

Naloga 3

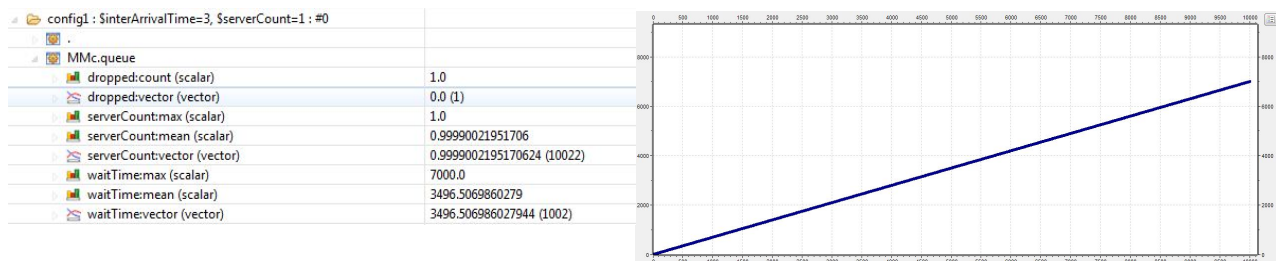
Domačo nalogo 2 priredimo po dani predlogi:

- implementiramo signala za čakalni čas in števec uporabljenih resursov (serverCount)
- implementiramo dodatni selfmessage, ki se bo prožil cca na vsako sekundo, da bomo periodično izvajali serverCount, s čimer signaliziramo količino porabljenih resursov. Ta način je boljši kot preverjanje ob vsakem sporočilu, saj nam poda zasedenost odvisno od časa.
- Čakalne čase sporočamo istočasno kot sprejemamo sporočila v obdelavo. Če je ob prihodu sporočila v strežnik dovolj prostih resursov je čakalni čas = 0s, drugače se sporoči primeren čakalni čas

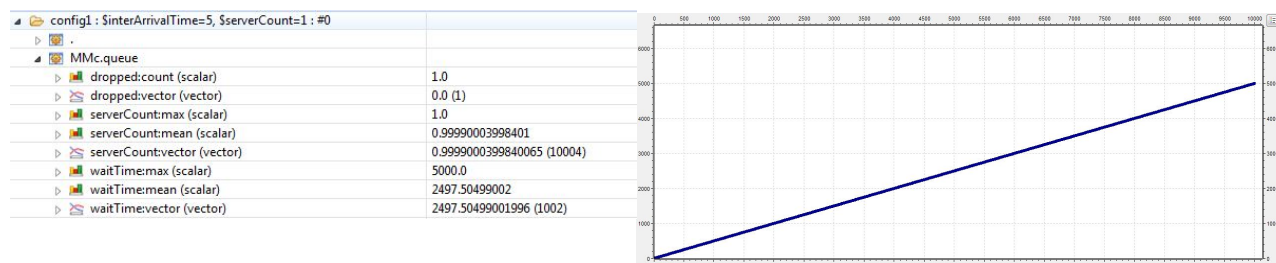
Ob izvajanju vseh devetih možnih kombinacij opazimo naslednje grafe:

Poizkus z enim strežnikom:

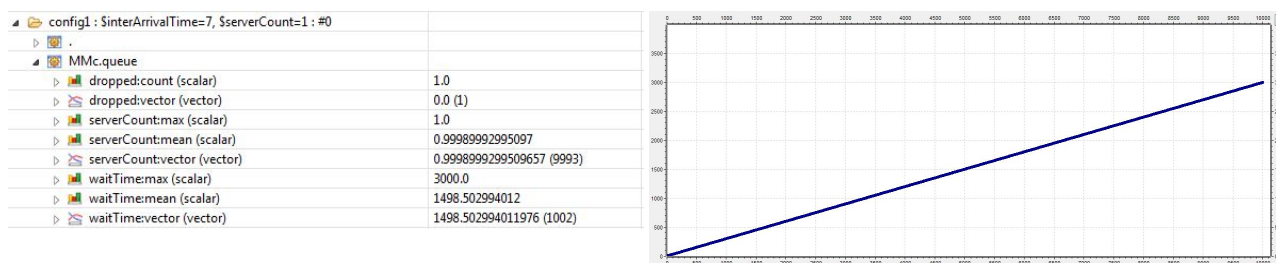
- medprihodni čas 3s: strežnik je neprenehoma zaseden, čakalni čas linearno narašča v neskončnost (strežniku ne uspeva reševati sporočil pravočasno)



