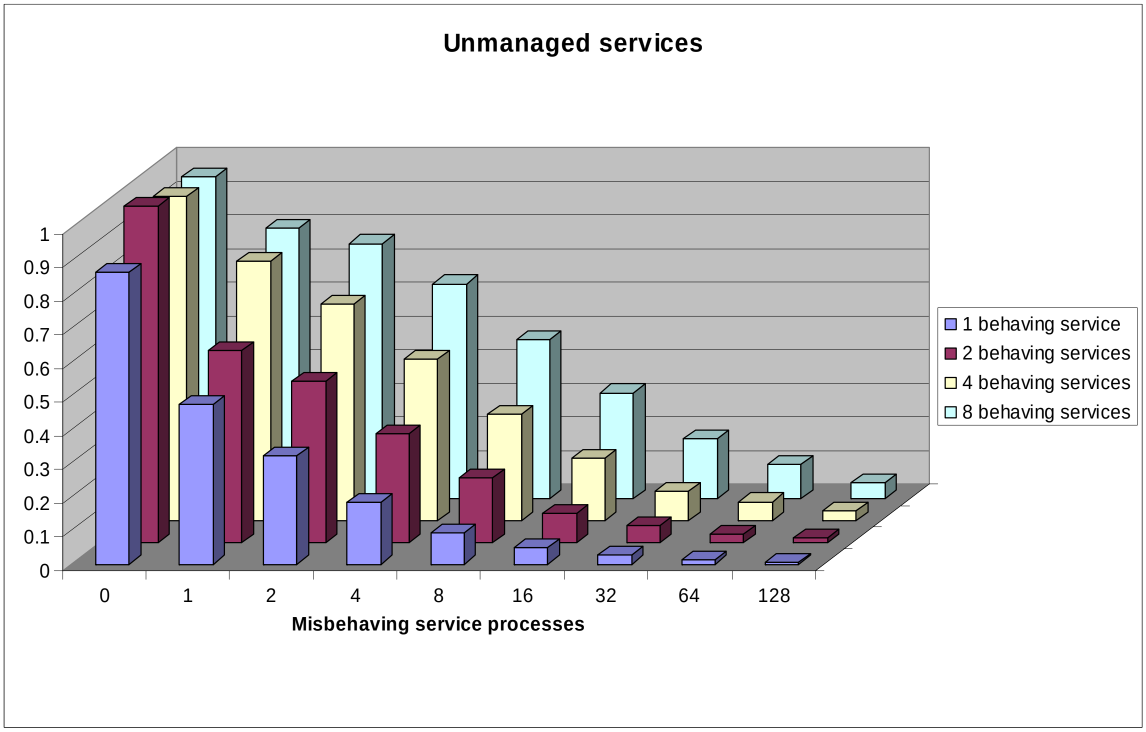
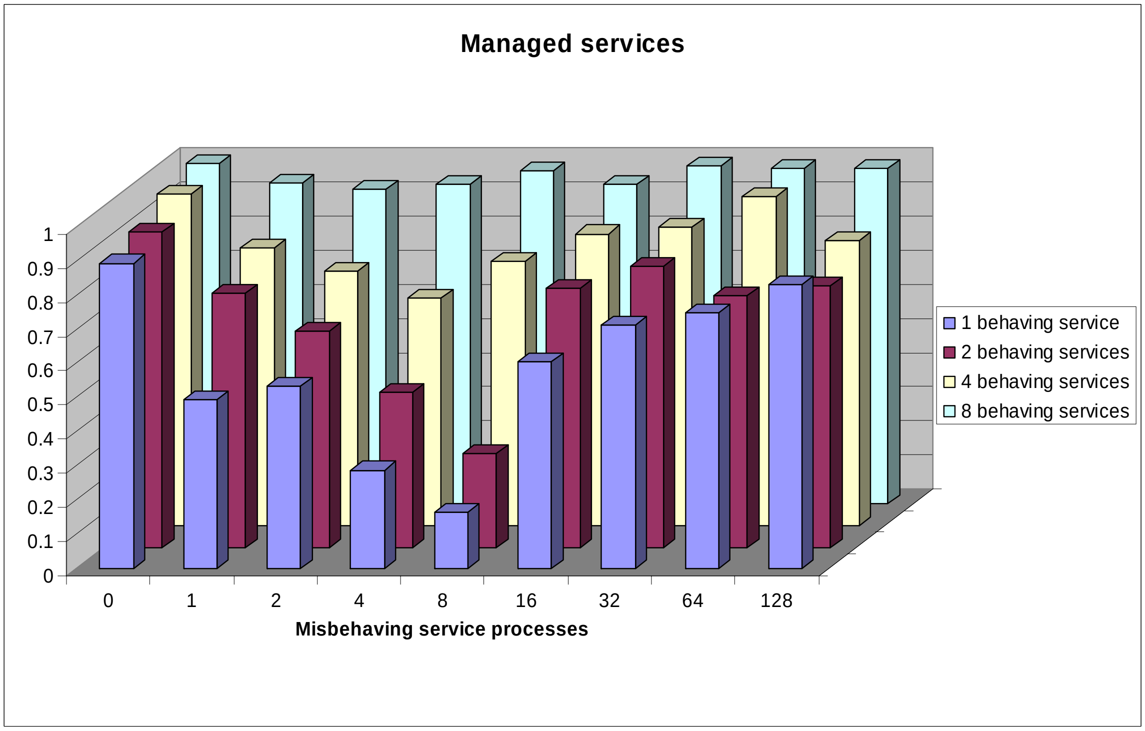
# Unmanaged vs. Managed Service



The graphs depict how increasing load affects services when their execution is managed and when it is not managed. To demonstrate that we defined two kinds of processes: "behaving processes" (implementing the services) and "misbehaving processes" (implementing load). The services (aka "behaving processes") are assumed to always stay within their resource allocation and their ability to do work is protected by regulating the load imposed by the "misbehaving processes". In our example, the processes are managed based on cumulative system load.

The "behaving services" were implemented as a small counter program, which counts how many times a certain multiplication can take place in a given time span. The resulting count obtained under load conditions is then normalized against the same program running with no other system load (maximum count). The normalization thus shows what proportion of the maximum count obtained under no load conditions a given process was able to achieve.

The "misbehaving services" were simulated with programs similar to the counter (all cpu bound) running in parallel. This generates load proportional to the number of processes.

We ran the tests with 1, 2, 4, and 8 behaving services as distinguished by different colors. In the graphs, the Y axis is the normalized counter value (output of the "behaving process"), and the X axis is the number of load-generating mis-behaving processes. What we want to show is that as the number of these processes (and thus system load) goes up, the counter program will not be starved of cpu time.

The policy implemented by the managed service is that if the load gets over 10, all jobs generating a load of 2 or above are suspended. When the load drops below 5, suspended jobs will be continued.

In the first graph, the behaving processes are not managed (i.e., the policy is not applied). Consequently, as the load increases, the output of the behaving processes decreases. In the second graph, the behaving processes are managed, (i.e. the policy is applied). For example, in the case on 1 behaving process and 8 misbehaving processes, enough load gets generated that the policy is applied, misbehaving processes are suspended and the behaving process claims a larger share of CPU. In the case of 4 behaving processes this dip occurs earlier (due to a larger number of behaving processes and thus larger load) and and earlier still with their further increase.

Comparing the two graphs, we can see that as the policy is applied (load>10), the counter applications is not getting affected very much as a managed services, but would be starved in a similar situation as a unmanaged service.