Automated multi-paradigm analysis of extended and layered queueing models with LINE

Giuliano Casale
Department of Computing
Imperial College London
g.casale@imperial.ac.uk
LINE Solver (line-solver.sf.net)

- MATLAB library for system performance and reliability analysis based on queueing theory
- Ver 2.0.0: Major tool overhaul and refactoring

Line Solver (line-solver.sf.net)

Performance and Reliability Analysis Engine

What is LINE?
LINE is an open source MATLAB library for system performance and reliability analysis based on queueing theory.

Main features
The tool offers a language to specify extended queueing networks and layered queueing networks together with analytical and simulation-based techniques for their solution.

Models are solved in LINE with either native algorithms (CTMC, fluid, simulation, MVA, ...) or via external solvers, such as JMT, LQNS, and BuTools. The tool output metrics include throughputs, utilizations, response times, queue-lengths, and state probabilities. Metrics can be averages or distribution/percentiles, either in steady-state or transient regime.

Download
Download the latest release for MATLAB (version 2018a or later) or clone the source code repository. Installation information is available in the README file.
What’s new

I. Object-oriented language to model extended and layered queueing networks (EQNs / LQNs)

II. Model specification fully decoupled from analysis paradigm

III. Seamless integration with JMT, LQNS, BuTools solvers

Line in numbers:

- 40+ algorithms
- 13 types of analyses
- 13 sched./routing strategies
- 100+ lang. classes
- 14 node types
- 4 metrics
JMT Integration

Seamless two-way integration:

• Define a model in MATLAB, visualize it in JMT
• Define a model in JMT, auto-generate LINE script
EQN Analysis

Continuous-Time Markov Chains
SolverCTMC(model).getAvg

Fluid ODEs
SolverFluid(model).getAvg

Java Modelling Tools
SolverJMT(model).getAvg

Mean-Value Analysis
SolverMVA(model).getAvg

\[ \frac{dx(t)}{dt} = F(x(t)), \quad t \geq 0, \]

\[ Q_i(N - 1) = \frac{N - 1}{N} Q_i(N) \]
EQN Analysis

Matrix-Analytic Methods
SolverMAM(model).getAvg

Normalizing Constant
SolverNC(model).getAvg

Stochastic Simulation
SolverSSA(model).getAvg

LQNS/LQSIM
SolverLQNS(lqnmodel).getAvg

\[ P = \begin{pmatrix}
B_0 & B_1 & B_2 & B_3 & \cdots \\
A_0 & A_1 & A_2 & A_3 & \cdots \\
 & A_0 & A_1 & \cdots \\
 & & A_0 & A_1 & \cdots \\
& & & & \ddots \\
\end{pmatrix} \]

\[ Q_i(N - 1) = \frac{N - 1}{N} Q_i(N) \]
**Parametric Solvers (LQN, Rand. Env.)**

- **Line Layered Solver**
  
  \[
  \text{SolverLN}(\text{model}, @\text{(layer)} \text{SolverJMT}(\text{layer}))
  \]

- **Random Environments**
  
  \[
  \text{SolverEnv}(\text{models, env, @}\text{(submodel)} \text{SolverFluid}(\text{submodel}))
  \]

---

Diagram showing the interactions between different states and models with associated parameters and CTMC models.
Demos

1. A M/M/1 queue
2. A multiclass M/G/1 queue
3. Machine interference problem
4. Round-robin load balancing
5. Modelling a re-entrant line
6. A queueing network with caching
7. Response time distribution and percentiles
8. Optimizing a performance metric