Production Lines Analysis Tool

Paulo Fernandes - PUCRS University, Brazil
Eddie O'Kelly - Waterford Institute of Technology, Ireland
Chrissoleon Papadopoulos - Aristotle University of Thessaloniki, Greece
Afonso Sales - PUCRS University, Brazil
Outline

• Motivation
• Product lines modeled by PLAT
• The PLAT tool
  – Solution of models
  – Examples of usage
• Conclusion and future work
Motivation

• The exact solution is a rare commodity
• Even approximate solutions are not easily performed by practitioners
• Recent results using SAN – Stochastic Automata Networks (Kronecker-based) deliver exact solution for fairly large models
• PLAT makes exact solution available to anyone:
  – Practitioners looking for predictions
  – Academics looking for comparing paradigms
Production Lines – the input

- $K$ reliable stations with finite intermediate buffers in a single line with blocking behavior
- The first machine has an infinite buffer and it never starves
- Exponentially distributed service rates described as a average
- Limitations according to the number of stations and buffer sizes
Product Lines – the output

Performance Indexes

- Throughput
  - the number of jobs served per time unit in each station
- Buffer occupation
  - average number of jobs in each buffer
- Server utilization
  - probability of a server being busy
- The sojourn time
  - the average time a job spend waiting in the buffer plus the time being served in the station (the inverse of the service rate)
The PLAT tool

• Modules (transparent to the user)
  – Conversion to a SAN (Kronecker-based) model
  – Solution using SAN solvers
    • Stationary probability distribution of all model states
    • Transient probability distribution of all model states, considering one single initial state
  – Computation of performance indexes to each station (buffer and server)

• Available as a webservice
  http://marfim.lad.pucrs.br:16000/plat/
The PLAT tool

• SAN Solvers
    • A large model has more than $10^6$ sized state space
  – GTA Express – SAN lite solver using Power method for Stationary solution of “small” models
    • A “small” model has less than $10^6$ sized state space
  – PEPS 2007 – Split algorithm using Uniformization method for Transient solution of all models
Example of Usage

- A product line with 8 stations:
  - incremental buffer sizes ($B_2 = 1$, $B_3 = 2$, ... $B_8 = 7$)
  - service rates
    - $\mu_1 = 1.00$ job/t.u.
    - $\mu_2 = 0.95$ jobs/t.u.
    - $\mu_3 = 0.98$ jobs/t.u.
    - $\mu_4 = 0.97$ jobs/t.u.
    - $\mu_5 = 0.90$ jobs/t.u.
    - $\mu_6 = 1.05$ jobs/t.u.
    - $\mu_7 = 1.02$ jobs/t.u.
    - $\mu_8 = 0.96$ jobs/t.u.
Example of Usage

• Such model has a 604,800 product state space with a 517,412 reachable state space
Example of Usage

- **Stationary solution**

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Status</th>
<th>K</th>
<th>Solution</th>
<th>Tool (Method)</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011-10-11 19:13:50</td>
<td>Complete</td>
<td>8</td>
<td>Stationary</td>
<td>GTAex (SAN Lite Solver)</td>
<td></td>
</tr>
</tbody>
</table>

  Performed in approx. 5 min.

<table>
<thead>
<tr>
<th>Buffer Size</th>
<th>Service Rate</th>
<th>Throughput</th>
<th>Buffer Occupation</th>
<th>Server Utilization</th>
<th>Sojourn Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\infty$</td>
<td>1.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.95</td>
<td>0.6607</td>
<td>0.6285</td>
<td>0.8430</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.98</td>
<td>0.6607</td>
<td>0.8645</td>
<td>0.7625</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.97</td>
<td>0.6607</td>
<td>1.0696</td>
<td>0.7432</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0.9</td>
<td>0.6607</td>
<td>1.2920</td>
<td>0.7487</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>1.05</td>
<td>0.6607</td>
<td>0.8940</td>
<td>0.6410</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>1.02</td>
<td>0.6607</td>
<td>1.0381</td>
<td>0.6587</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>0.96</td>
<td>0.6607</td>
<td>1.2572</td>
<td>0.6882</td>
</tr>
</tbody>
</table>
Example of Usage

• Transient solution
  – considering 10 time units and starting with an empty production line

Also performed in approx. 5 mins.
Example of Usage

- **Stationary Results**
  - Infinite time
  - The initial state does not matter

- **Transient Results**
  - 10 time units
  - Starting empty
Conclusions

• The tool is freely available at
  
  http://www.inf.pucrs.br/~paulof/plat/
  
• It offers an easy-to-use solution to rather large production lines
  – Up to 18 stations according to buffer sizes
  – Before the development of PLAT, steady state solutions were available for reliable production lines with exponential single server stations with up to 300,000 states
  – With PLAT the solution (transient and steady state) of a model with 129,140,163 states (model with $K = 18$ and $Bi = 0$, all $i$), i.e., a model 430 times larger!
Conclusions

• It has a black box translation to a SAN model delivering comprehensive performance indexes
  – The user does not need to be a performance specialist

• It uses very efficient solvers encapsulated in a webserver
  – There is no need of installation procedures, nor computational resources

• It delivers exact stationary and transient predictions
  – These results are not often available to production lines
Future Work

• Extension to model production lines with:
  – Multiple server stations
  – Different routing behaviors than blocking (e.g., loss, restart, etc.)
  – Multi-line architectures (e.g., fork, join, alternative and load dependant routing, etc.)
  – Different classes of jobs

• Extension to deal with larger models using different techniques
  – Aggregation techniques
  – Perfect sampling
Finally ...

- Do visit PLAT webpage and try solve some production line models!

  http://www.inf.pucrs.br/~paulof/plat/

- If you need to solve models larger than those currently available at PLAT, or just want to know more our models or even about the solvers we use, write us!

  paulo.fernandes@pucrs.br
  mejokelly@gmail.com
  hpap@econ.auth.gr
  afonso.sales@pucrs.br
References


