Matrix-Geometric Analysis and Its Applications

John C.S. Lui

Department of Computer Science & Engineering
The Chinese University of Hong Kong
Email: cslui@cse.cuhk.edu.hk

Summer Course at Tsinghua, 2005.
Introduction
   Why we need the Matrix-Geometric Technique?

Matrix-Geometric in Action
   Key Idea

General Matrix-Geometric Solution
   General Concept

Application of Matrix-Geometric
   Performance Analysis of Multiprocessing System

Properties of Solutions
   Properties

Computational Properties of $R$
   Algorithm for solving $R$
Outline

Introduction
  Why we need the Matrix-Geometric Technique?
Matrix-Geometric in Action
  Key Idea
General Matrix-Geometric Solution
  General Concept
Application of Matrix-Geometric
  Performance Analysis of Multiprocessing System
Properties of Solutions
  Properties
Computational Properties of $R$
  Algorithm for solving $R$
Closed-form solution is hard to obtain.

Need to seek efficient, numerical stable solutions.

Can be viewed as a generalization of conventional queueing analysis.

A Special way to solve a Markov Chain.
Motivation

- Closed-form solution is hard to obtain.
- Need to seek efficient, numerical stable solutions.
- Can be viewed as a generalization of conventional queueing analysis.
- A Special way to solve a Markov Chain.
Why we need the Matrix-Geometric Technique?

Motivation

- Closed-form solution is hard to obtain.
- Need to seek efficient, numerical stable solutions.
- Can be viewed as a generalization of conventional queueing analysis.
- A Special way to solve a Markov Chain.
Motivation

- Closed-form solution is hard to obtain.
- Need to seek efficient, numerical stable solutions.
- Can be viewed as a generalization of conventional queueing analysis.
- A Special way to solve a Markov Chain.
Outline

Introduction
  Why we need the Matrix-Geometric Technique?

Matrix-Geometric in Action
  Key Idea

General Matrix-Geometric Solution
  General Concept

Application of Matrix-Geometric
  Performance Analysis of Multiprocessing System

Properties of Solutions
  Properties

Computational Properties of $R$
  Algorithm for solving $R$
Key Ideas

- It is a technique to solve stationary state probability for vector state Markov processes. Two parts:
  1. Boundary set
  2. Repetitive set

- Example: a modified $M/M/1$, $\lambda^*$ if the system is empty, else $\lambda$. Customers require two exponential stages of service, $\mu_1$, and $\mu_2$

  $S : \{(i, s) | i \geq 0 \text{ and it is the no. of customer in the queue,} s \text{ is the current stage of service, } s \in (1, 2)\}$

- $s = 0$ if no customer in the system
- Well, let us proceed to specify the state transition diagram, then the $Q$ matrix.
Key Ideas

- It is a technique to solve stationary state probability for vector state Markov processes. Two parts:
  1. Boundary set
  2. Repetitive set

- Example: a modified $M/M/1$, $\lambda^*$ if the system is empty, else $\lambda$. Customers require two exponential stages of service, $\mu_1$, and $\mu_2$

- $S : \{(i, s)|i \geq 0 \text{ and it is the no. of customer in the queue,} \} | s \text{ is the current stage of service, } s \in (1, 2)\}$

- $s = 0$ if no customer in the system

- Well, let us proceed to specify the state transition diagram, then the $Q$ matrix.
Key Ideas

- It is a technique to solve stationary state probability for vector state Markov processes. Two parts:
  1. Boundary set
  2. Repetitive set
- Example: a modified $M/M/1, \lambda^*$ if the system is empty, else $\lambda$. Customers require two exponential stages of service, $\mu_1$, and $\mu_2$

$S : \{(i, s) | i \geq 0 \text{ and it is the no. of customer in the queue, } s \text{ is the current stage of service, } s \in (1, 2)\}$

- $s = 0$ if no customer in the system
- Well, let us proceed to specify the state transition diagram, then the $Q$ matrix.
Key Ideas

- It is a technique to solve stationary state probability for vector state Markov processes. Two parts:
  1. Boundary set
  2. Repetitive set

- Example: a modified $M/M/1$, $\lambda^*$ if the system is empty, else $\lambda$. Customers require two exponential stages of service, $\mu_1$, and $\mu_2$

  $$S : \{(i, s) | i \geq 0 \text{ and it is the no. of customer in the queue,} \}
  s \text{ is the current stage of service, } s \in (1, 2)\}$$

