Introduction to Queueing Petri Nets: Modeling Formalism, Tool Support and Case Studies

Samuel Kounev, Simon Spinner and Philipp Meier

Tutorial presented at ICPE 2012, Boston, USA, April 22, 2012
References: Foundations

Modeling Formalism and Modeling Methodology


Analytical and simulation-based solution techniques


Model-to-Model Transformations / QPNs as Intermediate Abstraction


Note: Most papers listed here can be downloaded from http://www.descartes-research.net
References: Selected Case Studies

Java EE Applications

Enterprise Grid Environments

Enterprise Data Fabrics

Message-oriented Event-driven Systems
References: Tools & Further Resources


- HiQPN-Tool: http://ls4-www.cs.tu-dortmund.de/QPN/

- Further Resources
  QPNs at TU Dortmund: http://ls4-www.cs.tu-dortmund.de/QPN/
  QPNs at KIT: http://descartes.ipd.kit.edu/projects/qpme/
  QPN Bibliography: http://descartes.ipd.kit.edu/projects/qpme/qpn_bibliography/
Roadmap

- Introduction to Petri Nets
- Extensions to ordinary Petri Nets
- Queueing Petri Nets (QPNs)
- Case Studies
- Tool support
Recall: Queueing Networks

- **QN**: Set of interconnected queues
- **Queue** = waiting area and servers
- Scheduling strategies (FCFS, PS, ...)
- Single-class vs. multi-class
- Open, closed or mixed

**PROS**: Very powerful for modelling **hardware contention** and scheduling strategies. Many efficient analysis techniques available.

**CONS**: Not as good for modelling software contention aspects such as blocking, synchronization and simultaneous resource possession. *Extended QNs* provide some limited support for the above, however, they are often restrictive and inaccurate.
Petri Nets (PNs)

- Introduced in 1962 by Carl Adam Petri
- Ordinary Petri Net (Place-Transition Nets)

Main concepts

- Places
- Tokens
- Transitions
  - Input place, output place
  - Enabled transition
  - Transition firing
  - Firing weights
- Incidence functions
- Initial marking

An ordinary PN before and after firing transition $t_1$. 

© Samuel Kounev, Simon Spinner and Philipp Meier
Extensions to Petri Nets

- Colored PNs
  - Entension introduced by Kurt Jensen
  - Allow a type, i.e., a color, to be attached to a token
  - Allow transitions to fire in different *modes* *(transition colors)*

- Stochastic PNs (SPNs)
  - Enabled transitions fire after a *firing delay* *(exponential)*

- Generalized Stochastic PNs (GSPNs)
  - Allow two types of transitions to be used: *immediate* and *timed*

- Colored Generalized Stochastic PNs (CGSPNs)
  - CPNs + GSPNs
CGSPNs: Pros and Cons

**PROS:** Suitable both for qualitative and quantitative analysis. Lend themselves very well to modelling blocking, synchronization, simultaneous resource possession and software contention.

**CONS:** No direct means for modelling scheduling strategies. Not as many algorithms/tools for efficient quantitative analysis are available as for Queueing Networks.
GSPN Models of Queueing Stations

M/M/1-FCFS queue

- Assumptions
  - Single token color (tokens undistinguishable)
  - Exponentially-distributed service times
GSPN Models of Queueing Stations (2)

Model of a M/M/1/3-FCFS queue with 2 token colors (have to encode the color of the token in each position of the queue):

- $(4 + B + B \cdot \text{NumColors})$ places needed for queue capacity $B$
- For non-exponentially distributed firing delays or multiple servers, further places/transitions would be needed.

* $(4 + B + B \cdot \text{NumColors})$ places needed for queue capacity $B$! 
* For non-exponentially distributed firing delays or multiple servers, further places/transitions would be needed.
PS and IS Queues with Multiple Colors

PS queues:
- Marking-dependent firing rates would be needed
- Problem: how to manage residual service times!
  - For queues with exponential service time distributions
  - For queues with general service time distributions

IS queues:
- Support for “IS server policy” would be needed
Queueing Petri Nets (QPNs = QNs + PNs)

- Introduced by **Falko Bause** in 1993.
- Combine queueing networks and Petri nets
- Allow integration of queues into places of PNs
- Ordinary vs. queueing places
- **Queueing place** = queue + depository

**PROS:** Combine the modelling power and expressiveness of QNs and PNs. Facilitate the modelling of both hardware and software aspects of system behavior in the same model.

