An Overview of The Application of Heavy Traffic Theory and Brownian Approximations to the Control of Multiclass Queueing Networks

Elif Uysal

Stochastic Networks

- Jobs (packets, customents), servers (resources), queues (buffers)
- Often cannot be analyzed exactly
- Fluid and diffusion models as approximation methods
- Theory developed by Riemann, Harrison, Williams, Bramson, ... using fluid and diffusion approximations to analyze stability, performance of open multiclass HL queueing networks

Goals of this Overview

- Heavy traffic (HT) scaling
- HT analysis technique
- Convergence to Brownian motion (BM)
- Resource pooling (RP)
- State-space collapse (SSC)

Brownian Motion

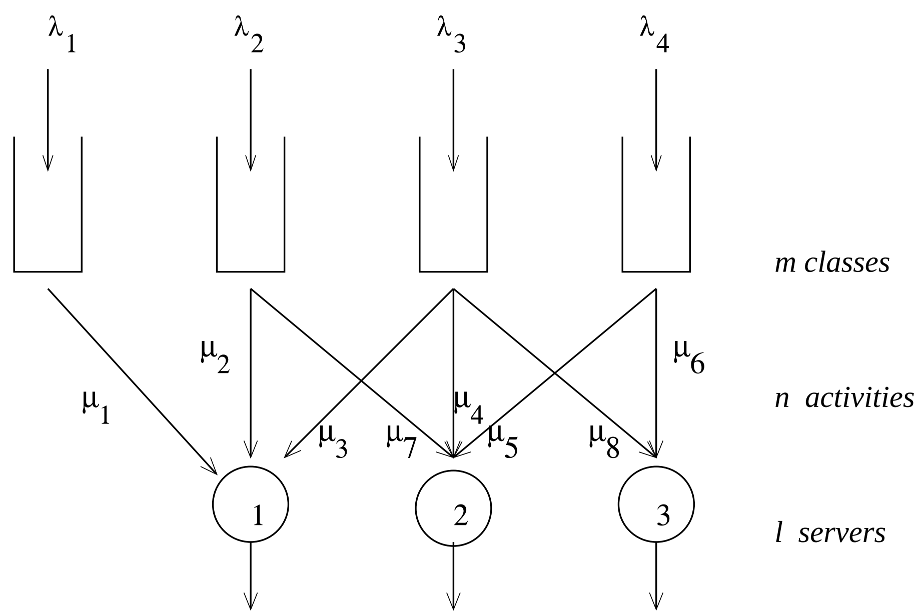
- A stochastic process $B(t)$ is a standard Brownian Motion if and only if it has
  - Continuous sample paths
  - Stationary increments: $B(t_1) - B(t_0)$ depends only on $t_1 - t_0$
  - Independent increments: $B(t_1) - B(t_0)$, $\ldots$, $B(t_n) - B(t_{n-1})$ are indep. for any $0 \leq t_1 < t_2 < \ldots < t_n < \infty$
  - Gaussian increments: $B(t_n) - B(t_{n-1}) \sim \mathcal{N}(0, t_{n-1} - t_n)$
- $Y(t) = Y(0) + \mu t + \sigma B(t)$ is a Brownian motion with *drift* μ and variance σ^2

General Procedure of Heavy Traffic Analysis

Methodology developed by [Reimann 1984], [Harrison 1988] and [Harrison and Williams 1992]:

1. Set up the dynamic scheduling problem, define "heavy traffic"
2. Derive a limiting Brownian control problem that plausibly approximates the original problem (after some scaling.)
3. Solve the Brownian control problem (much simpler than original problem)
4. Interpret the solution in original context
5. Ideally, prove that proposed solution is asymptotically optimal in the original problem.

A Parallel-Server System with Resource Pooling (Harrison and Lopez 1999)



- r job classes, n activities, l servers
- λ_i: avg. arrival rate of class i jobs; μ_j: reciprocal of mean service time for activity j
- Queue i incurs "holding cost" at a rate c_i per unit time per job in queue

A Parallel-Server System with Resource Pooling (Harrison and Lopez 1999)

- Problem: dynamically allocate jobs to servers to minimize avg holding cost per unit time

Heavy Traffic?