- $s = 0$ if no customer in the system

- Well, let us proceed to specify the state transition diagram, then the $Q$ matrix.
Key Ideas

- It is a technique to solve stationary state probability for vector state Markov processes. Two parts:
  1. Boundary set
  2. Repetitive set

- Example: a modified $M/M/1$, $\lambda^*$ if the system is empty, else $\lambda$. Customers require two exponential stages of service, $\mu_1$, and $\mu_2$

  $S : \{(i, s)| i \geq 0 \text{ and it is the no. of customer in the queue}, s \in (1, 2)\}$

  $s = 0$ if no customer in the system

- Well, let us proceed to specify the state transition diagram, then the $Q$ matrix.
Key Ideas

- It is a technique to solve stationary state probability for vector state Markov processes. Two parts:
  1. Boundary set
  2. Repetitive set

- Example: a modified $M/M/1, \lambda^*$ if the system is empty, else $\lambda$. Customers require two exponential stages of service, $\mu_1$, and $\mu_2$

  $S : \{(i, s) | i \geq 0$ and it is the no. of customer in the queue, $s$ is the current stage of service, $s \in (1, 2)\}$

- $s = 0$ if no customer in the system

- Well, let us proceed to specify the state transition diagram, then the $Q$ matrix.
Let $a_i = \lambda + \mu_i$, $i = 1, 2$. Arrive states lexicographically, $(0, 0), (0, 1), (0, 2), (1, 1), (1, 2), \ldots$
The transition rate matrix \( Q \) is:

\[
Q = \begin{bmatrix}
-\lambda^* & \lambda^* & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & -a_1 & \mu_1 & \lambda & 0 & 0 & 0 & 0 & \cdots \\
\mu_2 & 0 & -a_2 & 0 & \lambda & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & -a_1 & \mu_1 & \lambda & 0 & 0 & \cdots \\
0 & \mu_2 & 0 & 0 & -a_2 & 0 & \lambda & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & -a_1 & \mu_1 & \lambda & 0 & \cdots \\
0 & 0 & 0 & \mu_2 & 0 & 0 & -a_2 & 0 & \lambda & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots
\end{bmatrix}
\]
Let us re-write the $Q$ in matrix form:

$$A_0 = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}; A_1 = \begin{bmatrix} -a_1 & \mu_1 \\ 0 & -a_2 \end{bmatrix}; A_2 = \begin{bmatrix} 0 & 0 \\ \mu_2 & 0 \end{bmatrix}$$

$$B_{00} = \begin{bmatrix} -\lambda^* & \lambda^* & 0 \\ 0 & -a_1 & \mu_1 \\ \mu_2 & 0 & -a_2 \end{bmatrix}; B_{01} = \begin{bmatrix} 0 & 0 \\ \lambda & 0 \\ 0 & \lambda \end{bmatrix}; B_{10} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \mu_2 \\ 0 & \mu_2 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} B_{00} & B_{01} & 0 & 0 & 0 & \cdots \\ B_{10} & A_1 & A_0 & 0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & 0 & A_2 & A_1 & A_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$
Let us solve it. For the repetitive portion

\[ \pi_{j-1}A_0 + \pi_jA_1 + \pi_{j+1}A_2 = 0 \quad j = 2, 3, \ldots \]  

(1)

This is similar to the solution of \( M/M/1 \). Therefore, \( \pi_j \) is a function only of the transition rates between states with \( j - 1 \) queued customers and states with \( j \) queued customers.

\[ \pi_j = \pi_{j-1}R \quad j = 2, 3, \ldots \]

or

\[ \pi_j = \pi_1R^{j-1} \quad j = 2, 3, \ldots \]  

(2)
Putting (2) into (1), we have:

\[ \pi_1 R^{j-2} A_0 + \pi_1 R^{j-1} A_1 + \pi_1 R^j A_2 = 0 \quad j = 2, 3, \ldots \]

