# Introduktion

Denne performance test rapport har til formål at belyse evt. performance problemer og forbedringer med TM12 projektet.

Rapporten er en initial analyse og afviklingen er derfor foretaget på en udviklings-PC. Både SUT og selve tist-miljøet bliver eksekveret på samme PC. Dette er naturligvis ikke optimalt, da belastningen fra test-miljø’et vil reducere i reduceret performance i SUT.

Der bliver udført tre test.

1. En før-test til at vise hvordan belastningen er når TM12 er i hvile.
   1. Dette vil vise om systemet allokerer data eller anvender betydelig CPU-kraft på at lave ingenting. Der skal naturligvis anvendes lidt på evt. problemer eller ineffektiv design, da
2. En baseline test
   1. Denne test påtrykker et realistisk load på systemet og overvåger de relevante metrikker, som så senere kan evalueres for hvor der er mulighed for at forbedre performance, og hvor der evt. er flaskehalse. Denne test kan også bruges til at evaluere performance forbedringer senere hen.
3. En efter test
   1. Dette vil vise om systemet er i stand til at vende tilbage til hvile-tilstand, eller om der er ressourcer der ikke korrekt ryddes op.

# Test setup

## Hardware

CPU: Intel® Core™ i7-2640M CPU @2,80GHz

RAM: 4GB

OS: Windows 7 Professional SP1 – 64bit

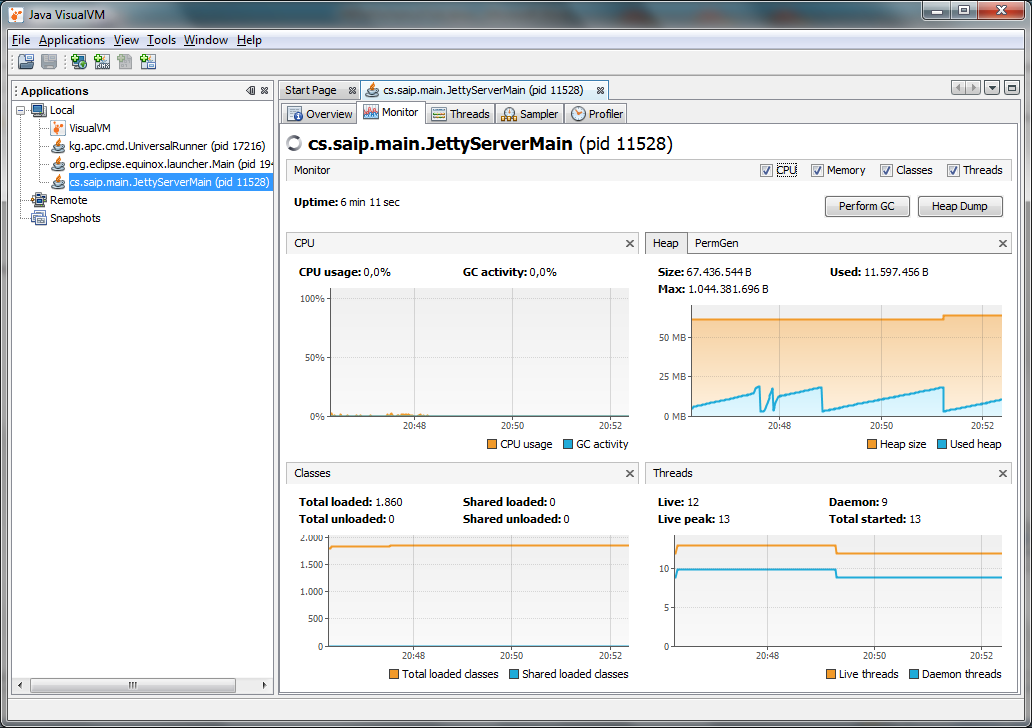
HDD: SSD

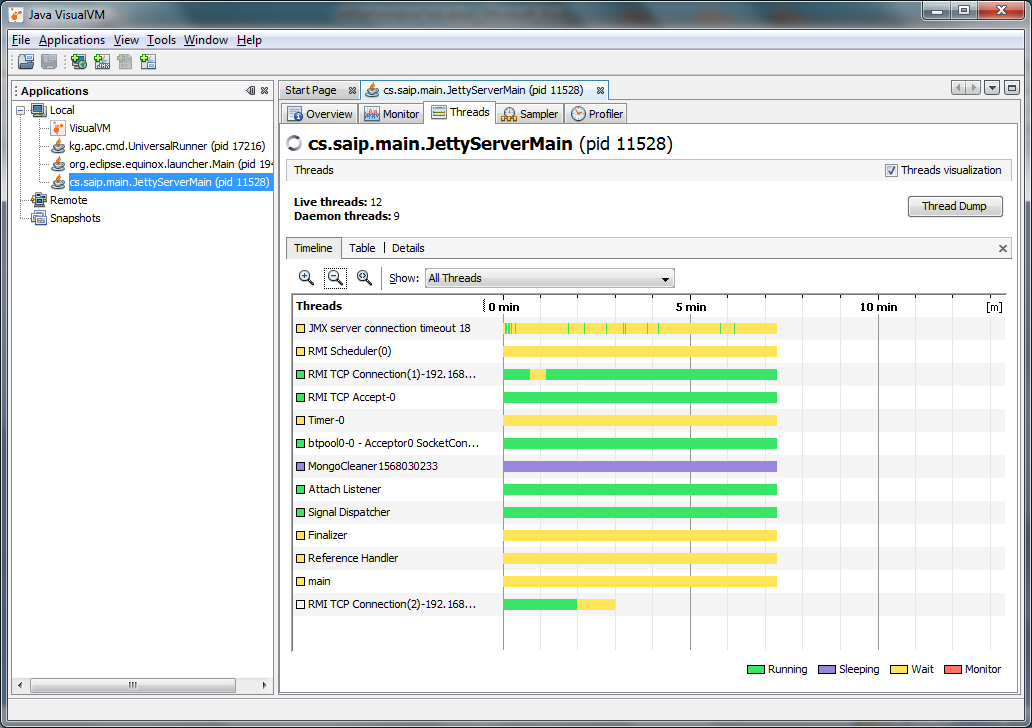
## TM12

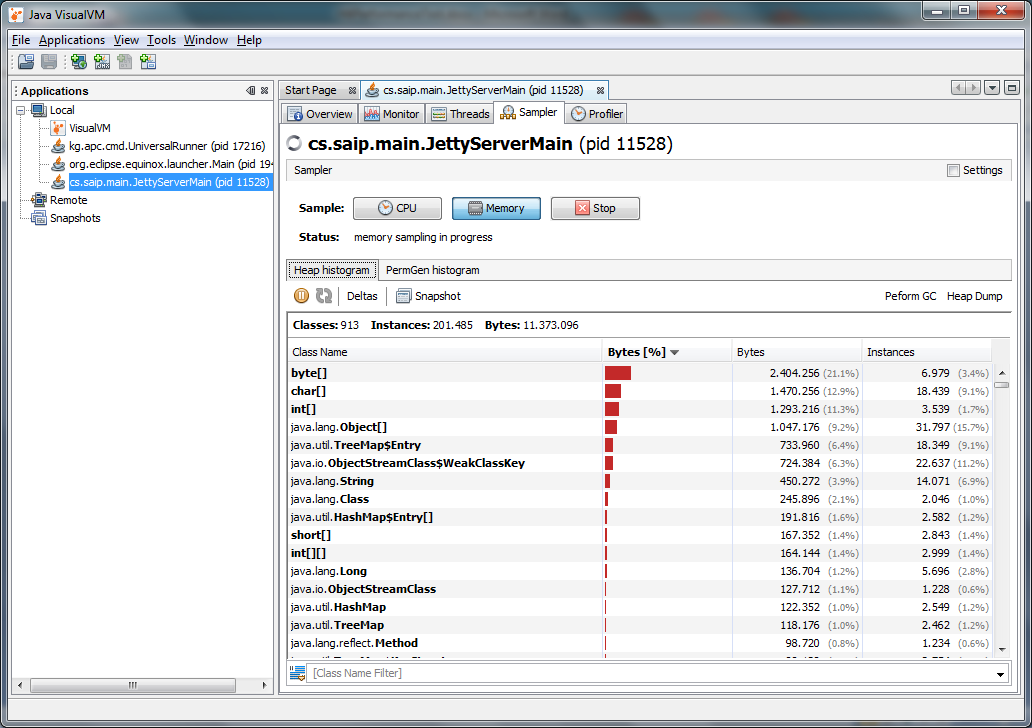
Mongodb enabled

# Før test

Til denne test anvendes jvisualvm til at beregne hvor meget RAM og CPU systemet anvender. Samtidig vil der blive evalueret på hvordan ressourceforbruget er og om der allokeres nogle ressourcer. Da systemet er i hvile bør det være muligt at allokere meget få ressourcer løbende. Hvis systemet er designet til lange perioder med ingen aktivitet, og ressourcerne i hvile skal minimeres, vil det være muligt at allokere de nødvendige ressourcer én gang, i stedet for at oprette og nedlægge dem. Det vil dog være helt op til hvad der er målet for performance.