**CONS:** Analysis suffers the **state space explosion** problem and this imposes a limit on the size of the models that are analyzable.
Queueing Petri Nets

Transition

Token

Queue

Depository

Nested QPN

Ordinary Place

Queueing Place

Subnet Place
Hierarchical Queueing Petri Nets (HQPNs)

- Allow hierarchical model specification
- **Subnet place**: contains a nested QPN
- Structured analysis methods alleviate the state space explosion problem

![Graphical Notation for Subnet Place](image-url)
Example QPN
Solution Techniques

- **Analytical solution techniques**
  - Implemented as part of the HiQPN-Tool from TU Dortmund: http://ls4-www.cs.tu-dortmund.de/QPN/

- **Simulation-based techniques**
  - Implemented as part of the Queueing Petri net Modeling Environment (QPME): http://qpme.sourceforge.net
Modeling Case Studies

- Java EE applications
- Enterprise data fabrics
- Enterprise Grid environments
- Message-oriented systems
- Distributed event-based systems
- Component-based software architectures
Java EE Applications

- Modeling methodology for distributed component systems
- Modeled SPECjAppServer2004 / SPECjEnterprise2010
- Extensive validation in a realistic environment
- Model accuracy for
  - throughput (+/- 5%)
  - utilization (+/- 10%)
  - response times (+/- 30%)

Further details in:


SPECjEnterprise2010 Business Model

Dealers

Dealer Domain

Customer Domain

Corporate Domain

Suppliers

Supplier Domain

Manufacturing Domain

© Samuel Kounov, Simon Spinner and Philipp Meier
Case Study - Deployment Environment

Suppliers

Dealers

Internet

HTTP Load Balancer
1 x AMD XP2000+ CPU, 1GB

WebLogic 8.1 Cluster
Each node with 1 x AMD XP2000+ CPU, 1GB

Oracle 9i Server
2xAMD MP2000+
2GB RAM

1 GBit LAN

© Samuel Kounev, Simon Spinner and Philipp Meier
QPN Model

P/p = purchase; M/m = manage; B/b = browse
W/w = workorder; l = largeorder
D = P, M or B
d = p, m or b
o = d, l or w
QPN Model (2)

\[ P/p = \text{purchase}; \quad M/m = \text{manage}; \quad B/b = \text{browse} \]
\[ W/w = \text{workorder}; \quad l = \text{largeorder} \]
\[ D = P, M \text{ or } B \]
\[ d = p, m \text{ or } b \]
\[ o = d, l \text{ or } w \]
Summary Considered Scenarios

- 8AS / HEAVY 4
- 8AS / HEAVY 3
- 8AS / HEAVY 2
- 8AS / HEAVY 1
- 6AS / PEAK / UPG. LB
- 6AS / PEAK / ORIG. LB
- 6AS / NORMAL
- 4AS / NORMAL
Enterprise Grid Environments

- QPN models used for online QoS Control
- Autonomic workload characterization on-the-fly
- Dynamic deployment of Grid servers on demand
- Dynamic system reconfiguration after a server failure

Further details in:

Enterprise Grid Environments
Enterprise Grid Environments

QoS-Aware Resource Manager

Client $t_1$ Service Queue Server $N$ Thread Pool

Server 1 Thread Pool

Grid Server 1

$N$
Enterprise Grid Environments

8-way Pentium Xeon
2.60 GHz, 9 GB, 64 bit,
Xen hypervisor

4-way Pentium Xeon
3.16 GHz, 10 GB, 64 bit,
Xen hypervisor

Globus Toolkit 4.0.5
Enterprise Grid Environments

![Diagram showing response time SLA, response time with overload control, and response time with QoS control.](image)
Enterprise Data Fabrics

- Lightweight modeling approach
- Automatic performance model extraction
- Automated capacity planning tool
- Promising results

Further details in:

Enterprise Data Fabrics

Specify server configuration and target workload → Measure resource demands → Generate model skeleton

Analyze model using SimQPN → Generate model instance → Parameterize model

QoS requirements met?

