

# How do you pre allocate memory to a process in solaris?

Asked 16 years, 3 months ago   Modified 10 years, 8 months ago

Viewed 4k times

---



1



My problem is:

I have a perl script which uses lot of memory (expected behaviour because of caching). But, I noticed that the more I do caching, slower it gets and the process spends most of the time in sleep mode.

I thought pre-allocating memory to the process might speed up the performance.

Does someone have any ideas here?

## Update:

I think I am not being very clear here. I will put question in clearer way:

I am not looking for the ways of pre-allocating inside the perl script. I dont think that would help me much here. What I am interested in is a way to tell OS to allocate X amount of memory for my perl script so that it does not have to compete with other processes coming in later.

Assume that I cant get away with the memory usage. Although, I am exploring ways of reducing that too but dont expect much improvement there. FYI, I am working on a solaris 10 machine.

unix

memory

solaris

Share

edited Sep 24, 2008 at 18:49

Improve this question

Follow

asked Sep 24, 2008 at 15:13



Jagmal

5,966 ● 9 ● 37 ● 35

---

That would make your process slower. By forcing sub-optimal memory use, other processes will need more I/O, making your process slower. – [David Schwartz](#) Sep 23, 2011 at 8:04

---

10 Answers

Sorted by:

Highest score (default)



4



What I gathered from your posting and comments is this:

- Your program gets slow when memory use rises
- Your pogram increasingly spends time sleeping, not computing.



Most likely eplanation: Sleeping means waiting for a resource to become available. In this case the resource



most likely is memory. Use the `vmstat 1` command to verify. Have a look at the `sr` column. If it goes beyond ~150 consistently the system is desperate to free pages to satisfy demand. This is accompanied by high activity in the `pi`, `po` and `fr` columns.

If this is in fact the case, your best choices are:

- Upgrade system memory to meet demand
- Reduce memory usage to a level appropriate for the system at hand.

Preallocating memory will not help. In either case memory demand will exceed the available main memory at some point. The kernel will then have to decide which pages need to be in memory now and which pages may be cleared and reused for the more urgently needed pages. If all regularly needed pages (the working set) exceeds the size of main memory, the system is constantly moving pages from and to secondary storage (swap). The system is then said to be thrashing and spends not much time doing useful work. There is nothing you can do about this except adding memory or using less of it.

[Share](#) [Improve this answer](#)

[Follow](#)

answered Sep 27, 2008 at 0:25



[hekr](#)

76 ● 2

---

Very helpful answer. Perhaps the best till now. I will try vmstat and update here if I could get a crack. I am upvoting it (sorry, I can upvote only once) – [Jagmal](#) Sep 27, 2008 at 16:38

---

From a comment:

2

The memory limitations are not very severe but the memory footprint easily grows to GBs and when we have competing processes for memory, it gets very slow. I want to reserve some memory from OS so that thrashing is minimal even when too many other processes come. Jagmal

Let's take a different tack then. The problem isn't really with your Perl script in particular. Instead, all the processes on the machine are consuming too much memory for the machine to handle as configured.

You can "reserve" memory, but that won't prevent thrashing. In fact, it could make the problem worse because the OS won't know if you are using the memory or just saving it for later.

I suspect you are suffering [the tragedy of the commons](#). Am I right that many other users are on the machine in question? If so, this is more of a social problem than a technical problem. What you need is someone (probably the System Administrator) to step in and coordinate all the processes on the machine. They should find the most

extravagant memory hogs and work with their programmers to reduce the cost on system resources. Further, they ought to arrange for processes to be scheduled so that resource allocation is efficient. Finally, they may need to get more or improved hardware to handle the expected system load.

Share Improve this answer

Follow

edited Sep 28, 2009 at 13:39



Roger Lipscombe

91.6k ● 59 ● 252 ● 396

answered Sep 26, 2008 at 19:17



Jon Ericson

21.5k ● 12 ● 102 ● 151

---

This was very helpful indeed. I am upvoting it (I wish I could vote for than once). – Jagmal Sep 27, 2008 at 16:36

---



0



Some questions you might ask yourself:

- are my data structures really useful for the task at hand?
- do I really have to cache that much?
- can I throw away cached data after some time?