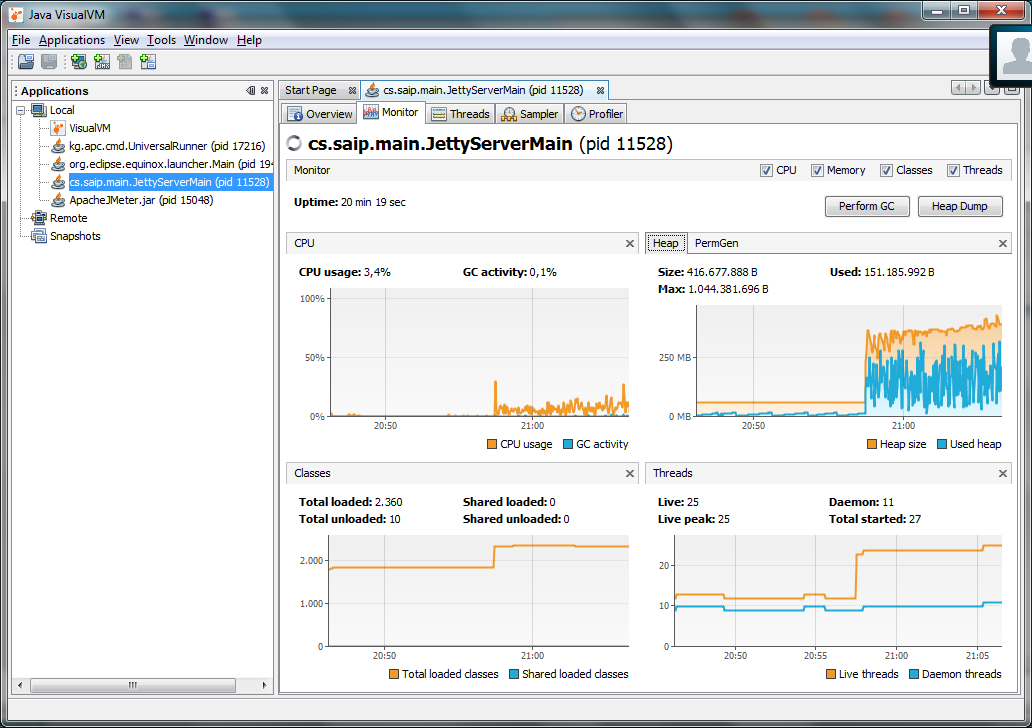


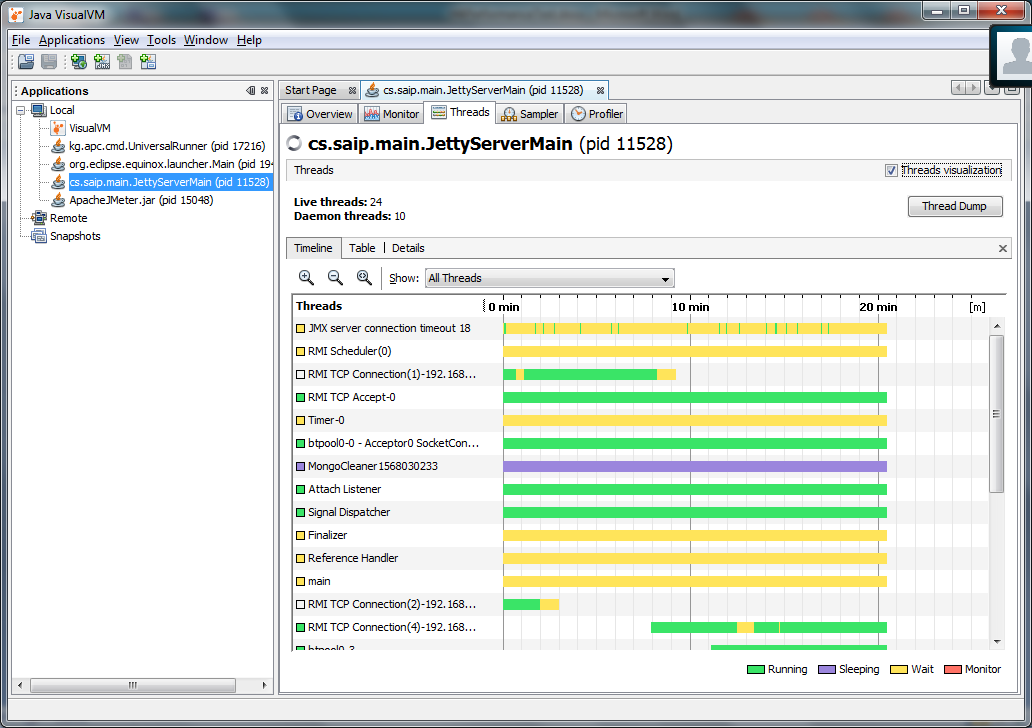


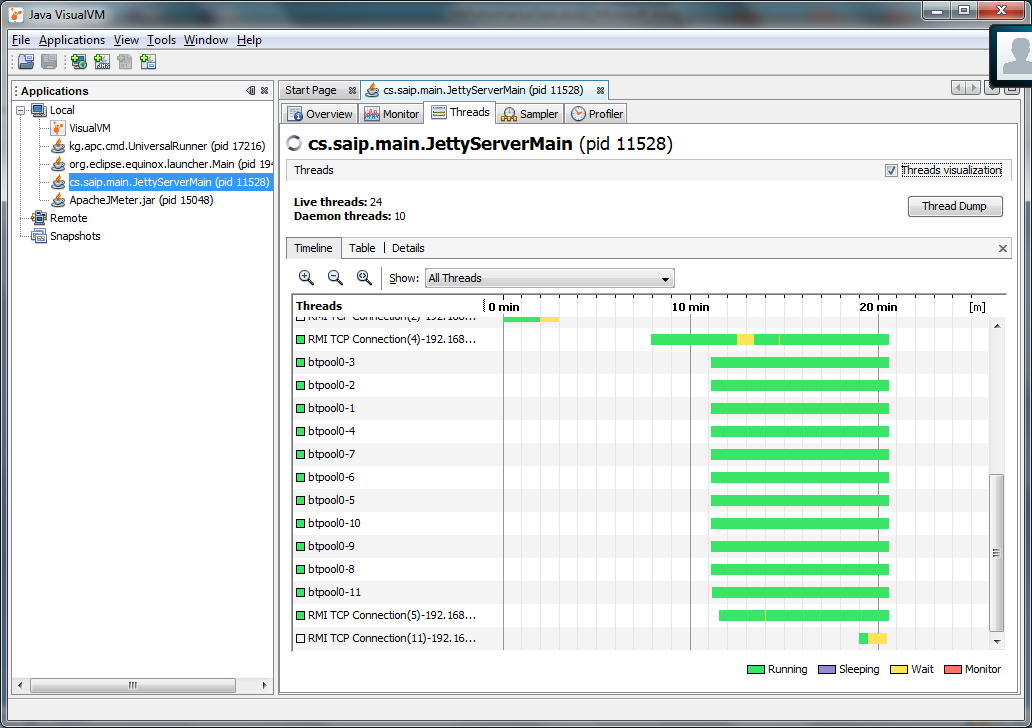


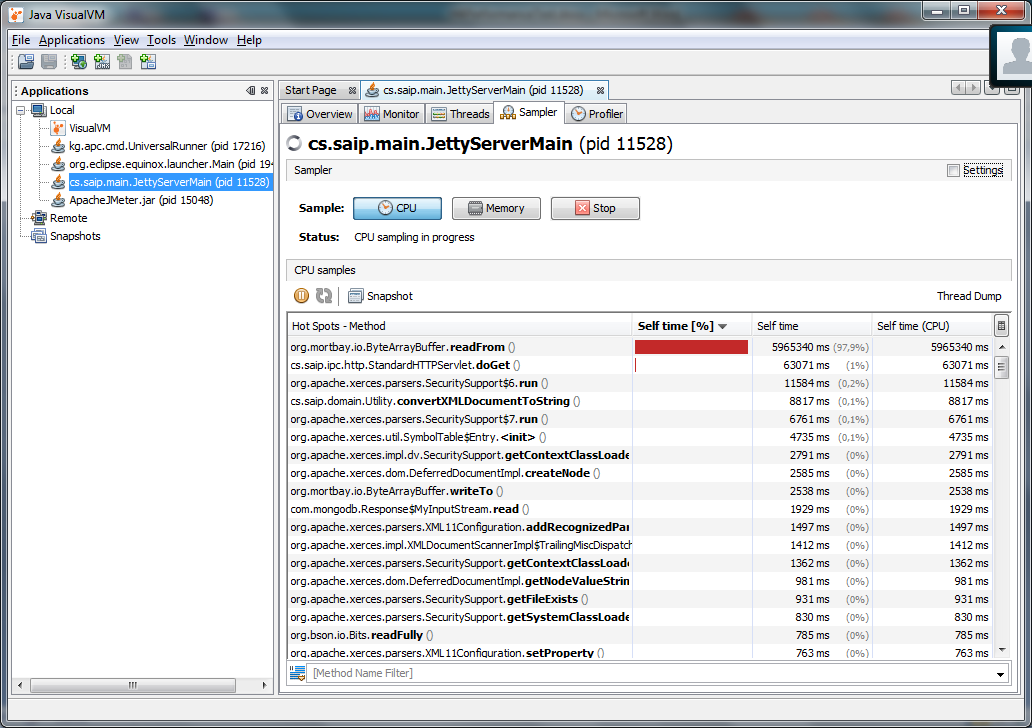
