UMEÅ UNIVERSITET
Instituionen förDatavetenskap

26 oktober 2018

5DV088: Systemnära programmering
Hösten 2018, 7.5p

Reflektioner över

Laboration 4 - mfind

Name Buster Hultgren Wärn
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 katalog trädet 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 gemensammt 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 gränsyta.

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.

Lösningen till detta är att använda en global variabel (trådarna delar minne på denna, och även denna variabel har ett mutex lås) som räknar hur många trådar som aktivt söker igenom en katalog. Då en tråd börjar söka igenom en katalog så ökas denna räknare med ett, och när tråden sökt igenom katalogen så minskas räknaren med ett. Då en tråd finner att kön är tom så kollar tråden ifall det finns några andra trådar som kör - ifall räknaren har ett värde större en noll. Då det finns andra trådar som kör väntar tråden, i ett condition lås, på en signal innan den igen kollar igenom kön. Varje gång en tråd hittar ett objekt att lägga till i kön så skickas en signal till condition låset som släpper på en tråd. Denna tråd plockar då ut det nya objektet ur kön och börjar arbeta med detta.

Mot slutet av algoritmen så bör alla trådar förutom en sitta kvar i condition låset. Då den sista tråden inte finner några fler objekt att lägga till i kön så går den tillbaka till sin första huvuddel - att hämta ett objekt ur kön. Då både kön är tom och inga fler trådar kör så är algoritmen i princip färdig - alla möjliga kataloger är igenomsökta. Då skickar tråden en brodcast till alla andra trådar att de kan släppas från condition låset. Dessa trådar försöker då plocka upp ett nytt objekt, men kön är tom. De hamnar dock inte i condition låset igen då de ser att det inte finns någon tråd som kör.

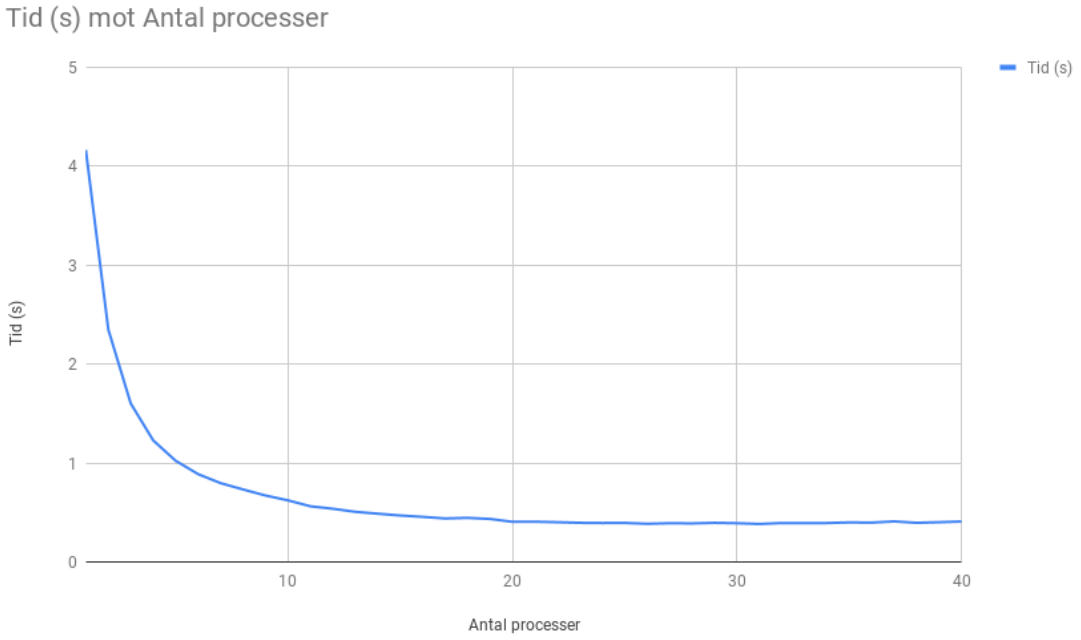
2 Testfall

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 trådintervallet 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.



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.

Tidskomplexiten 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 högre blir prestandan.