Since it is true for \( j = 2, 3, \ldots \), substitute \( j = 2 \), we have:

\[ A_0 + RA_1 + R^2 A_2 = 0 \]
For the initial portion:

\[
\begin{align*}
\pi_0 B_{00} + \pi_1 B_{10} &= 0 \\
\pi_0 B_{01} + \pi_1 A_1 + \pi_2 A_2 &= 0
\end{align*}
\]

or

\[
[\pi_0, \pi_1] \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & A_1 + RA_2 \end{bmatrix} = 0
\]

We also need:

\[
1 = \pi_0 e + \pi_1 \sum_{j=1}^{\infty} R^{j-1} e = \pi_0 e + \pi_1 (I - R)^{-1} e
\]

\[
[\pi_0, \pi_1] \begin{bmatrix} e & B_{00}^* \\ (I - R)^{-1} e & B_{10}^* \\ A_1 + RA_2 \end{bmatrix} = [1, 0]
\]

where \( M^* \) is \( M \) with first column being eliminated.
\[
\tilde{N}_q \quad = \quad E[\text{queued customers}]
\]
\[
= \sum_{j=1}^{\infty} j \pi_j e = \sum_{j=1}^{\infty} (j) \pi_1 R^{j-1} e = \pi_1 (I - R)^{-2} e
\]

Note:
\[
S = \sum_{j=1}^{\infty} R^{j-1} = I + R + R^2 + \cdots
\]
\[
SR = R + R^2 + R^3 + \cdots
\]
\[
S(I - R) = I
\]
\[
S = I (I - R)^{-1} = (I - R)^{-1}
\]

This is true only when the spectral radius of \( R \) is less than unity.
Outline

Introduction
  Why we need the Matrix-Geometric Technique?
Matrix-Geometric in Action
  Key Idea
General Matrix-Geometric Solution
  General Concept
Application of Matrix-Geometric Properties of Solutions
Performance Analysis of Multiprocessing System
Properties of Solutions
  Properties
Computational Properties of $R$
Algorithm for solving $R$
We index the state by \((i, j)\) where \(i\) is the level, \(i \geq 0\) and \(j\) is the state within the level, \(0 \leq j \leq m - 1\) for \(i \geq 1\).
For the repetitive portion,

\[
\sum_{k=0}^{\infty} \pi_{j-1+k} A_k = 0 \quad j = 2, 3, \ldots \tag{3}
\]

\[
\pi_j = \pi_1 R^{j-1} \quad j = 2, 3, \ldots \tag{4}
\]

Putting (4) to (3), we have:

\[
\sum_{k=0}^{\infty} R^k A_k = 0
\]

For the boundary states, we have:

\[
[\pi_0, \pi_1] \begin{bmatrix}
\sum_{k=1}^{\infty} B_{00} R^{k-1} B_{k0} & \sum_{k=1}^{\infty} B_{01} R^{k-1} B_{k1}
\end{bmatrix}
\]
Using the same normalization, we have

\[
\begin{bmatrix} \pi_0, \pi_1 \end{bmatrix} \begin{bmatrix} \mathbf{e} & \mathbf{B}_{00}^* \sum_{k=1}^{\infty} \mathbf{R}^{k-1} \mathbf{B}_0 & \mathbf{B}_{01} \sum_{k=1}^{\infty} \mathbf{R}^{k-1} \mathbf{B}_1 \end{bmatrix} = [1, 0]
\]

Therefore, it boils down to

1. Solving \( \mathbf{R} \).

2. Solving the initial portion of the Markov process.
Using the same normalization, we have

\[
[\pi_0, \pi_1] \begin{bmatrix} e & B^*_0 \sum_{k=1}^{\infty} R^{k-1} B_0 \end{bmatrix}^* \begin{bmatrix} B_{01} \sum_{k=1}^{\infty} R^{k-1} B_1 \end{bmatrix} = [1, 0]
\]

Therefore, it boils down to

1. Solving \( R \).
2. Solving the initial portion of the Markov process.
Outline

Introduction
  Why we need the Matrix-Geometric Technique?
Matrix-Geometric in Action
  Key Idea
General Matrix-Geometric Solution
  General Concept
Application of Matrix-Geometric
  Performance Analysis of Multiprocessing System
Properties of Solutions
  Properties
Computational Properties of $R$
  Algorithm for solving $R$
We have a multiprocessing system in which

- \( K \) homogeneous processors.
- Each processor is subjected to failure with rate \( \gamma \).
- A single repair facility with repair rate \( \alpha \).
- Jobs arrive at a Poisson rate \( \lambda \).
- Whenever there is no processor available, all jobs are lost.
We have a multiprocessing system in which