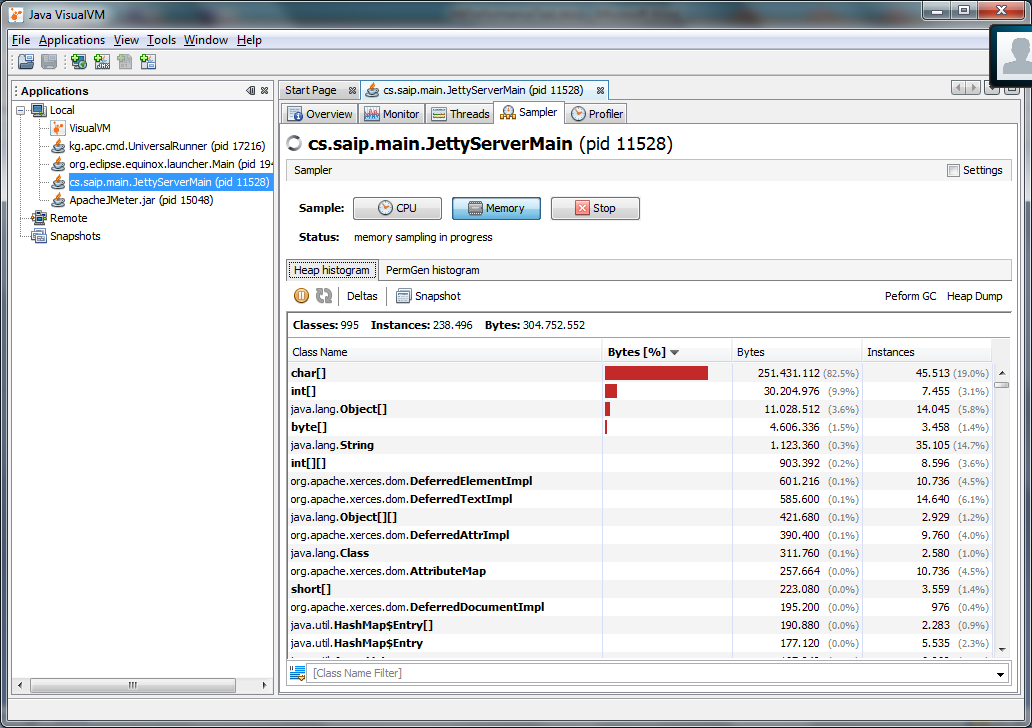
# Baseline test

Baseline testen baserer sig på et test script for vores test-afdeling, og det er designet til at udøve en realistisk belastning af systemet. Denne test anvender jvisualvm og JMeter, samt Loadosophia til at generer test rapporten.