- Not obvious: multiple queues can share servers
- Consider:

$$\begin{aligned} \text{minimize} \quad & \rho \\ \text{subject to } Rx &= \lambda \\ Ax &\leq \rho e \\ x, \rho &\geq 0 \end{aligned}$$

where $A_{ij} = 1$ if server i serves activity j, $A_{ij} = 0$ o.w.
$R_{ij} = \mu_j$ if server i serves activity j, $R_{ij} = 0$ otherwise
x is $n \times 1$ the vector of fractions of times allocated to each activity by its server

Heavy Traffic

- **Heavy traffic assumption: The data (R, A, λ) of the static allocation problem are such that**
  - **its solution (x^*, ρ^*) is unique;**
  - $\rho^* = 1$**, and**
  - $Ax^* = e$**.**
- x^*: *nominal processing plan*
- Activities for which $x_j^* > 0$ are *basic activities*
- When $\rho^* = 1$, the system manager is just able, using the basic activities, to process jobs of the various classes at the required average rates

Dynamic Scheduling Question

Can server work assignments be *dynamically* adjusted, relative to the nominal processing plan x^*, to minimize cost?
Recall: queues have different holding cost rates

Resource Pooling

Definition: **Communicating Servers.** Server k communicates *directly* with server k' if there exist basic activities j and j' such that $j = j(i,k)$ and $j' = j(i,k')$ for some class i.
Server k *communicates* with server k' if there exist servers $k_1, \ldots, k_w$ such that $k_1 = k$, $k_w = k'$, and k_α communicates directly with $k_{\alpha+1}$ for all $\alpha = 1, \ldots, w-1$.

Resource Pooling

- Servers can be partitioned into disjoint communicating sets
- 1 Partition = CRP
- Intuition: Under CRP, can shift load from one queue to another
- Perhaps in heavy traffic scaled system this can be done in zero time?

Wait: We Need to Define HT Scaling

- $X^r(t) = X(rt)/\sqrt{r}$
- $F_i^0(t)$: number of arrivals into buffer i in $[0, t]$
- $F_i^j(t)$: number of departures from buffer i resulting from the first t time units devoted to activity j by its server
- The scaled process $F^0(rt)/\sqrt{r} - \lambda t\sqrt{r}$ converges weakly as $r \to \infty$ to a Brownian motion with zero limit and with an $m \times m$ covariance matrix Γ^0
- "Shrink time, expand space"

Convergence to Brownian Motion

- For all j and $t \geq 0$

$$F^j(rt)/\sqrt{r} - R^j t\sqrt{r}$$

converge weakly to BM with zero drift, covariance matrix Γ^j

Policy and System Dynamics

- Scheduling Policy: n-dimensional stochastic process $T = \{T(t), t \geq 0\}$
- $T_j(t)$: total time devoted to activity j over $[0, t]$
- The m-dimensional jobcount process:

$$Q(t) = F^0(t) - \sum_{j=1}^{n} F^j(T_j(t)), t \geq 0$$

The cost rate process:

$$C(t) = cQ(t)$$

Queuing System Dynamics

Under the HT assumption

- Define the centered allocation: $V(t) = x^*t - T(t)$
- Define "cumulative idleness process" $I(t)$: vector of cumulative idle time for the n activities
- All components of $I(t)$ must be nondecreasing for T to be an admissible policy
- Scaled processes

$$\begin{aligned} Y^r(t) &= V(rt)/\sqrt{r} \\ Z^r(t) &= Q(rt)/\sqrt{r} \\ U^r(t) &= I(rt)/\sqrt{r} \text{ and,} \\ \xi^r(t) &= C(rt)/\sqrt{r} \end{aligned}$$

Limiting Brownian control problem

- Under HT assumption, as $r \to \infty$, the scaled dynamic scheduling problem is well approximated by

$$Z(t) = X(t) + RY(t)$$

where $X = \{X(t), t \geq 0\}$ is the m-dimensional Brownian motion with zero drift, covariance matrix $\Gamma = \Gamma^0 + \sum_{j=1}^{b} x_j^* \Gamma^j$ [Reiman 1984]