- medprihodni čas 5s: strežnik še vedno polno zaseden, čakalni čas linearno narašča v neskončnost



- medprihodni čas 7s: strežnik še vedno polno zaseden, čakalni čas linearno narašča v neskončnost

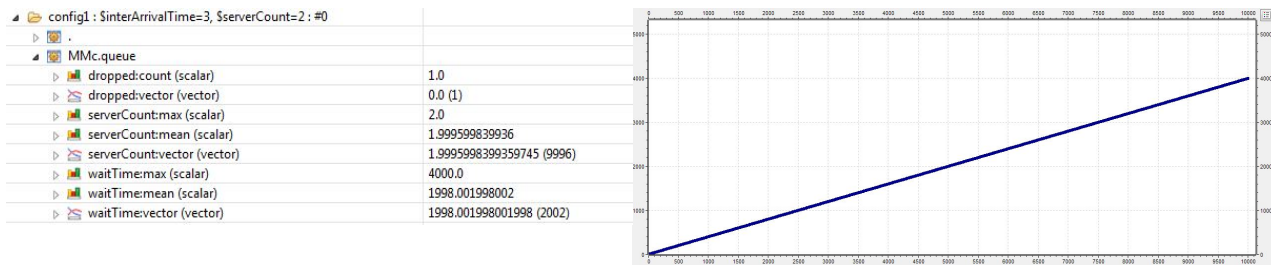


Rezultati za en strežnik so pričakovani, vemo da strežnik obdeluje sporočila na vsake 10s, iz česar sledi da ni sposoben pravočasno obdelati nobenega od medprihodnih časov pod 10s.

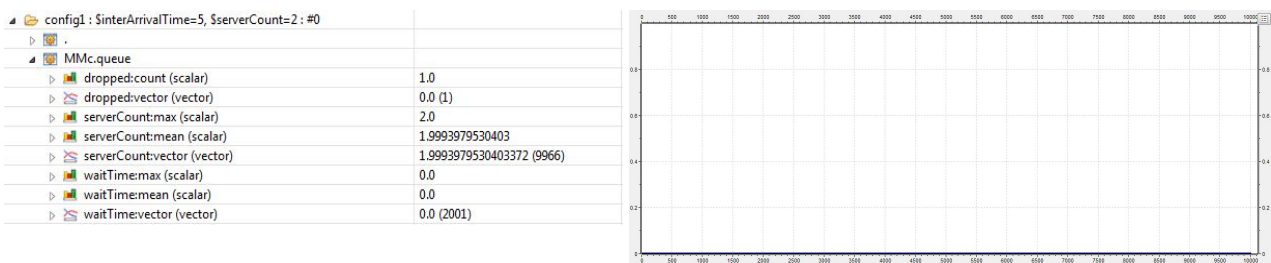
Poizkus z dvema strežnikoma:

Efektivna moč dveh strežnikov, ki vsak obdela sporočilo v 10s, je eno sporočilo na 5s, po čemer bi sklepali, da bosta pri medprihodnem času 3s odpovedala, pri 7s situacijo obvladala, zanimiv pa bo primer z medprihodnim časom 5s.

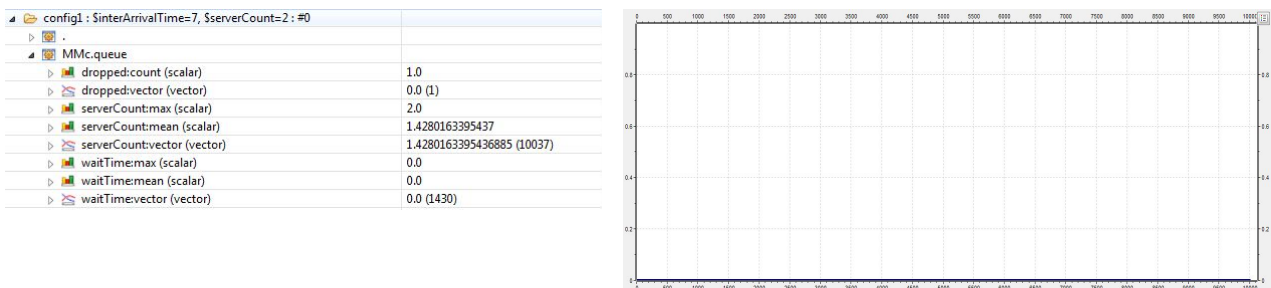
- medprihodni čas 3s: po pričakovanju odpovesta



