The Convolution Approach to Queuing Networks

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

1

Convolution Algorithm

- Developed by J. P. Buzen in his doctoral dissertation at Harvard in 1971.
- Basic: recurrence relation to compute the normalization constant of a product-form QN.
- Performance metrics can be obtained from the normalization constant.

2

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

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

$$\sum_{\vec{x} \in S(N,K)} \prod_{k=1}^{K} D_k^{n_k} = 1 \quad \text{and} \quad S(N,K) = \left\{ (n_{1,}\ldots,n_K) \mid \sum_{k=1}^{K} n_k = N \right\}$$

Buzen's Convolution Expression

$$g_k(n) = g_{k-1}(n) + D_k g_k(n-1)$$

where

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

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)}

$$g_2(3) = D_1^0 D_2^3 + D_1^1 D_2^2 + D_1^2 D_2^1 + D_1^3 D_2^0$$
$$g_1(3) = D_1^3 = D_1^3 \times 1 = D_1^3 D_2^0$$
$$g_2(2) = D_1^0 D_2^2 + D_1^1 D_2^1 + D_1^2 D_2^0$$
$$g_2(3) = D_1^3 D_2^0 + D_2 (D_1^0 D_2^2 + D_1^1 D_2^1 + D_1^2 D_2^0) =$$
$$= D_1^3 D_2^0 + D_1^0 D_2^3 + D_1^1 D_2^2 + D_1^2 D_2^1$$

5

Convolution Algorithm

$$g_1(0) = 1$$
$$g_1(n) = g_0(n) + D_1 g_1(n-1) = D_1 g_1(n-1)$$
$$g_k(0) = 1 \qquad \forall \quad k$$

$$g_k(n-1) \times D_k$$

$$\downarrow +$$

$$g_{k-1}(n) \xrightarrow{+} g_k(n)$$

6

Matrix g

Demands	2	1.8	1.5	
	Devices			Throughput
	1	2	3	
0	1.000	1.000	1.000	0.000000
1	2.000	3.800	5.300	0.188679
2	4.000	10.840	18.790	0.282065
3	8.000	27.512	55.697	0.337361
4	16.000	65.522	149.067	0.373637
5	32.000	149.939	373.540	0.399066

7

Performance Metrics

- Throughput:

$$X_0(N) = \frac{G(N-1)}{G(N)}$$

- Utilization

$$U_k(N) = D_k X_0(N) = D_k \frac{G(N-1)}{G(N)}$$

8

Performance Metrics

- Mean Queue Length (for LI devices)

$$\overline{n}_i(N) = \sum_{n=1}^{N} D_i^n \frac{G(N-n)}{G(N)}$$

- Recursive Equation for Queue Length:

$$\overline{n}_k(N) = U_k(N) \times \left[1 + \overline{n}_k(N-1)\right]$$

9