Share Improve this answer

Follow

answered Sep 24, 2008 at 15:16



Thorsten79

10.1k ● 6 ● 40 ● 54

---

For now, the answers are Yes, Yes, No. And I am throwing unnecessary things asap. I have updated the question accordingly for clarity. – [Jagmal](#) Sep 24, 2008 at 15:19

---



0



```
my @array;  
$#array = 1_000_000; # pre-extend array to one  
million elements,  
#  
http://perldoc.perl.org/perldata.html#Scalar-  
values  
  
my %hash;  
keys(%hash) = 8192; # pre-allocate hash buckets  
# (same documentation section)
```

Not being familiar with your code, I'll venture some wild speculation here [grin] that these techniques aren't going to offer new great efficiencies to your script, but that the pre-allocation could help a little bit.

Good luck!

-- Douglas Hunter

Share Improve this answer

answered Sep 24, 2008 at 15:46

Follow



[douglashunter](#)

9 ● 2



0

I recently rediscovered an excellent [Randal L. Schwartz article](#) that includes preallocating an array. Assuming this



is your problem, you can test preallocating with a variation on that code. But be sure to test the result.



The reason the script gets slower with more caching might be [thrashing](#). Presumably the reason for caching in the first place is to increase performance. So a quick answer is: reduce caching.

Now there may be ways to modify your caching scheme so that it uses less main memory and avoids thrashing. For instance, you might find that caching to a file or database instead of to memory can boost performance. I've found that file system and database caching can be more efficient than application caching and can be shared among multiple instances.

Another idea might be to alter your algorithm to reduce memory usage in other areas. For instance, instead of pulling an entire file into memory, Perl programs tend to work better reading line by line.

Finally, have you explored [the Memoize module](#)? It might not be immediately applicable, but it could be a source of ideas.

Share Improve this answer

answered Sep 24, 2008 at 16:22

Follow



**Jon Ericson**

21.5k ● 12 ● 102 ● 151



I could not find a way to do this yet.

0

But, I found out that (See [this](#) for details)



Memory allocated to lexicals (i.e. `my()` variables) cannot be reclaimed or reused even if they go out of scope. It is reserved in case the variables come back into scope. Memory allocated to global variables can be reused (within your program) by using `undef()`ing and/or `delete()`.

So, I believe a possibility here could be to check if i can reduce the total memory print of lexical variables at a given point in time.

Share Improve this answer

Follow

answered Sep 25, 2008 at 12:38



Jagmal

5,966 ● 9 ● 37 ● 35



0

It sounds like you are looking for *limit* or *ulimit*. But I suspect that will cause a script that goes over the limit to fail, which probably isn't what you want.



A better idea might be to share cached data between processes. Putting data in a database or in a file works well in my experience.



I hate to say it, but if your memory limitations are this severe, Perl is probably not the right language for this application. C would be a better choice, I'd think.

Share Improve this answer

answered Sep 25, 2008 at 15:27



Follow



Jon Ericson

21.5k ● 12 ● 102 ● 151

---

The memory limitations are not very severe but the memory footprint easily grows to GBs and when we have competing processes for memory, it gets very slow. I want to reserve some memory from OS so that thrashing is minimal even when too many other processes come. – Jagmal Sep 26, 2008 at 1:32

---



One thing you could do is to use solaris zones (containers) .

0

You could put your process in a zone and allocate it resources like RAM and CPU's.



Here are two links to some tutorials :



1. [Solaris Containers How To Guide](#)



2. [Zone Resource Control in the Solaris 10 08/07 OS](#)

Share Improve this answer

answered Oct 1, 2008 at 6:49

Follow



cavver

130 ● 4



While it's not pre-allocating as you asked for, you may also want to look at the large page size options, so that when perl has to ask the OS for more memory for your program, it gets it in larger chunks.

0



See [Solaris Internals: Multiple Page Size Support](#) for more information on the difference this makes and how to





do it.

Share Improve this answer

answered Jan 12, 2009 at 0:29

Follow



alanc

4,170 ● 23 ● 24



0

Look at <http://metacpan.org/pod/Devel::Size>

You could also inline a c function to do the above.



As far as I know, you cannot allocate memory directly from Perl. You can get around this by writing an XS module, or using an inline C function like I mentioned.



Share Improve this answer

edited Apr 5, 2014 at 15:05

Follow



szabgab

6,262 ● 11 ● 54 ● 64

answered Sep 24, 2008 at 15:32



J.J.

4,892 ● 1 ● 26 ● 29