

Re: Scheduler: Process priority fed back to parent?

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-03/4045.html>

From: Timothy Miller (miller_at_techsource.com)

Date: 03/16/04

Date: Tue, 16 Mar 2004 16:35:33 -0500

To: Eric <eric@cisunix.net>

Eric wrote:

> *On Tuesday 16 March 2004 10:46 am, Timothy Miller wrote:*

>

>> *Eric wrote:*

>>

>>> *Sort of like what windows does with its prefetch stuff? Have a directory*

>>> *that contains info about the best way to run a particular program and its*

>>> *memory layouts/ disk accesses and patterns?*

>>

>> *I'm not familiar with that aspect of Windows, but... sure. :)*

>

>

> *All i know is it does some sort of prefetch/caching of information to make*

> *user processes load faster.*

I'm not sure that this is quite the same. Mac OS X has a special cache on disk of things that get loaded on boot. It's arranged near the outer-most tracks of the disk, and it's stored linearly, so it can all be read as efficiently as possible. This saves a lot of time.

This isn't what I'm talking about, however. This caching idea is something that applies only to, for instance, loading system services, and that would all be in user space.

>> *I appreciate the problems, and the cost may be greater than the benefit.*

>> *But if the cache is just a file in the file system, then there are no*

>> *file-system dependencies.*

>

> *True. Read on for my thoughts about cost vs. benefit. The biggest costs would*

> *only be incurred the first couple times a process is launched, at least the*

> *cost of calculating what the heck the process is doing. After that it would*

> *only be using already gathered info.*

Yes and no. We'd probably want to keep a running average or a life-time average. We want to smooth out the bumps in program behavior over the

Linux–Kernel: Re: Scheduler: Process priority fed back to parent?

long–term, and we don't want unfortunate first–impressions to cause grudges. So, let's say that some program you install uses an inordinate amount of CPU for the first few days of its life because it's doing all sorts of initialization, but after that point, it behaves more sanely. We want th