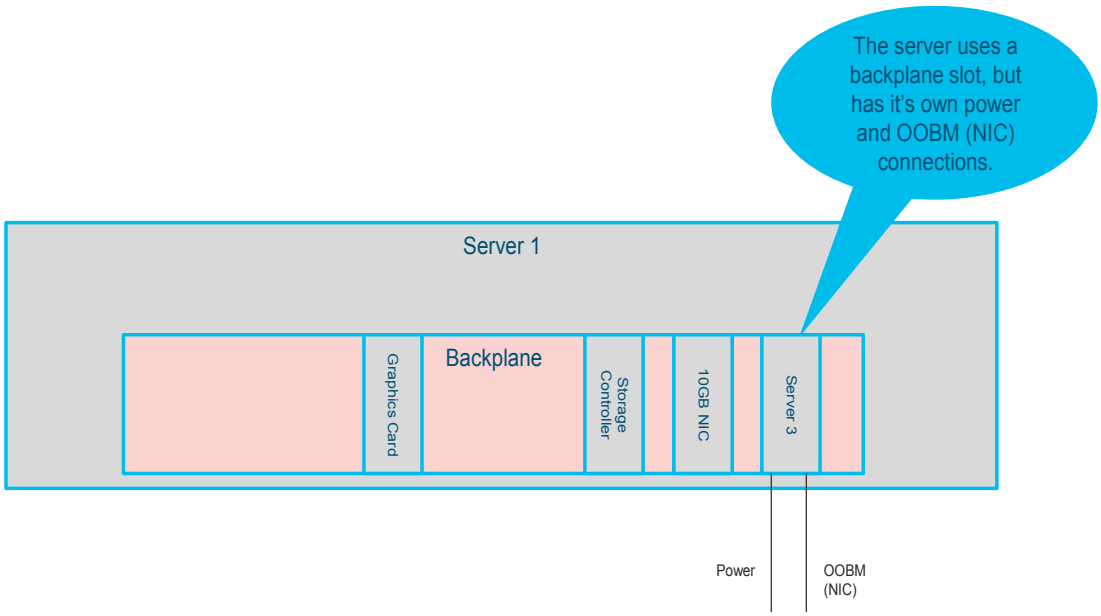


*Figure 3*

As illustrated in Figure 3, a third server (Server 3) uses a backplane slot of a host server (Server 1) but power and OOBM (NIC) connections for the third server are provided via the host server's backplate.

A system as proposed herein is clearly achievable at low cost (e.g., 1/100th of a full standard data center (DC) server) based on the cost of small Advanced RISC (reduced instruction set computing) Machine (ARM) systems. The capacity of ARM systems is no bar, as relatively few components in such systems require quorum, typically, for example, reliable messaging systems such as RabbitMQ, databases such as Etcd and MySQL, and redundancy helper systems such as the Corosync Cluster Engine. However, as a bare system they simply have no home in a rack.

A system as proposed herein–imagine a PCI bus with a copper-free slot, simply held in place by the PCI connector, as one example–occupies no additional space in the rack. It is quite common in PCI based servers for most of the slots to be unoccupied, so the space that the third server does occupy has no particular value.

If power is kept independent, there are very few points of common failure between the additional node and the host server that houses the additional node. Although thermal

failure is a possibility, this could also be mitigated by drawing air through the backplate and providing a sealed system in which the air is circulated. However, thermal runaway failures are not common and this might be considered an acceptable risk.

For additional operational simplicity, the form factor might be modified to a PCI 'card' holding a module that is inserted via the backplate, allowing replacement without having to depower and/or disassemble the host server. In addition, the system as proposed herein could also work in non-PCI form factors by having a physical space built into the host server in some other place, such as the front place, or by occupying a disk sled slot.

There are certainly cases where one chassis has multiple blades in it, and this does also solve the quorum consensus problem. However, such an implementation typically has a very large quantity of compute resources. In contrast, a system as proposed herein adds a very small quantity of compute resources because the need is not for an expanded footprint but an additional failure domain. Thus, while alternatives are possible, the system proposed herein has more value and would be preferred over other potential solutions.

As an example, when considering managed-cloud solutions, the management footprint required is orders of magnitude smaller than the footprint for the workhorse boxes. For instance, in OpenStack®, one core is needed for the control plane and 20 cores for workloads. This could be partially resolved by putting the control plane on the compute nodes, but this limits such a system to a minimum three node cluster when many enterprises are seeking 2-node solutions.

For Kubernetes (often referred to as "k8s"), the problem is worse as such systems seek to run the control plane off of the workhorse boxes completely, which is enforced through default scheduling properties. This means that no matter how many bare metal k8s workhorse boxes are desired for a deployment, three additional nodes technically have to be added on top to create a local and redundant control plane for them or, alternatively, consider virtualizing the control plane in a virtual machine (VM) on the worker (which then gets back to the "three node problem" in small edge/cloud footprints).

This would be similarly true for environments such as Mesosphere and for more targeted workload management systems as might be involved in controlling a Hadoop cluster, for example, for bare metal management systems that look after a collection of

OOB management, skinny Software Defined Network (SDN) solutions, or bare metal deployment systems such as Razor.

Also, larger servers can be a problem in edge sites. For example, consider a scenario in which it is desired to implement an edge site having power limits of 4.5kW per rack. This is easily exceeded with servers (750W-1kW) and switchgear requirements (switches plus routers plus management switches).

In contrast, a system as proposed herein would provide for the ability to construct very small Raspberry-Pi level control nodes with insignificant power draw and which are perfectly feasible to fit in the dimensions of a conventional server opening (e.g., a 2.5in (inch) or 3.5in disk bay), a custom-designed slot on a single purpose server, or a switch, router, gateway, etc. The system as proposed herein would even work in blade and sled chassis form factors as an additional smaller bay for the specialized low power compute, whereas such servers today have a single slot size as they only consider that one processor thermal design power (TDP)/footprint is required.

In summary, techniques are described for using free space in a server chassis to house an additional node of low powered compute resources in order to enable 3-node quorum systems where three full-size servers are not installed.