- $K$ homogeneous processors.
- Each processor is subjected to failure with rate $\gamma$.
- A single repair facility with repair rate $\alpha$.
- Jobs arrive at a Poisson rate $\lambda$.
- Whenever there is no processor available, all jobs are lost.
We have a multiprocessing system in which

- $K$ homogeneous processors.
- Each processor is subjected to failure with rate $\gamma$.
- A single repair facility with repair rate $\alpha$.
- Jobs arrive at a Poisson rate $\lambda$.
- Whenever there is no processor available, all jobs are lost.
We have a multiprocessing system in which

- $K$ homogeneous processors.
- Each processor is subjected to failure with rate $\gamma$.
- A single repair facility with repair rate $\alpha$.
- Jobs arrive at a Poisson rate $\lambda$.
- Whenever there is no processor available, all jobs are lost.
We have a multiprocessing system in which

- $K$ homogeneous processors.
- Each processor is subjected to failure with rate $\gamma$.
- A single repair facility with repair rate $\alpha$.
- Jobs arrive at a Poisson rate $\lambda$.
- Whenever there is no processor available, all jobs are lost.
Markov Model

- **0,0** connected to **1,0** with probability $\gamma$
- **0,0** connected to **0,1** with probability $\alpha$
- **0,0** connected to **0,2** with probability $\alpha$
- **0,0** connected to **0,3** with probability $\alpha$
- **0,0** connected to **0,K** with probability $K\gamma$
- **0,1** connected to **1,1** with probability $\gamma$
- **0,1** connected to **0,0** with probability $\alpha$
- **0,2** connected to **1,2** with probability $\gamma$
- **0,2** connected to **0,1** with probability $\alpha$
- **0,2** connected to **0,0** with probability $\alpha$
- **0,3** connected to **1,3** with probability $\gamma$
- **0,3** connected to **0,2** with probability $\alpha$
- **0,3** connected to **0,0** with probability $\alpha$
- **1,0** connected to **2,0** with probability $\lambda$
- **1,0** connected to **1,1** with probability $\gamma$
- **1,0** connected to **0,0** with probability $\alpha$
- **1,1** connected to **2,1** with probability $\gamma$
- **1,1** connected to **1,0** with probability $\alpha$
- **1,1** connected to **0,0** with probability $\alpha$
- **1,2** connected to **2,2** with probability $\gamma$
- **1,2** connected to **1,1** with probability $\alpha$
- **1,2** connected to **0,1** with probability $\alpha$
- **1,2** connected to **0,0** with probability $\alpha$
- **1,3** connected to **2,3** with probability $\gamma$
- **1,3** connected to **1,2** with probability $\alpha$
- **1,3** connected to **0,3** with probability $\alpha$
- **1,3** connected to **0,2** with probability $\alpha$
- **1,3** connected to **0,1** with probability $\alpha$
- **1,3** connected to **0,0** with probability $\alpha$
- **2,0** connected to **3,0** with probability $\lambda$
- **2,0** connected to **2,1** with probability $\gamma$
- **2,0** connected to **1,0** with probability $\alpha$
- **2,0** connected to **0,0** with probability $\alpha$
- **2,1** connected to **3,1** with probability $\gamma$
- **2,1** connected to **2,0** with probability $\alpha$
- **2,1** connected to **1,1** with probability $\alpha$
- **2,1** connected to **0,1** with probability $\alpha$
- **2,1** connected to **0,0** with probability $\alpha$
- **2,2** connected to **3,2** with probability $\gamma$
- **2,2** connected to **2,1** with probability $\alpha$
- **2,2** connected to **1,2** with probability $\alpha$
- **2,2** connected to **0,2** with probability $\alpha$
- **2,2** connected to **0,1** with probability $\alpha$
- **2,2** connected to **0,0** with probability $\alpha$
- **2,3** connected to **3,3** with probability $\gamma$
- **2,3** connected to **2,2** with probability $\alpha$
- **2,3** connected to **1,3** with probability $\alpha$
- **2,3** connected to **0,3** with probability $\alpha$
