
Subject: Re: Scaling UBC values: Why?

Posted by [maratrus](#) on Fri, 04 Sep 2009 12:54:17 GMT

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

Hi,

all UBC configuration parameters should obey the consistency rules which are described at the page

http://wiki.openvz.org/UBC_consistency_check

It's just a simple math exercise

For example

Let's assume that there are exist two configuration examples.

Each of them should obey the inequalities that are described in a page mentioned above i.e.

$$\text{privvmpages_1_bar} \geq \text{vmguarpages_1_bar}$$
$$\text{privvmpages_2_bar} \geq \text{vmguarpages_2_bar}$$

If

$$\text{privvmpages_3_bar} = a_1 * \text{privvmpages_1_bar} + a_2 * \text{privvmpages_2_bar}$$
$$\text{vmguarpages_3_bar} = a_1 * \text{vmguarpages_1_bar} + a_2 * \text{vmguarpages_2_bar}$$
$$a_1 + a_2 = 1, a_1 \geq 0, a_2 \geq 0$$

then the consistency rule is still being held

$$\text{privvmpages_3_bar} \geq \text{vmguarpages_3_bar}$$

So, the third configuration which is obtained as

$$[\text{THIRD_CONFIGURATION}] = a_1 * [\text{FIRST_CONFIGURATION}] +$$
$$a_2 * [\text{SECOND_CONFIGURATION}]$$
$$a_1 \geq 0, a_2 \geq 0, a_1 + a_2 = 1$$

obey the consistency rule.

Consider another rule

$$\text{tcp\textit{r}cvbuf_1_bar} \geq 64$$
$$\text{tcp\textit{r}cv)buf_2_bar} \geq 64$$

So, the third configuration which is obtained as

$[THIRD_CONFIGURATION] = a_1 * [FIRST_CONFIGURATION] + a_2 * [SECOND_CONFIGURATION]$
 $a_1 \geq 0, a_2 \geq 0, a_1 + a_2 = 1$

should obtain this rule too

$tcprcvbuf_3_bar = a_1 * tcprcvbuf_1_bar + a_2 * tcprcvbuf_2_bar \geq a_1 * 64 + a_2 * 64 = 64 * (a_1 + a_2) = 64$

because $a_1 + a_2 = 1$.

But subtraction doesn't guarantee that all rules are preserved.

Example

$5 > 3, 4 > 1$ but $(5 - 4) < (3 - 1)$ because $1 < 2$