QueState Models Documentation
jfbrady@admin.nv.gov

Overview

There are a significant number of queueing model calculation programs on the internet which produce performance statistics such as the probability of blocking, but rarely enumerate the model's state probabilities. That is, the probability of 0, 1, 2, etc. customers in either queue or service. These state probabilities are the most fundamental results the model produces, and more importantly, represent the customer concurrency distribution. They answer the question; what are the chances there are "N" customers in the queueing system? The "QueState" programs provide this customer concurrency information in addition to the usual performance statistics for eight queueing models in an open-source Perl script environment where results are produced in a spreadsheet friendly comma delimited format.

Queueing Model Descriptions

All eight queueing models contained in the QueState package assume that customers arrive at the queueing system independently, Markovian arrivals, with service on a first come first serve basis. The models vary in service time distribution, server count, queue length, and customer population.

Kendall notation is used to articulate the properties of each model as follows:

Kendall Notation: A/S/c/k/N/D

Where:

1. A: The arrival process
2. S: The service time distribution
3. c: The number of servers (inf – indicates an infinite number)
4. k: The number of places in the queue
5. N: The size of the customer population
6. D: The queue's discipline which is always first come first serve

The eight models are:

A. **M/M/c** - Markovian arrivals and service with c servers.
- Blocked arrivals are delayed model: Erlang C

B. **M/M/c/c** - Markovian arrivals and distribution free service with c servers and no queueing
- Blocked arrivals are cleared model: Erlang B
- Truncated Poisson distribution accommodating the disposition of the blocked arrival
- Distribution free service implies the second M in the Kendall notation is superfluous

C. **M/M/c/c/N** - Markovian arrivals and distribution free service with c servers, no queueing, and population limit.
- M/M/c/c model with customer population limit: Engset
- Truncated Binomial distribution accommodating the disposition of a blocked arrival
- Distribution free service implies the second M in the Kendall notation is superfluous
- The outside observer's view of the system differs from the arriving customer's view

D. **M/M/c/k** – Markovian arrivals and service with c servers and queue length limit k
- M/M/c model with finite waiting space

E. **M/M/c/N** - Markovian arrivals and service with c servers and population limit
- M/M/c model with customer population limit
- The outside observer's view of the system differs from the arriving customer's view

F. **M/M/inf** – Poisson distribution ordinates
- Blocked arrivals are held model: Poisson

- M/M/c/c without accommodating the disposition of the blocked arrival

G. **M/M/inf/N** – Binomial distribution ordinates
- M/M/c/c/N model without accommodating the disposition of blocked arrival
- The outside observer's view of the system differs from the arriving customer's view

H. **Normal** – Normal distribution ordinates
- The Normal distribution is included for completeness because some models, e.g., M/M/Inf, converge to it under certain conditions.

Directory Structure

The bin directory contains the computer programs, and the demo directory provides an example for each.
1. bin – eight Perl scripts
2. demo – eight directories each with an input file and a program run file.

Program Setup

The environmental variable **QueState** must be created pointing to the **bin** directory. Use "set" for Windows and "export" for Unix / Linux.

Program Execution

The following tables identify each model program's required input parameters, calculated traffic load quantities, and produced performance statistics.

Figure 1 contains:
a. Input parameters
b. Load quantities
c. Performance statistics

Input Parameters

Model	Servers	Sources	p[show]	Qsize	lamda	mu	sdev	t
MMc	Y	-	Y	-	Y	Y	-	Y
MMcc	Y	-	-	-	Y	Y	-	-
MMccN	Y	Y	-	-	Y	Y	-	-
MMck	Y	-	-	Y	Y	Y	-	-
MMcN	Y	Y	-	-	Y	Y	-	Y
MMinf	Y	-	Y	-	Y	Y	-	-
MMinfN	Y	Y	-	-	Y	Y	-	-
Normal	Y	-	Y	-	Y	Y	Y	-

Servers = Number of Servers (c)
Sources = Number of Traffic Sources in the Limited Population (N)
p[show] = Number of State Probabilities to Display (when the possible number is infinite)
Qsize = Maximum Length of the Queue (k)
lamda = Arrival Rate
mu = Service Rate
sdev = Standard Deviation of Offered Load (a)
t = Time Interval To Wait

Figure 1a: Input parameters

As indicated in the "Input Parameters" table, all models require specification of an arrival rate, lamda, a service rate, mu, and the number of servers, Servers.