- **2,3** connected to **0,2** with probability $\alpha$
- **2,3** connected to **0,1** with probability $\alpha$
- **2,3** connected to **0,0** with probability $\alpha$
- **3,0** connected to **4,0** with probability $\lambda$
- **3,0** connected to **2,0** with probability $\gamma$
- **3,0** connected to **1,0** with probability $\alpha$
- **3,0** connected to **0,0** with probability $\alpha$
- **3,1** connected to **4,1** with probability $\gamma$
- **3,1** connected to **2,1** with probability $\alpha$
- **3,1** connected to **1,1** with probability $\alpha$
- **3,1** connected to **0,1** with probability $\alpha$
- **3,1** connected to **0,0** with probability $\alpha$
- **3,2** connected to **4,2** with probability $\gamma$
- **3,2** connected to **2,2** with probability $\alpha$
- **3,2** connected to **1,2** with probability $\alpha$
- **3,2** connected to **0,2** with probability $\alpha$
- **3,2** connected to **0,1** with probability $\alpha$
- **3,2** connected to **0,0** with probability $\alpha$
- **3,3** connected to **4,3** with probability $\gamma$
- **3,3** connected to **2,3** with probability $\alpha$
- **3,3** connected to **1,3** with probability $\alpha$
- **3,3** connected to **0,3** with probability $\alpha$
- **3,3** connected to **0,2** with probability $\alpha$
- **3,3** connected to **0,1** with probability $\alpha$
- **3,3** connected to **0,0** with probability $\alpha$
- **4,0** connected to **5,0** with probability $\lambda$
- **4,0** connected to **3,0** with probability $\gamma$
- **4,0** connected to **2,0** with probability $\gamma$
- **4,0** connected to **1,0** with probability $\alpha$
- **4,0** connected to **0,0** with probability $\alpha$
- **4,1** connected to **5,1** with probability $\gamma$
- **4,1** connected to **3,1** with probability $\alpha$
- **4,1** connected to **2,1** with probability $\alpha$
- **4,1** connected to **1,1** with probability $\alpha$
- **4,1** connected to **0,1** with probability $\alpha$
- **4,1** connected to **0,0** with probability $\alpha$
- **4,2** connected to **5,2** with probability $\gamma$
- **4,2** connected to **3,2** with probability $\alpha$
- **4,2** connected to **2,2** with probability $\alpha$
- **4,2** connected to **1,2** with probability $\alpha$
- **4,2** connected to **0,2** with probability $\alpha$
- **4,2** connected to **0,1** with probability $\alpha$
- **4,2** connected to **0,0** with probability $\alpha$
- **4,3** connected to **5,3** with probability $\gamma$
- **4,3** connected to **3,3** with probability $\alpha$
- **4,3** connected to **2,3** with probability $\alpha$
- **4,3** connected to **1,3** with probability $\alpha$
- **4,3** connected to **0,3** with probability $\alpha$
- **4,3** connected to **0,2** with probability $\alpha$
- **4,3** connected to **0,1** with probability $\alpha$
- **4,3** connected to **0,0** with probability $\alpha$
Define $b_i = \lambda + i\gamma + \alpha$ for $i = 1, 2, \ldots, K$. We have:

$$
\begin{align*}
B_{00} &= \begin{bmatrix} -\alpha \end{bmatrix}; \\
B_{01} &= \begin{bmatrix} \alpha, 0, \cdots, 0 \end{bmatrix}; \\
B_{j0} &= \begin{bmatrix} \gamma \\ 0 \\ \vdots \\ 0 \end{bmatrix}; \\
B_{0,j} &= \mathbf{0} \quad j = 2, 3, \ldots,
\end{align*}
$$

$$
B_{1,1} = \\
\begin{bmatrix}
-\lambda_1 & \lambda & 0 & 0 & \cdots & 0 \\
2\gamma & -\lambda_2 & \lambda & 0 & \cdots & 0 \\
0 & 3\gamma & -\lambda_3 & \lambda & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & (K-1)\gamma - \lambda_{K+1} & \lambda \\
0 & 0 & 0 & 0 & \cdots & K\gamma & -\lambda_K
\end{bmatrix}
$$
The matrices of the repeating portion of the process are:

\[ A_0 = \lambda I; \quad A_1 = B_{1,1}; \quad A_2 = \begin{bmatrix}
\mu & 0 & \cdots & 0 & 0 \\
0 & 2\mu & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & (K-1)\mu & 0 \\
0 & 0 & \cdots & 0 & K\mu
\end{bmatrix} \]

