[i]Exercice 1

1 *Ecrire une synthèse en français d'une vingtaine de lignes environ du document ci-dessous.*
2 *Calculer la valeur d'inter-check delay fournie par Nagios, dans le cas où on aurait 750 vérifications à faire : 500 vérifications toutes les 5 minutes et 250 toutes les 2 minutes. Vérifier la validité de la valeur de l'inter-check-delay que vous avez trouvé. Quelle erreur a été commise ?*
3 *On considère un axe horizontal représentant le temps (de 0 à 8 minutes). Placer sur cet axe des repères indiquant à quel moment sont vérifiés les services (on considère l'exemple précédent avec les 500 et 250 services).*

Inter-Check Delay

As mentioned before, Nagios attempts to equalize the load placed on the machine that is running Nagios by equally spacing out initial service checks. The spacing between consecutive service checks is called the inter-check delay. By giving a value to the service_inter_check_delay_method variable in the main config file, you can modify how this delay is calculated. I will discuss how the "smart" calculation works, as this is the setting you will want to use for normal operation.

When using the "smart" setting of the service_inter_check_delay_method variable, Nagios will calculate an inter-check delay value by using the following calculation:

inter-check delay = (average check interval for all services) / (total number of services)

*Let's take an example. Say you have 1,000 services that each have a normal check interval of 5 minutes (obviously some services are going to be checked at different intervals, but let's look at an easy case...). The total check interval time for all services is 5,000 (1,000 * 5). That means that the average check interval for each service is 5 minutes (5,000 / 1,000). Given this information, we realize that (on average) we need to re-check 1,000 services every 5 minutes. This means that we should use an inter-check delay of 0.005 minutes (0.3 seconds) when spacing out the initial service checks. By spacing each service check out by 0.3 seconds, we can somewhat guarantee that Nagios is scheduling and/or executing 3 new service checks every second. By spacing the checks out evenly over time like this, we can hope that the load on the local server that is running Nagios remains somewhat balanced.*

Service Interleaving

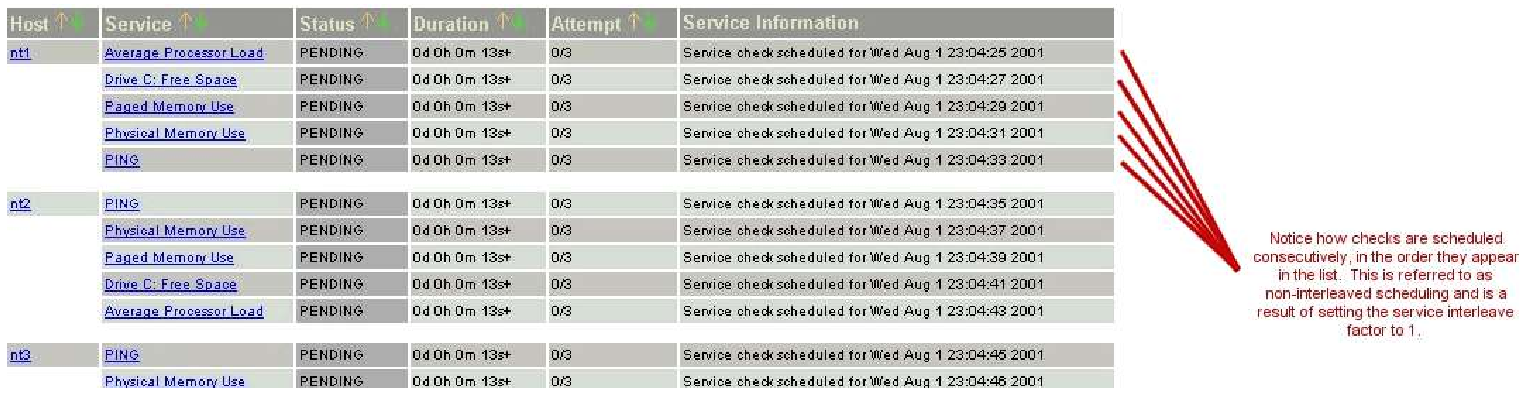
As discussed above, the inter-check delay helps to equalize the load that Nagios imposes on the local host. What about remote hosts? Is it necessary to equalize load on remote hosts? Why? Yes, it is important and yes, Nagios can help out with this. Equalizing load on remote hosts is especially important with the advent of service check parallelization. If you monitor a large number of services on a remote host and the checks were not spread out, the remote host might think that it was the victim of a SYN attack if there were a lot of open connections on the same port. Plus, attempting to equalize the load on hosts is just a nice thing to do...

By giving a value to the service_interleave_factor variable in the main config file, you can modify how the interleave factor is calculated. I will discuss how the "smart" calculation works, as this will probably be the setting you will want to use for normal operation. You can, however, use a pre-set interleave factor instead of having Nagios calculate one for you. Also of note, if you use an interleave factor of 1, service check interleaving is basically disabled.