Load Quantities

Model	a	a_src	ai_src	a_offered	a_carried	a_intend	lamda_e	rho	rho_used
MMc	Y	-	-	-	-	-	-	Y	-
MMcc	Y	-	-	-	Y	-	Y	Y	Y
MMccN	-	-	Y	Y	Y	Y	Y	Y	Y
MMck	Y	-	-	-	Y	-	Y	Y	Y
MMcN	-	-	Y	Y	-	Y	Y	Y	-
MMinf	Y	-	-	-	-	-	-	Y	-
MMinfN	Y	Y	-	-	-	-	-	Y	-
Normal	Y	-	-	-	-	-	-	Y	-

a = Offered Load (lamda / mu)
a_src = Offered Load per Source
ai_src = Offered Load per Idle Source
a_offered = Offered Load (Vs Carried Load)
a_carried = Carried Load
a_intend = Intended Offered Load
lamda_e = Effective Arrival Rate
rho = lamda / (c * mu)
rho_used = lamda_e / (c * mu)

Figure 1b: Load quantities

Calculation of "Load Quantities" is a function of model characteristics. Offered Load per Idle Source, for example, is a key load quantity for the models with limited customer populations, e.g., M/M/c/N.

Performance Statistics

Model	E(nq)	E(ns)	E(qt)	E(st)	E(qt/q)	p[q]	p[q>t]	p[s>t]	p[q>t/q]	p[block]
MMc	Y	Y	Y	Y	Y	Y	Y	Y	Y	-
MMcc	-	-	-	-	-	-	-	-	-	Y
MMccN	-	-	-	-	-	-	-	-	-	Y
MMck	Y	Y	Y	Y	Y	Y	-	-	-	Y
MMcN	Y	Y	Y	Y	Y	Y	Y	Y	Y	-
MMinf	-	-	-	-	-	-	-	-	-	Y
MMinfN	-	-	-	-	-	-	-	-	-	Y
Normal	-	-	-	-	-	-	-	-	-	Y

E(nq) = Average Number in Queue
E(ns) = Average Number in System (Queue + Service)
E(qt) = Average Time in Queue
E(st) = Average Time in System (Queue + Service)
E(qt/q) = Average Time in Queue if Queue
p[q] = Probability of Queueing
p[q>t] = Probability of Queueing > t Time
p[s>t] = Probability of Being in System > t Time
p[q>t/q] = Probability of Queueing > t Time if Queue
p[block] = Probability of Blocking (clear or hold)

Figure 1c: Performance statistics

The models where customers queue when all servers are busy produce the most performance statistics, e.g., M/M/c.

Example Run

All QueState queueing model programs are written in Perl and each can be executed with a single command line for multiple input files containing more than one input entry. The input parameters are in ".txt" files and the load, performance, and state probabilities are written to ".csv" files under a "*Model*_QueState_Results" directory. Each model execution produces a new set of ".csv" files uniquely specified with a YYYYMMDDhhmmss file name component.

The following is a summary of QueState model input, output, and execution components for the M/M/c example contained in the demo directory.

A. **Program Run File**:
   - Run_MMc_QueState.bat
      i. %QueState%\ MMc_QueState.pl
      ii. %QueState% is an environmental variable pointing to the bin directory

B. **Input File**:
   - MMc_2000_AggRpt.txt
      i. Servers, p[show],Lamda,Mu,t - MMc_QueState
      ii. 8,20,77.55,13.34,.1

C. **Output Directory**:
   - MMc_QueState_Results

D. **Output File**:
   - MMc_2000_AggRpt_results_ YYYYMMDDhhmmss.csv
      i. Input Data
      ii. Performance Statistics
      iii. State Probabilities

A portion of the MMc_2000_AggRpt_results_ YYYYMMDDhhmmss.csv output file is shown in Figure 2.