Process results → Determine bottleneck resource → Increase capacity

No

Yes
Enterprise Data Fabrics
Message-oriented Event-driven Systems

- Modeling methodology for event-based systems
- Modeling patterns
- Modeling case study based on SPECjms2007

Further details in:

Example Event-based Interaction

Order (P/T)
Order sent from SM to DC.

OrderConf (P/T)
Order confirmation sent from DC to SM.

Notify: Order received

Notify: OrderConf

ShipConf (P/T)
Ship Confimation sent from SM to DC

Notify: Shipment arrived

ShipInfo (P/T)
Shipment from DC registered by RFID readers upon arrival at SM.

Notify: Shipment arrived

Notify: ShipConf

ShipDep (P/T)
Ship registered upon leaving warehouse

Notify: Stat. Data

StatInfo (P/T)
Sales statistics sent from DC to HQ

Notify: StatInfo

StatInfoOrderDC (NP/NT)
Sales statistics sent from DC to HQ
QPN Model of Example Interaction
Example Pattern

In T2, for every incoming message n message notifications are created.
Example Pattern (2)

The Modes of Transition 2

T2 has four modes (T2-I to T2-IV), which
a) implement states in the Controller place and
b) create a configured number of notification tokens for each incoming message.

T2-I: Switch to State B

T2-II: Switch to State A

T2-III: State A

T2-IV: State B
QPNs as Intermediate Abstraction

- Software performance models at the architecture level
- Palladio Component Model (PCM)
- PCM models mapped to QPN models
- Tracking of token identity across multiple places
- Efficient simulation using SimQPN

Further details in:

Model-to-Model Transformations

- SimuCom Simulator (standard solver)
  - Highest accuracy
  - Most detailed results
  - High overhead
  - Scalability issues

- Stochastic Regular Expressions
  - Fast analytical solutions
  - Detailed results
  - Single user only

- Layered Queueing Networks
  - Fast analytical solutions
  - Lower accuracy

- Queueing Petri Nets
  - Efficient simulation
  - Lower accuracy

- SimQPN Simulator

- QPME
  - New

- QPE Editor

© Samuel Kounev, Simon Spinner and Philipp Meier
Example Mapping: Loop

Loop Body

incidence functions
QPNs as Intermediate Abstraction
Accuracy vs. Overhead Trade-Off

- In-depth trade-off analysis of different solution techniques
- QPNs used with SimQPN as solution technique
- Optimal balance between accuracy and overhead

Further details in:

Benefits of Queueing Petri Nets

- Combine benefits of QNs and SPNs
- Integration of hardware and software aspects of behavior
- Flexible mapping of logical to physical queues/resources
- Efficient simulation techniques optimized for QPNs
- Good trade-off btw. expressiveness and analysis overhead
- Intuitive graphical representation
- Mature tool support
Tool Support

- Queueing Petri net Modeling Environment (QPME)
- Distributed under Eclipse Public License (EPL) v1.0
- QPN Editor (QPE) and Simulator (SimQPN)
- Runs on all platforms supported by Eclipse
- Website at http://qpme.sourceforge.net/

Further details in:

Tool Support (2)

- HiQPN-Tool
- Supports analytical solution techniques
- Runs on Sun-OS 5.5.x / Solaris 2
- Website at: http://ls4-www.cs.tu-dortmund.de/QPN/

Further details in:

Further Reading


Further Reading


- Papers available for download at http://descartes.ipd.kit.edu/publications/

- See also http://descartes.ipd.kit.edu/projects/qpme/qpn_bibliography/
Thank You!

http://www.descartes-research.net
http://qpme.sourceforge.net