

Subsysteem dat instaat voor het loggen

(Gebruikersprogramma & socket vs. Agent)

Motivatie

Wanneer we een programma selectief tracen zal het nodig zijn om de log op schijf op te slaan. Dit kan ofwel door de Agent zelf afgehandeld worden, wat zeker het performantste is. Ofwel kunnen alle gegevens naar de gebruikersapplicatie doorgestuurd worden, en daar opgeslagen worden. Het is mogelijk dat als de bottleneck bijna volledig op de schijf toegang zit, dat we de gegevens op een min of meer efficiënte manier naar de client kunnen doorsturen. Dit maakt het controleren van de resultaten makkelijker.

Referentiesysteem

De tests werden uitgevoerd op een systeem met:

- 1.6 Ghz. mobile processor.
- 512 Mib RAM – SWAP UITGESCHAKELD.
- Linux met ReiserFS.

Oplossingen & tests

In de eerste plaats testen we een c-programma dat blokken van 1 Mib. genereert en deze rechtstreeks naar de schijf wegschrijft. Met deze test wordt steeds 512 Mib. weggeschreven en de benodigde tijd gemeten.

19.741 ms.
19.153 ms.
16.102 ms.
17.754 ms.
16.897 ms.
15.506 ms.
13.171 ms.
17.906 ms.
19.932 ms.
16.732 ms.
18.331 ms.
17.832 ms.
14.861 ms.

In de tweede test genereert een c-programma ook blokken van 1 Mib, en stuurt deze door via een TCP socket. Een serverapplicatie ontvangt deze data en slaat deze dan op in een bestand. Beide componenten meten de tijd die hiervoor nodig is.

<i>Server</i>	<i>Generator</i>
20.778 ms.	20.035 ms.
22.980 ms.	22.509 ms.
22.851 ms.	21.826 ms.
23.750 ms.	23.207 ms.

Conclusie

De resultaten schommelen nogal doordat er veel gebufferd wordt. Zelfs met de swap uitgeschakeld schommelen de resultaten nog serieus. Wel is duidelijk te zien dat de socket voor een serieuze overhead. Als de enkel de agent instaat voor het wegschrijven is ook het cpu-verbruik zeer laag (zit niet in de beschrijving opgenomen).