MMc QueState Results: Monday 12/21/2020 09:13:25																					
c	p[show]	lamda	mu	t	a	rho	E(nq)	E(ns)	E(qt)	E(st)	E(qt/q)	p[q]	p[q>t]	p[s>t]	p[q>t/q]	p_sum					
8	20	77.55	13.34	0.1	5.81	0.7267	0.8372	6.6505	0.0108	0.0858	0.0343	0.3149	0.0170	0.3190	0.0541	0.9950					
StateProb	p[0]	p[1]	p[2]	p[3]	p[4]	p[5]	p[6]	p[7]	p[8]	p[9]	p[10]	p[11]	p[12]	p[13]	p[14]	p[15]	p[16]	p[17]	p[18]	p[19]	p[20]
QSystem	0.0027	0.0155	0.0450	0.0871	0.1266	0.1472	0.1426	0.1184	0.0861	0.0625	0.0455	0.0330	0.0240	0.0174	0.0127	0.0092	0.0067	0.0049	0.0035	0.0026	0.0019

Figure 2: MMC model output file

The second and third row of this file contain input data and performance statistics while the fourth and fifth row provide state probabilities 0 - 20.

Results Comparison

A summary of state probabilities for all models using the same input for each of them is shown in Figure 3 and included in this repository as file QueState_doc.xlsx. This figure graphically depicts the state probabilities for each of the eight models using the input in Table B below the graph. Table B numbers are based on Table A, the web-generator-toolkit2 repository's demo concurrency information - 2000_Agg.

Table C lists the web-generator-toolkit2 repository's demo concurrency proportion of time decimal fractions (thick black line) along with the first twenty-one state probabilities contained in the QueState demo output files. For models like MMcN that possess two sets of state probabilities, arrival and outside observer, the outside observer results are displayed in Table C.

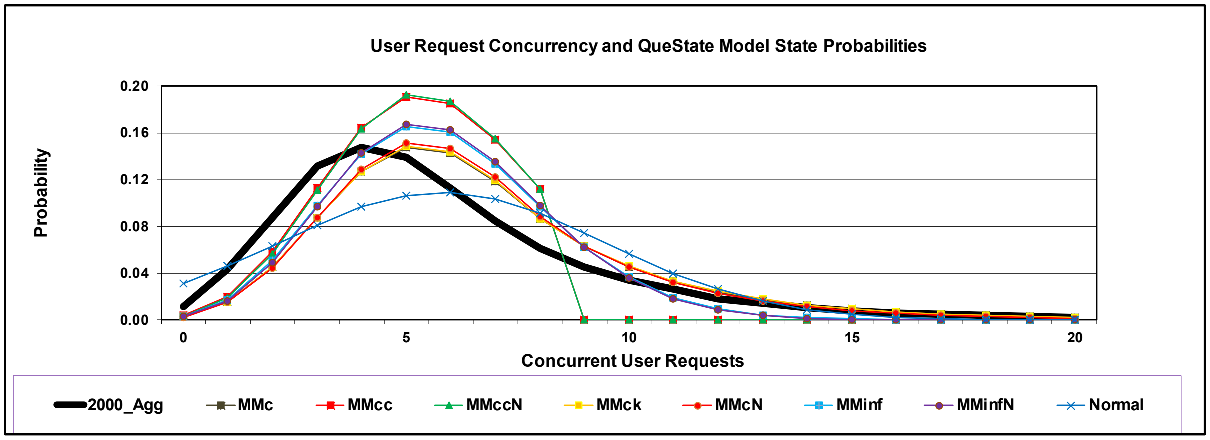

Table A	User Request Concurrency					
	Concurrent Requests					
Test Run	mean	sdev	var	max	p_sum	f_sum
2000_Agg	5.81	3.65	13.35	33	1.00	171699

