

UMEÅ UNIVERSITET
Institutionen för Datavetenskap

13 november 2018

5DV088: Systemnära programmering
Hösten 2018, 7.5p

Reflektioner över Laboration 4 - mfind

Name Buster Hultgren Wörn
E-mail dv17bhn@cs.umu.se

Kursansvarig
Mikael Rännar
Handledare

Klas af Geijerstam, Klas af Geijerstam, Elias Åström

1 Trådsäkerhet

Algoritmen för en tråd, då den ska söka igenom katalogträdet är byggd med två huvuddelar. Först och främst så hämtar varje tråd ett objekt, som är en katalog, ur kön för att sedan leta i denna katalog.

Trådarna delar gemensamt minne för kön, och den första risken mot trådsäkerheten är ifall två trådar samtidigt försöker nå åt kön. För att stoppa detta så är ett mutex lås inlagt varje gång en tråd använder köns gränssyta.

Med denna lösning så uppkommer ett nytt problem - en tråd har plockat ut det första objektet ur kön och arbetar med denna (sin andra huvuddel). I objektet finns det fler kataloger att lägga till i kön, men tråden har inte hunnit dit än. En ny tråd kommer för att hämta ett objekt ur kön, men kön är då tom. Detta skulle vanligtvis tolkas som att det inte finns några kataloger att söka igenom, och att algoritmen är färdig.

Detta löses genom en semaphor som håller reda på hur många kataloger som ligger i kön. Semaphoren initieras med hur många kataloger som först läggs in i kön. Varje gång en tråd ska plocka ut en katalog ur kön, så räknar semaphoren ner. Varje gång ett objekt läggs till så räknar semaphoren upp.

Men vad händer då en tråd har sökt igenom den sista katalogen? Då kommer semaphoren vara nere på 0, och resterande trådar väntar på att köra. Det är här en andra räknare, en global variabel, kommer in. Varje gång en tråd söker igenom en katalog så ökas räknaren med ett, och då tråden sökt igenom katalogen så minskas räknaren med ett.

När en tråd sökt igenom en katalog undersöker den ifall kön är tom och om det finns några trådar som kör. Då kön är tom, och inga trådar kör, så har den sista katalogen sökts igenom. Semaphoren räknas då uppåt, och nästa tråd kör igenom. Den tråden märker samma sak: kön är tom och inga trådar kör. Tråden räknar då upp semaphoren för att sedan avsluta, och så avslutar alla trådar en efter en.

2 Testfall

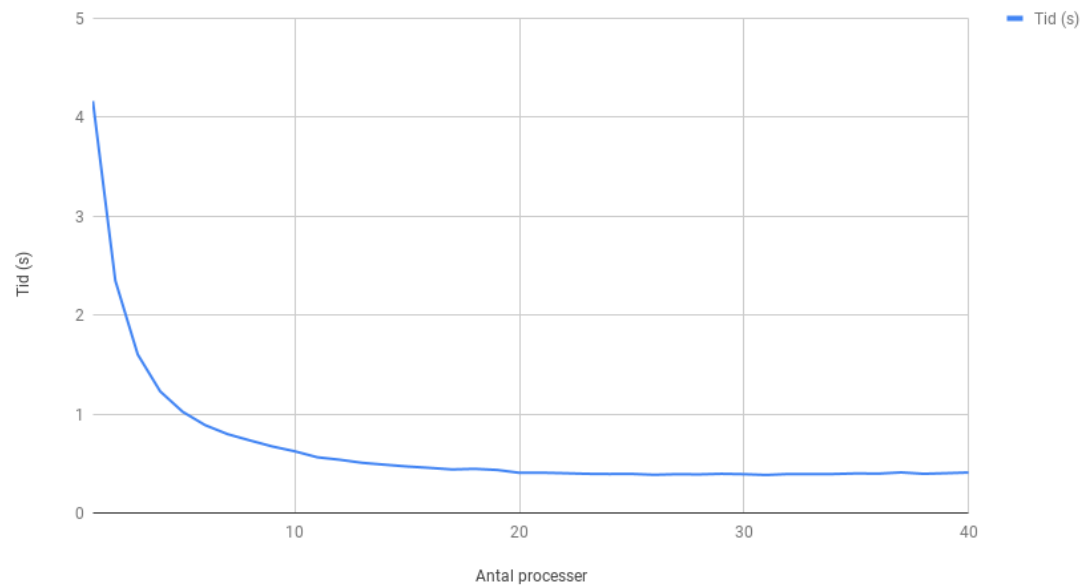
Mfind testkördes med olika mängder trådar för att se om det blev någon skillnad i prestanda. Programmet kördes på scratchy med följande argument:

```
./mfind -p i /pkg comsol
```

där *i* är antalet trådar.

Enligt specifikationerna så skulle 10 olika testfall göras, där varje testfall ökar med en tråd. Alltså testas programmet med trådirtervallet 1-10. Här gjordes testet med intervallet 1-40, för att kunna få en större bild av hur mfind påverkas med en större mängd trådar. Resultaten finns i Figur 1.

Tid (s) mot Antal processer



Figur 1: Testresultat: mfind med trådintervall 1-40

Som Figur 1 visar så ökar prestandan drastiskt fram till ca 10 trådar. Efter detta så verkar antalet trådar få mindre påverkan på prestandan.

Tidskomplexiteten för programmet verkar därmed vara följa funktionen $f(t) = kt^{-m}$, där k är en konstant, t är antalet trådar och m är gradtalet som utgör hur snabbt prestandan påverkas med antalet trådar. Detta är dock inte testat för mer trådar, och risken finns att desto fler trådar som används desto lägre blir prestandan.