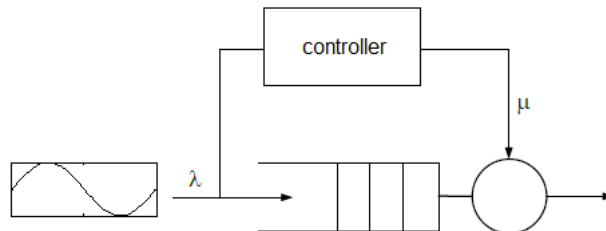


# Populärvetenskaplig sammanfattning av examensarbetet Händelsebaserad reglering av serversystem.

Elisabeth Jeppsson Krell  
Jennie Olsson

Målet med examensarbetet är att undersöka om händelsebaserad reglering av serversystem är möjlig, och om det kan förbättra systemets prestanda.

Ett enkelt serversystem, som kallas för ett M/M/1-system, består av en kö som är oändligt lång och en server. Det kommer in uppgifter till systemet som sedan bearbetas i servern. Om servern är upptagen placeras de inkomna uppgifterna sist i kön.



Figur 1: Vårt utvidgade M/M/1-system.

Bilden här ovanför visar hur vårt utvidgade M/M/1-system ser ut. Den första delen av systemet är kön som har ankomsthastigheten  $\lambda$ , den anger alltså hur ofta det kommer en uppgift till systemet. Den andra delen av systemet är servern som hanterar de inkomna uppgifterna med en viss hastighet, servicehastigheten  $\mu$ .

I examensarbetet så varierar ankomsthastigheten över tiden, till skillnad från det vanliga systemet där hastigheten är konstant. För att kompensera för den varierande ankomsthastigheten kan man göra servern dynamisk. Detta görs

genom att lägga till en regulator i systemet som kan styra serverns hastighet och öka eller minska den vid behov. För att kunna göra detta måste ankomsthastigheten vara känd. Tyvärr så går den inte att mäta vilket utgör ett nytt problem som måste lösas.

Dessa problem kan lösas händelsebaserat vilket innebär att när en ny uppgift anländer till systemet så bestäms en ny servicehastighet. Detta kan jämföras med att servicehastigheten istället bestäms vid jämna mellanrum, vilket är mycket vanligare.

En problematisk del är att när man ska ändra servicehastigheten tar det en viss tid att genomföra ändringen. Detta måste man ta hänsyn till då man utformar sin regulator. Med hjälp av mätbar information från systemet kan olika beräkningar göras och på så vis kan den nya servicehastigheten bestämmas.

Som en referenspunkt har vi valt att 85% av serverns kapacitet ska användas för att kunna uppnå ett optimalt resultat. Det betyder att servern i snitt används till 85%, och vi får en marginal som gör att vi kan använda lite mer av servern om det skulle behövas, exempelvis om kön skulle bli väldigt lång. Då vill vi öka serverns hastighet för att kunna beta av kön snabbare och det är möjligt om det finns lite kvar av servern att använda. När man använder denna procentsats så får man ett väl fungerande system med den dynamiska kapaciteten som vi var ute efter.

Den dynamiska kapaciteten används bland annat i datormoln där en användare kan hyra kapacitet efter behov. Fördelen med datormoln är att du kan hyra mer kapacitet då efterfrågan ökar och hyra mindre då efterfrågan minskar, vilket ger upphov till det dynamiska beteendet. Exempel på tjänster som använder datormoln är Googles Gmail och Dropbox.