

---

Subject: Re: IO scheduling

Posted by [HaroldB](#) on Mon, 25 Sep 2006 07:05:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello. The ability for a VE to utilize all of the disk i/o bandwidth in a system seems to be a very big problem. Has anyone investigated a project called CKRM?

"If you want a way to assign io priorities without relying on process inheritance and (re)nice you might find CKRM, with it's cfq-based IO controller, useful.

Quote:

Basically you create a set of classes that group tasks and give an appropriate share of IO performance to tasks in that class. As processes get created CKRM will assign tasks to the IO classes based on a set of rules."

ref:

<http://ckrm.sourceforge.net/>

<http://www.gatago.com/linux/kernel/14683383.html>

Seems like if each CKRM "class" was a openvz VE, this could be a nice framework for limiting and more importantly guaranteeing disk i/o bandwidth per VE. Quoted from the CKRM patch:

Quote:

Resource allocations for a class is controlled by the parameters:

**guarantee:** specifies how much of a resource is guranteed to a class. A special value DONT\_CARE(-2) mean that there is no specific guarantee of a resource is specified, this class may not get any resource if the system is runing short of resources

**limit:** specifies the maximum amount of resource that is allowed to be allocated by a class. A special value DONT\_CARE(-2) mean that there is no specific limit is specified, this class can get all the resources available.

**total\_guarantee:** total guarantee that is allowed among the children of this class. In other words, the sum of "guarantee"s of all children of this class cannot exit this number.

**max\_limit:** Maximum "limit" allowed for any of this class's children. In other words, "limit" of any children of this class cannot exceed this value.

---