This can be solved numerically rather than using the transform method.
The matrices of the repeating portion of the process are:

\[
A_0 = \lambda I; \quad A_1 = B_{1,1}; \quad A_2 = \begin{bmatrix}
\mu & 0 & \cdots & 0 & 0 \\
0 & 2\mu & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & (K-1)\mu & 0 \\
0 & 0 & \cdots & 0 & K\mu
\end{bmatrix}
\]

This can be solved numerically rather than using the transform method.
Outline

Introduction
  Why we need the Matrix-Geometric Technique?
Matrix-Geometric in Action
  Key Idea
General Matrix-Geometric Solution
  General Concept
Application of Matrix-Geometric
  Performance Analysis of Multiprocessing System
Properties of Solutions
  Properties
Computational Properties of $R$
  Algorithm for solving $R$
Informally, the stability of a process depends on the *drift* of the process for states in the repetitive portion.

For example, $M/M/1$, the expected drift toward higher states is $\lambda 1$. The expected drift toward lower states is $\mu(-1) = -\mu$. The drift of the process is $\lambda - \mu$. Process is stable if the total expected drift is *NEGATIVE*, or $\lambda < \mu$ in our case.

Now suppose the process can go up by 1 and go down by at most $K$ steps. Let the rate for $l$ steps be $r(l)$, $l = -K, -K - 1, \ldots, 0, 1$.

$$
r(1) + \sum_{l=1}^{K} (-l) r(-l) \rightarrow r(1) < \sum_{l=1}^{K} lr(l)
$$
Informally, the stability of a process depends on the *drift* of the process for states in the repetitive portion.

For example, $M/M/1$, the expected drift toward higher states is $\lambda 1$. The expected drift toward lower states is $\mu(-1) = -\mu$. The drift of the process is $\lambda - \mu$. Process is stable if the total expected drift is *NEGATIVE*, or $\lambda < \mu$ in our case.

Now suppose the process can go up by 1 and go down by at most $K$ steps. Let the rate for $l$ steps be $r(l)$, $l = -K, -K-1, \ldots, 0, 1$.

$$r(1) + \sum_{l=1}^{K} (-l) r(-l) \rightarrow r(1) < \sum_{l=1}^{K} l r(l)$$
Informally, the stability of a process depends on the drift of the process for states in the repetitive portion.

For example, $M/M/1$, the expected drift toward higher states is $\lambda 1$. The expected drift toward lower states is $\mu(-1) = -\mu$. The drift of the process is $\lambda - \mu$. Process is stable if the total expected drift is NEGATIVE, or $\lambda < \mu$ in our case.

Now suppose the process can go up by 1 and go down by at most $K$ steps. Let the rate for $l$ steps be $r(l)$, $l = -K, -K - 1, \ldots, 0, 1$.

$$
 r(1) + \sum_{l=1}^{K} (-l) r(-l) \rightarrow r(1) < \sum_{l=1}^{K} lr(l)
$$
Analogous to the scalar case, we can think of the drift of the process in terms of *levels*. Assume that for the repetitive portion, we have \( m \) states, a transition from level \( i, i >> 0 \), to level \( i - k, 1 \leq k \leq K \)

\[
-k \sum_{l=1}^{m} A_{k+1}(j, l)
\]

where \( A_{k+1}(j, l) \) is the transition from state \( j \) in level \( i \) to state \( l \) in level \( i - k \).

Let \( f_j, 0 \leq j \leq m - 1 \) be the probability that the process is in inter-level \( j \) of the repeating portion of the process of level \( i >> 0 \). The average drift from level \( i \) to level \( i - k \) is

\[
-k \sum_{j=0}^{m-1} f_j \sum_{l=1}^{m-1} A_{k+1}(j, l)
\]
Analogous to the scalar case, we can think of the drift of the process in terms of *levels*. Assume that for the repetitive portion, we have $m$ states, a transition from level $i, i \gg 0$, to level $i - k, 1 \leq k \leq K$

$$-k \sum_{l=1}^{m} A_{k+1}(j, l)$$

where $A_{k+1}(j, l)$ is the transition from state $j$ in level $i$ to state $l$ in level $i - k$.