- medprihodni čas 5s: situacijo KOMAJDA obvladata, saj vidimo da je zasedenost ZELO blizu 2 (serverCountje 1.999). Pa vendar se vrsta niti ne pojavlja.



- medprihodni čas 7s: serverja rešujeta sporočila pravočasno in nista polno zasedena.

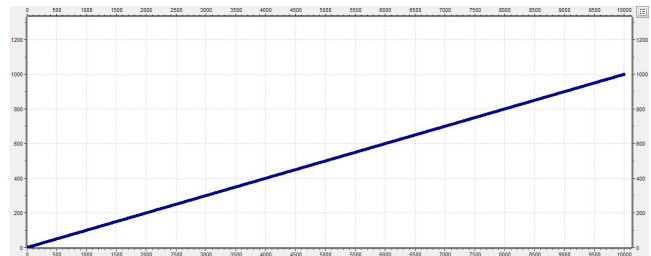


Poizkus s tremi strežniki:

efektivna moč treh strežnikov je eno obdelano sporočilo na 3.3s, kar bi pomenilo da 3je strežniki z lahkoto obdelajo tako 7s intervale kot 5s, z 3s medprihodnim časom pa se ne bodo morali kosati, saj se bo tistih 0.3s kopičilo na čakalno vrsto.

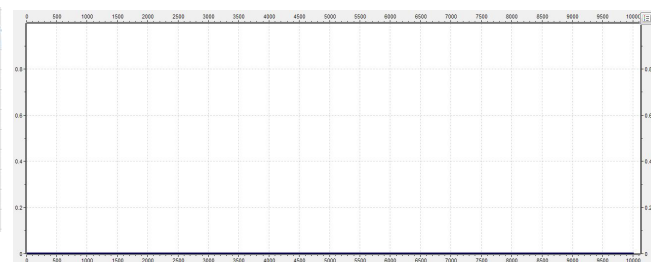
- medprihodni čas 3s: polna zasedenost, vrsta raste v neskončnost

config1 : \$interArrivalTime=3, \$serverCount=3 ; #0	
MMc.queue	
dropped:count (scalar)	1.0
dropped:vector (vector)	0.0 (1)
serverCount:max (scalar)	3.0
serverCount:mean (scalar)	2.9989974937343
serverCount:vector (vector)	2.9989974937343358 (9975)
waitTime:max (scalar)	1000.0
waitTime:mean (scalar)	499.50033311126
waitTime:vector (vector)	499.5003331112592 (3002)



- medprihodni čas 5s: delna zasedenost, opazimo da je serverCount:max zgolj 2, kar pomeni da je en strežnik odveč, saj se ni niti enkrat aktiviral.

config1 : \$interArrivalTime=5, \$serverCount=3 ; #0	
MMc.queue	
dropped:count (scalar)	1.0
dropped:vector (vector)	0.0 (1)
serverCount:max (scalar)	2.0
serverCount:mean (scalar)	1.9993983152828
serverCount:vector (vector)	1.9993983152827919 (9972)
waitTime:max (scalar)	0.0
waitTime:mean (scalar)	0.0
waitTime:vector (vector)	0.0 (2001)



- medprihodni čas 7s: še nižja zasedenost kot pri 5s, tretji strežnik še vedno neaktiviran in potrata resursov

config1 : \$interArrivalTime=7, \$serverCount=3 ; #0	
MMc.queue	
dropped:count (scalar)	1.0
dropped:vector (vector)	0.0 (1)
serverCount:max (scalar)	2.0
serverCount:mean (scalar)	1.4262262262262
serverCount:vector (vector)	1.426226226226262 (9990)
waitTime:max (scalar)	0.0
waitTime:mean (scalar)	0.0
waitTime:vector (vector)	0.0 (1430)