When using the "smart" setting of the service_interleave_factor variable, Nagios will calculate an interleave factor by using the following calculation:
interleave factor = ceil (total number of services / total number of hosts)

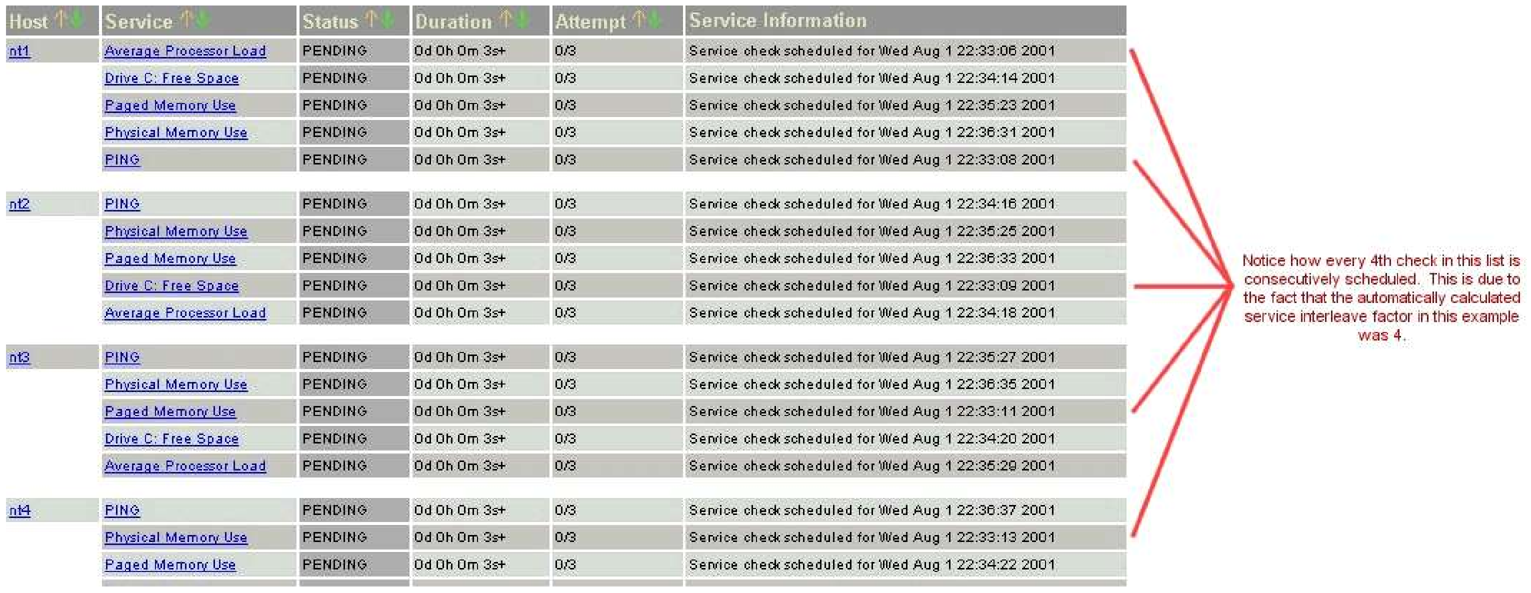
Let's take an example. Say you have a total of 1,000 services and 150 hosts that you monitor. Nagios would calculate the interleave factor to be 7. This means that when Nagios schedules initial service checks it will schedule the first one it finds, skip the next 6, schedule the next one, and so on... This process will keep repeating until all service checks have been scheduled. Since services are sorted (and thus scheduled) by the name of the host they are associated with, this will help with minimizing/equalizing the load placed upon remote hosts.

The images below depict how service checks are scheduled when they are not interleaved (service_interleave_factor=1) and when they are interleaved with the service_interleave_factor variable equal to 4.



Host	Service	Status	Duration	Attempt	Service Information
nt1	Average Processor Load	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:25 2001
	Drive C: Free Space	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:27 2001
	Paged Memory Use	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:29 2001
	Physical Memory Use	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:31 2001
	PING	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:33 2001
nt2	PING	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:35 2001
	Physical Memory Use	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:37 2001
	Paged Memory Use	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:39 2001
	Drive C: Free Space	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:41 2001
	Average Processor Load	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:43 2001
nt3	PING	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:45 2001
	Physical Memory Use	PENDING	0d 0h 0m 13s+	0/3	Service check scheduled for Wed Aug 1 23:04:46 2001

Notice how checks are scheduled consecutively, in the order they appear in the list. This is referred to as non-interleaved scheduling and is a result of setting the service interleave factor to 1.



Host	Service	Status	Duration	Attempt	Service Information
nt1	Average Processor Load	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:33:06 2001
	Drive C: Free Space	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:34:14 2001
	Paged Memory Use	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:35:23 2001
	Physical Memory Use	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:36:31 2001
	PING	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:33:08 2001
nt2	PING	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:34:16 2001
	Physical Memory Use	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:35:25 2001
	Paged Memory Use	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:36:33 2001
	Drive C: Free Space	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:33:09 2001
	Average Processor Load	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:34:18 2001
nt3	PING	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:35:27 2001
	Physical Memory Use	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:36:35 2001
	Paged Memory Use	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:33:11 2001
	Drive C: Free Space	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:34:20 2001
	Average Processor Load	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:35:29 2001
nt4	PING	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:36:37 2001
	Physical Memory Use	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:33:13 2001
	Paged Memory Use	PENDING	0d 0h 0m 3s+	0/3	Service check scheduled for Wed Aug 1 22:34:22 2001

Notice how every 4th check in this list is consecutively scheduled. This is due to the fact that the automatically calculated service interleave factor in this example was 4.