Table B	QueState Model Input						lamda		
Model	Servers	Source	Qsize	p[show]	lamda	mu	/mu	var	sdev
MMc	8	-	-	20	77.55	13.34	5.81	-	-
MMcc	8	-	-	-	77.55	13.34	5.81	-	-
MMccN	8	200	-	-	77.55	13.34	5.81	-	-
MMck	8	-	20	-	77.55	13.34	5.81	-	-
MMcN	8	200	-	-	77.55	13.34	5.81	-	-
MMinf	8	-	-	20	77.55	13.34	5.81	-	-
MMinfN	8	200	-	-	77.55	13.34	5.81	-	-
Normal	8	-	-	20	77.55	13.34	5.81	13.35	3.65

Table C	State Probabilities																					
StateProb	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Total
2000_Agg	0.0118	0.0431	0.0871	0.1318	0.1478	0.1387	0.1130	0.0851	0.0614	0.0448	0.0339	0.0263	0.0182	0.0144	0.0109	0.0078	0.0062	0.0051	0.0037	0.0025	0.0018	0.9953
MMc	0.0027	0.0155	0.0450	0.0871	0.1266	0.1472	0.1426	0.1184	0.0861	0.0625	0.0455	0.0330	0.0240	0.0174	0.0127	0.0092	0.0067	0.0049	0.0035	0.0026	0.0019	0.9950
MMcc	0.0035	0.0201	0.0583	0.1130	0.1642	0.1909	0.1849	0.1536	0.1116	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
MMccN	0.0032	0.0189	0.0562	0.1111	0.1639	0.1923	0.1871	0.1552	0.1121	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
MMck	0.0027	0.0155	0.0452	0.0876	0.1272	0.1479	0.1433	0.1190	0.0865	0.0629	0.0457	0.0332	0.0241	0.0175	0.0127	0.0093	0.0067	0.0049	0.0036	0.0026	0.0019	1.0000
MMcN	0.0025	0.0148	0.0441	0.0872	0.1286	0.1509	0.1468	0.1218	0.0880	0.0632	0.0452	0.0321	0.0227	0.0160	0.0112	0.0078	0.0054	0.0037	0.0025	0.0017	0.0012	0.9977
MMinf	0.0030	0.0174	0.0505	0.0978	0.1422	0.1653	0.1601	0.1330	0.0966	0.0624	0.0363	0.0192	0.0093	0.0042	0.0017	0.0007	0.0002	0.0001	0.0000	0.0000	0.0000	1.0000
MMinfN	0.0027	0.0164	0.0489	0.0966	0.1424	0.1671	0.1626	0.1349	0.0974	0.0622	0.0356	0.0184	0.0087	0.0038	0.0015	0.0006	0.0002	0.0001	0.0000	0.0000	0.0000	1.0000
Normal	0.0307	0.0458	0.0633	0.0812	0.0966	0.1066	0.1092	0.1037	0.0913	0.0747	0.0566	0.0398	0.0260	0.0157	0.0088	0.0046	0.0022	0.0010	0.0004	0.0002	0.0001	0.9586

Figure 3: User request concurrency (2000_Agg) and QueState model state probabilities

The Table B listing of eight servers for all models and two hundred sources for the finite source models matches the actual load test, Table A, which was run with eight CPU cores and two hundred virtual user threads. The model curves that closely aligned with the web-generator-toolkit2 results may be useful in estimating concurrency for environments with similar traffic flow characteristics when load testing data is unavailable.

Summary

There are computer programs on the internet that provide performance statistics for these eight queueing models but few of them include state probabilities. The QueState programs produce this information, providing a picture of customer concurrency within the queueing system. For details regarding model formulas and assumptions, please consult the references. Reference [2.] was used the most as a conceptual guide in QueState program development and can be downloaded from the internet at no cost.

References

[1.] A.O. Allen, “Probability, Statistics, and Queueing Theory”, Academic Press, Inc., Orlando, FL, 1978.

[2.] R. B. Cooper, “Introduction to Queueing Theory”, Elsevier Science Publishing Co., Inc, New York, N.Y., (1984).

[3.] W.C. Giffin, “Queueing: Basic Theory and Applications”, Grid, Inc, Columbus, Ohio, 1978.

[4.] L. Kleinrock, “Queueing Systems Volume 1 and 2”, John Wiley & Sons, New York, N.Y., (1975).