Let $f_j, 0 \leq j \leq m - 1$ be the probability that the process is in inter-level $j$ of the repeating portion of the process of level $i \gg 0$. The average drift from level $i$ to level $i - k$ is

$$-k \sum_{j=1}^{m-1} f_j \sum_{l=1}^{m-1} A_{k+1}(j, l)$$
To get the total drift, we sum the previous equation for all $k$, $0 \leq k \leq K + 1$.

But what is $f_j$? Let us define $A = \sum_{i=0}^{K+1} A_i$, we have $f = (f_0, f_1, \ldots, f_m)$. Therefore:

$$fA = 0 \quad \& \quad fe = 1$$

The stability condition is:

$$fA_0e < \sum_{k=2}^{K+1} (k - 1)fA_k e$$
To get the total drift, we sum the previous equation for all $k$, $0 \leq k \leq K + 1$.

But what is $f_j$? Let us define $A = \sum_{i=0}^{K+1} A_i$, we have $f = (f_0, f_1, \ldots, f_m)$. Therefore:

$$fA = 0 \quad \& \quad fe = 1$$

The stability condition is:

$$fA_0e < \sum_{k=2}^{K+1} (k - 1)fA_1e$$
To get the total drift, we sum the previous equation for all $k$, $0 \leq k \leq K + 1$.

But what is $f_j$? Let us define $A = \sum_{i=0}^{K+1} A_i$, we have $f = (f_0, f_1, \ldots, f_{m-1})$. Therefore:

$$fA = 0 \quad \& \quad fe = 1$$

The stability condition is:

$$fA_0e < \sum_{k=2}^{K+1} (k - 1)fA_k e$$
Outline

Introduction
  Why we need the Matrix-Geometric Technique?
Matrix-Geometric in Action
  Key Idea
General Matrix-Geometric Solution
  General Concept
Application of Matrix-Geometric
  Performance Analysis of Multiprocessing System
Properties of Solutions
  Properties
Computational Properties of $R$
  Algorithm for solving $R$
It is an iterative method. Let

\[ R(0) = 0 \]

\[ R(n + 1) = - \sum_{l=0, l \neq 1}^{\infty} R_l^{(n)} A_l A_1^{-1} \quad n = 0, 1, 2, \ldots \]

The iterative process halts whenever entries in \( R(n + 1) \) and \( R(n) \) differ in absolute value by less than a given constant.

The sequence \( \{R(n)\} \) are entry-wise non-decreasing and converge monotonically to a non-negative matrix \( R \).

the number of iteration needed for convergence increases as the spectral radius of \( R \) increases. This is similar to the scalar case where \( \rho \to 1 \). As the system utilization increases, it becomes computationally more difficult to get
It is an iterative method. Let

\[ R(0) = 0 \]

\[ R(n + 1) = - \sum_{l=0, l \neq 1}^{\infty} R^l(n) A_l A_1^{-1} \quad n = 0, 1, 2, \ldots \]

The iterative process halts whenever entries in \( R(n + 1) \) and \( R(n) \) differ in absolute value by less than a given constant.

The sequence \( \{R(n)\} \) are entry-wise non-decreasing and converge monotonically to a non-negative matrix \( R \).

The number of iteration needed for convergence increases as the spectral radius of \( R \) increases. This is similar to the scalar case where \( \rho \to 1 \). As the system utilization increases, it becomes computationally more difficult to get
It is an iterative method. Let

\[ R(0) = 0 \]

\[ R(n + 1) = - \sum_{l=0, l \neq 1}^{\infty} R^l(n) A_l A_1^{-1} \quad n = 0, 1, 2, \ldots \]

The iterative process halts whenever entries in \( R(n + 1) \) and \( R(n) \) differ in absolute value by less than a given constant.

The sequence \( \{R(n)\} \) are entry-wise non-decreasing and converge monotonically to a non-negative matrix \( R \).

The number of iteration needed for convergence increases as the spectral radius of \( R \) increases. This is similar to the scalar case where \( \rho \to 1 \). As the system utilization increases, it becomes computationally more difficult to get...
It is an iterative method. Let

\[ R(0) = 0 \]

\[ R(n + 1) = - \sum_{l=0, l \neq 1}^{\infty} R^l(n) A_l A_1^{-1} \quad n = 0, 1, 2, \ldots \]

The iterative process halts whenever entries in \( R(n + 1) \) and \( R(n) \) differ in absolute value by less than a given constant.

