The Convolution Approach to Queuing Networks

Daniel A. Menascé
Department of Computer Science
www.cs.gmu.edu/faculty/menasce.html

Convolution Algorithm

• Basic: recurrence relation to compute the normalization constant of a product-form QN.
• Performance metrics can be obtained from the normalization constant.
Product Form Solution- Single Class – Load Independent Devices

State probability:

\[ P_{n_1,\ldots,n_K} = \frac{1}{G(N)} \prod_{k=1}^{K} D_k^{n_k} \]

Where \( G(N) \) is a normalization constant such that

\[
\sum_{\bar{x} \in S(N,K)} \prod_{k=1}^{K} D_k^{n_k} = 1 \quad \text{and} \quad S(N,K) = \left\{ (n_1,\ldots,n_K) \mid \sum_{k=1}^{K} n_k = N \right\}
\]

Buzen’s Convolution Expression

\[ g_k(n) = g_{k-1}(n) + D_k g_k(n-1) \]

where

\[ g_k(n) = \sum_{\bar{x} \in S(n,k)} \prod_{i=1}^{k} D_i^{n_i} \]

Note that the normalization constant is

\[ G(N) = G_K(N) \]
Buzen’s Convolution Expression

Example

- Let $n=3$ and $k=2$.
- Then $S(3,2) = \{(0,3),(1,2),(2,1),(3,0)\}$

\[
\begin{align*}
g_2(3) &= D_0^2D_2^3 + D_1^1D_2^2 + D_1^2D_2^1 + D_1^3D_2^0 \\
g_1(3) &= D_1^3 = D_1^3 \times 1 = D_1^3D_2^0 \\
g_2(2) &= D_0^2D_2^2 + D_1^1D_2^1 + D_1^2D_2^0 \\
g_2(3) &= D_0^2D_2^0 + D_2^0(D_0^2D_2^2 + D_1^1D_2^1 + D_1^2D_2^0) = \\
&= D_1^3D_2^0 + D_1^0D_2^3 + D_1^1D_2^2 + D_1^2D_2^1
\end{align*}
\]

Convolution Algorithm

\[
\begin{align*}
g_1(0) &= 1 \\
g_1(n) &= g_0(n) + D_1g_1(n-1) = D_1g_1(n-1) \\
g_k(0) &= 1 \quad \forall \quad k
\end{align*}
\]

\[
\begin{array}{c}
g_{k-1}(n) \quad + \rightarrow g_k(n) \\
\end{array}
\]

\[
\begin{array}{c}
g_k(n-1) \quad \times D_k \\
\end{array}
\]

\[
\begin{array}{c}
\downarrow \\
+ \\
\end{array}
\]
Matrix $g$

<table>
<thead>
<tr>
<th>Demands</th>
<th>2</th>
<th>1.8</th>
<th>1.5</th>
<th>Devices</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td></td>
<td>0.000000</td>
</tr>
<tr>
<td>1</td>
<td>2.000</td>
<td>3.800</td>
<td>5.300</td>
<td></td>
<td>0.188679</td>
</tr>
<tr>
<td>2</td>
<td>4.000</td>
<td>10.840</td>
<td>18.790</td>
<td></td>
<td>0.282065</td>
</tr>
<tr>
<td>3</td>
<td>8.000</td>
<td>27.512</td>
<td>55.697</td>
<td></td>
<td>0.337361</td>
</tr>
<tr>
<td>4</td>
<td>16.000</td>
<td>65.522</td>
<td>149.067</td>
<td></td>
<td>0.373637</td>
</tr>
<tr>
<td>5</td>
<td>32.000</td>
<td>149.939</td>
<td>373.540</td>
<td></td>
<td>0.399066</td>
</tr>
</tbody>
</table>

Performance Metrics

- Throughput:
  \[ X_0(N) = \frac{G(N-1)}{G(N)} \]

- Utilization
  \[ U_k(N) = D_k X_0(N) = D_k \frac{G(N-1)}{G(N)} \]

© 1999–2001 D. A. Menascé. All Rights Reserved.
Performance Metrics

• Mean Queue Length (for LI devices)

\[ \bar{n}_i(N) = \sum_{n=1}^{N} D_i^n \frac{G(N-n)}{G(N)} \]

• Recursive Equation for Queue Length:

\[ \bar{n}_k(N) = U_k(N) \times [1 + \bar{n}_k(N-1)] \]