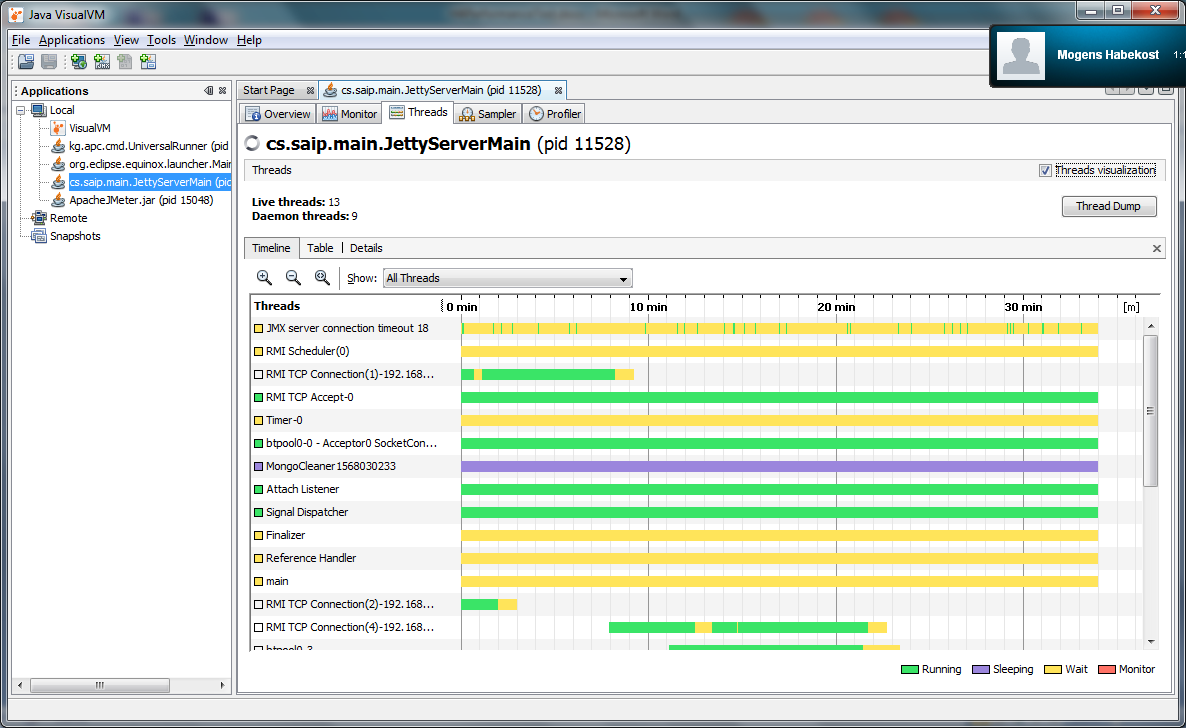


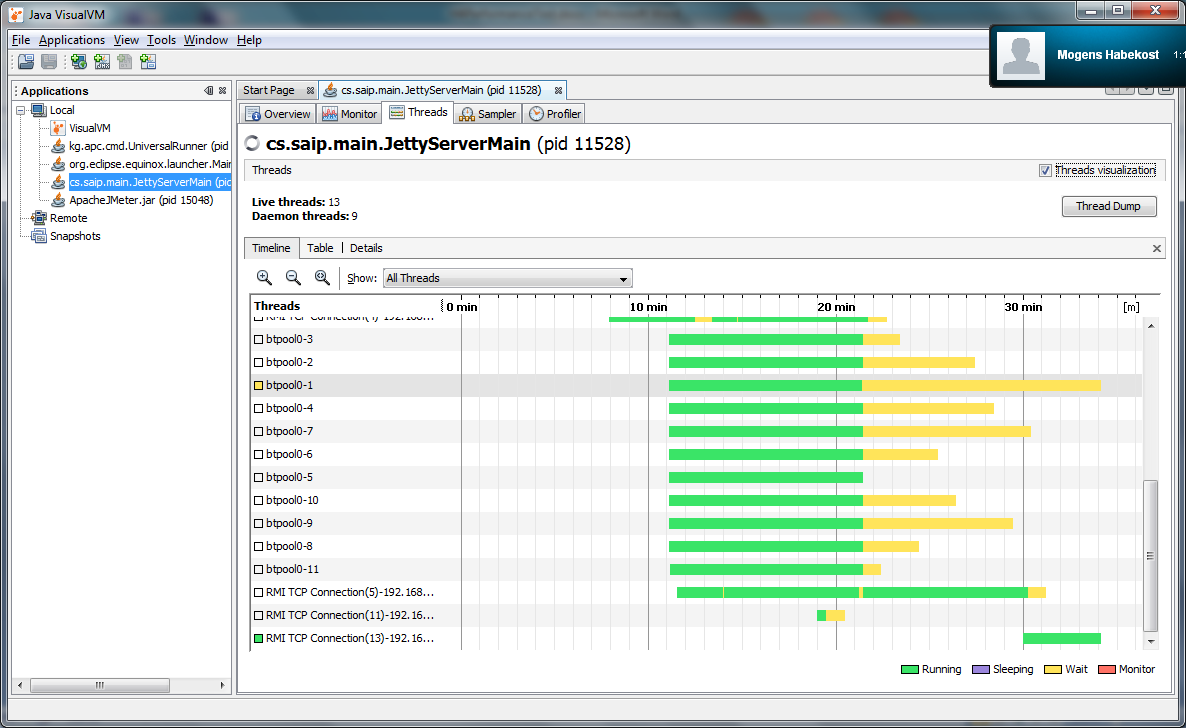


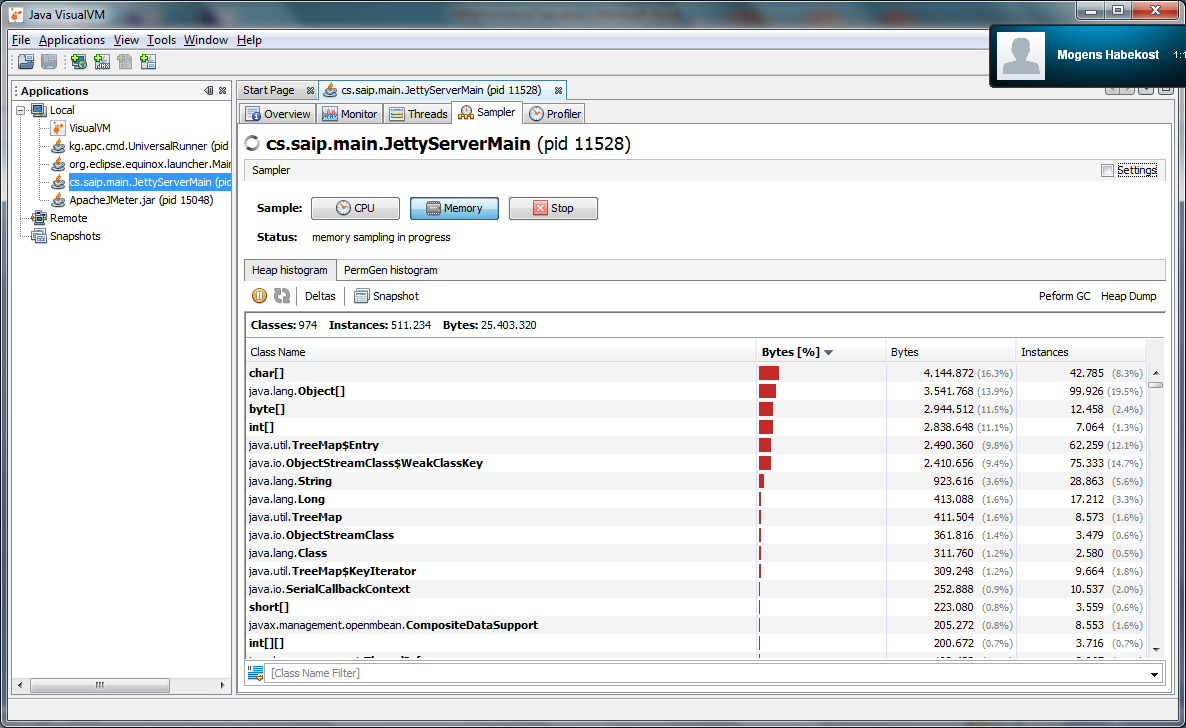
 

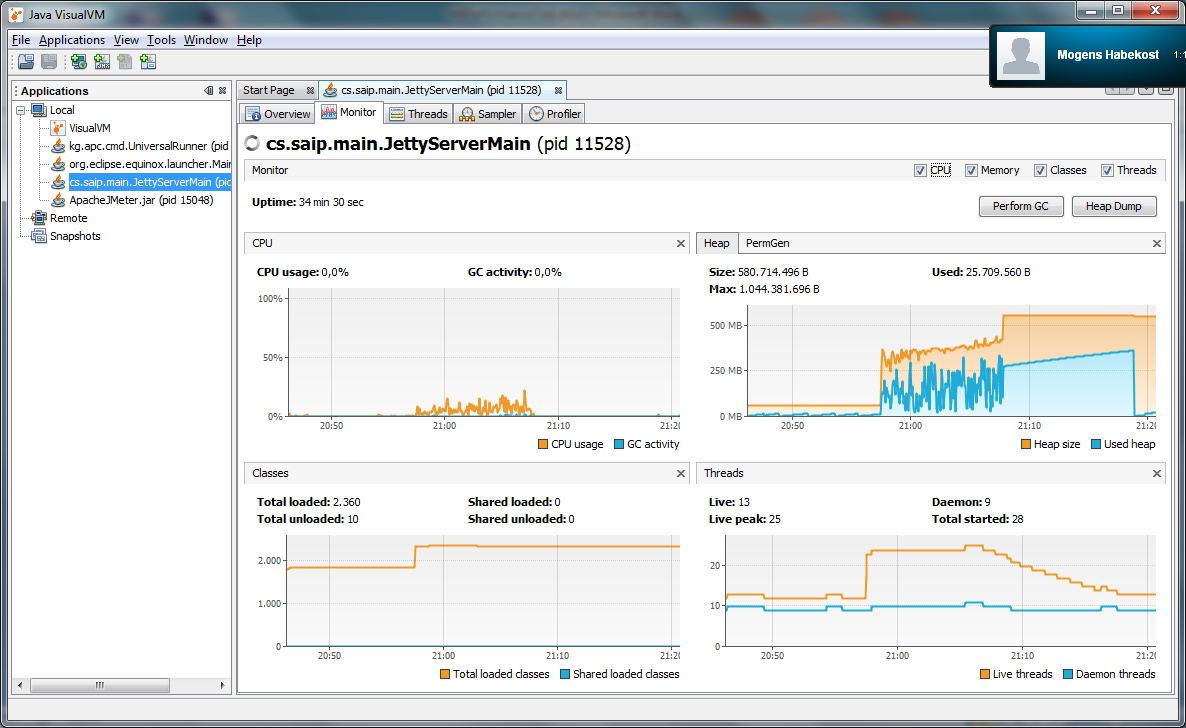
# Efter test

GC manuelt eksekveret for at se hvile efter test. Dette er gjort efter 10min med hvile og er tydelig i resultatet.









# Konklusion

Uden at kende de konkrete performance krav til systemet er det naturligvis umuligt at give nogle rigtig fornuftige anbefalinger, da det alt sammen vil være under forudsætning af det HW der skal eksekvere systemet og den load som systemet skal kunne håndtere, inklusiv den peak den forventes at kunne håndtere og den evt. forøgelse af load med tiden.

Hvis vi laver følgende antagelser:

1. Test afdeling har afleveret et test-script der repræsenterer normalsituationen.
2. Server PC’en vil være mindst ligeså kraftig som udviklings-PC’en på alle parametre
3. Server PC’en eksekverer både application og DB (ligesom udviklings PC’en) og er dedikeret (ingen andre applikationer eller requests skal serviceres).

Dette er nok ikke særlig realistisk, men under disse forudsætninger kan vi konkludere at