

## Application Server Objects using all your CPU?

Here is the scenario, over time you have developed your system and lots of objects have been added. Now there are so many objects running on your cpu that performance is seriously hurts. When pushing a button it takes 8 seconds to respond. Well there is a solution. Those objects can run on a different processor core on the same machine or a different machine all together. In most cases this is as simple as dragging and dropping the objects to the other processor core.

Lets go into more detail on what I mean by dragging and dropping to another processor. You simply make another AppEngine and move the objects under it. So now if you had a 4 core processor instead of the objects just running on one core they would be running on 2 cores. Your thinking well why not use all 4 cores, that is a great idea, go for it. What might even be a better idea is to leave 1 core open to run generic system processes, but I haven't tested it myself.

What if all your cores are maxed out? A few options come to mind. Probably your best is going to be just buying a new machine, they just don't cost much now a days. Or you could deploy objects to another existing machine. A nice feature of the galaxy is it doesn't matter where the object is it can communicate with the rest of the galaxy. Another option would be to upgrade the processor. People don't think about upgrading components much anymore because of how cheap it is to buy a whole new computer, but there is an advantage here, you wouldn't have to get a new machine. Some possible costs of a new machine are: finding the money, time to install software, implementing corporate standards, having to go through the process of undeploying/deploying. Anyways it may be simpler to just upgrade a component instead of buying a new machine, in most cases just buying a new machine will be the way to go but something to think about.

Being proactive by load balancing before a performance issue is noticed by operators would be a good idea. The way you do that is by opening object viewer then going to your engines to check `Scheduler.ExecutionTimeAvg`, this value should not exceed 40% of an engine scan period. Also the overall CPU usage should not exceed 40%. If either of these are true you'll likely run into performance issues.

If you are wondering if issues have already occurred you can check `Scheduler.ScanOverrunsCnt`, this tells of if a scan took longer than the time allotted. Basically if there are a lot of these there is a problem, a thousand overruns might be ok but not 100,000. Overruns occur naturally during deployment or initialization. If checking for "a lot" isn't scientific enough for you then compare `Scheduler.ScanOverrunsCnt` to `Scheduler.ScanCyclesCnt`, the percentage of overruns to runs should never be more than 0.05%.

In summary, Application server makes it drag and drop easy to load balance your system. If you see your `Scheduler.ScanOverrunCnt` get big then you simply need to add a new engine, buy a new computer, or upgrade your current computer's hardware.

Reference: tech note 671