The sequence \( \{R(n)\} \) are entry-wise non-decreasing and converge monotonically to a non-negative matrix \( R \).

The number of iteration needed for convergence increases as the spectral radius of \( R \) increases. This is similar to the scalar case where \( \rho \to 1 \). As the system utilization increases, it becomes computationally more difficult to get
Replicated Database

- Poisson arrival with rate $\lambda$.
- Probability it is a read request: $r$.
- A read request can be served by any server.
- A write request has to be served by BOTH servers.
- What is the proper state space?
Replicated Database

- Poisson arrival with rate $\lambda$.
- Probability it is a read request: $r$.
- A read request can be served by any server.
- A write request has to be served by BOTH servers.
- What is the proper state space?
Replicated Database

- Poisson arrival with rate $\lambda$.
- Probability it is a read request: $r$.
- A read request can be served by any server.
- A write request has to be served by BOTH servers.
- What is the proper state space?
Replicated Database

- Poisson arrival with rate $\lambda$.
- Probability it is a read request: $r$.
- A read request can be served by any server.
- A write request has to be served by **BOTH** servers.
- What is the proper state space?
Replicated Database

- Poisson arrival with rate $\lambda$.
- Probability it is a read request: $r$.
- A read request can be served by any server.
- A write request has to be served by BOTH servers.
- What is the proper state space?
The state space $S = (i, j)$ where $i$ is the number of queued customers and $j$ is the number of replications that are involved in service. So $i \geq 0$ and $j = 0, 1, 2$. 

\[
\begin{align*}
\lambda(1-r) & \quad 0,1 \\
\lambda & \quad 1,1 \\
\lambda & \quad 2,1 \\
\lambda(1-r) & \quad 0,2 \\
2\mu & \quad 1,2 \\
2\mu(1-r) & \quad 2,2 \\
\mu & \quad 0,0 \\
\mu & \quad 1,1 \\
\mu & \quad 2,1 \\
2\mu & \quad 0,2 \\
2\mu r & \quad 1,2 \\
2\mu r & \quad 2,2 \\
2\mu r & \quad \ldots \\
2\mu r & \quad \ldots
\end{align*}
\]
\[
Q = \begin{bmatrix}
-\lambda & \lambda r & \lambda(1 - r) & 0 & 0 & 0 & \cdots \\
\mu & -(\lambda + \mu) & \lambda r & \lambda(1 - r) & 0 & 0 & \cdots \\
0 & 2\mu & -(\lambda + 2\mu) & 0 & \lambda & 0 & \cdots \\
0 & 0 & \mu & -(\lambda + \mu) & 0 & \lambda & \cdots \\
0 & 0 & 2\mu r & 2\mu(1 - r) & -(\lambda + 2\mu) & 0 & \cdots \\
0 & 0 & 0 & 0 & \mu & -(\lambda + \mu) & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\]

\[
B_{00} = \begin{bmatrix}
-\lambda & \lambda r & \lambda(1 - r) \\
\mu & -(\lambda + \mu) & \lambda r \\
0 & 2\mu & -(\lambda + 2\mu)
\end{bmatrix};
B_{01} = \begin{bmatrix}
0 & 0 \\
\lambda(1 - r) & 0
\end{bmatrix};
B_{10} = \begin{bmatrix}
0 & 0 & \mu \\
0 & 0 & 2\mu r
\end{bmatrix};
B_{11} = A_1 = \begin{bmatrix}
-(\lambda + \mu) & 0 \\
2\mu(1 - r) & -(\lambda + 2\mu)
\end{bmatrix};
$A_0 = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}; A_2 = \begin{bmatrix} 0 & \mu \\ 0 & 2\mu r \end{bmatrix}$

To determine the stability:

$A = \begin{bmatrix} -\mu & \mu \\ 2\mu(1 - r) & -2\mu(1 - r) \end{bmatrix} = A_0 + A_1 + A_2$

$f_1 = \frac{2(1 - r)}{3 - 2r}; f_2 = \frac{1}{3 - 2r}$

$fA_0 e < \sum_{k=2}^{K+1} (k - 1)fA_k e \rightarrow \lambda < \frac{2\mu}{3 - 2r}$