- Note that $Y(t)$ is our n-dimensional control

- The problem:

$$\begin{aligned} Y & \quad \text{is non-anticipating with respect to } X \\ Z(t) &\geq 0 \text{ for all } t \geq 0, \text{ and} \\ U & \quad \text{is nondecreasing with } U(0) \geq 0 \\ Z(t) &= X(t) + RY(t) \text{ for all } t \geq 0 \\ U(t) &= KY(t) \text{ for all } t \geq 0 \end{aligned}$$

How big is the state space of this problem?

- The original state $Q(t)$ is an m-dimensional vector
- But the limiting Brownian problem will have a 1-dimensional state space!
- State space **collapse**

State Space Collapse [Harrison and Van Mieghem 1997]

- In the Brownian control problem apply an immediate impulse control $Y(0) = \delta$ at $t = 0$.
- Then, initial values $Z(0) = z + \delta$ and $U(0) = u$, where $\delta = Ry$ and $u = Ky$.
- The impulse control is admissible only of $z + \delta \geq 0$ and $u \geq 0$
- If $u = 0$ (i.e. $Ky = 0$), can immediately apply another control increment of $-y$ which returns the system to state z
- Thus, if $Ky = 0$, the control increment y is *reversible*
- Any two state vectors whose difference is a reversible displacement are equivalent!
- Summary of the system is given by $W(t) = MZ(t)$, where M is a matrix whose rows are orthogonal to all reversible displacements, meaning $MRy = 0$ if $Ky = 0$. (So, M gets rid of the reversible component of the system state.)

Equivalent Workload Formulation

- From SSC, we obtain "equivalent workload formulation"
- state of the system at any time t is described by a workload vector $W(t)$ having dimension $d \leq m$
- In our case $d = 1$. Specifically, $W(t) = y^* Z(t)$ where y^* is the unique optimal dual solution of the static allocation problem
- The simplified problem is to choose the pair (Z, U) such that:

$$\begin{aligned}
U & \quad \text{is non-anticipating with respect to } X \\
Z(t) &\geq 0 \text{ for all } t \geq 0, \text{ and} \\
U & \quad \text{is nondecreasing with } U(0) \geq 0 \\
W(t) &= \Psi(t) + GU(t) \text{ for all } t \geq 0 \\
W(t) &= y^* Z(t) \text{ for all } t \geq 0
\end{aligned}$$

Pathwise solution of Brownian control problem

- Define $L(t) = GU(t)$
- Recall $W(t) = \Psi(t) + GU(t)$
- Let $W^*(t) = \Psi(t) + L^*(t)$ where

$$L^*(t) = -\inf_{0 \le s \le t} \Psi(s), \ t \ge 0$$

- $W^*(t)$ is a regulated Brownian motion with a regulating barrier at 0

Pathwise solution of Brownian control problem

- Note that L^* is continuous, non-decreasing, $L^*(0) = 0$
- Note also that L^* is the **only** choice of feasible L's such that L increases only at times t when $W^*(t) = 0$
- So, for any admissible strategy (Z, U), the workload process W satisfies $W(t) \geq W^*(t)$ for all $t \geq 0$!
- That is, idle only when there is no work!
- Note that this is a pathwise bound
- Then, cost rate process satisfies, for any admissible strategy, $\xi(t) \geq \xi^*(t)$. Hence the policy (Z^*, U^*) is pathwise optimal

What is this policy doing?

- System manager tries to keep the system non-idle
- When workload is approaching zero, rather than idling, it engages some server on a non-basic activity. Since non-basic activities are inefficient, workload rises fast under Brownian scaling, so shortly the system manager can return to a mode in which all servers are fully occupied with basic activities.

Back to the original queuing system

- Authors conjecture ideal system behavior can be approached through a family of simple scheduling policies related to the $c\mu$ rule
- Rank classes in increasing order of the index $c_i\mu_i = c_i/y_i^*$
- Higher ranked classes given priority

Simplified case

- Multiple classes, single server, equal holding costs across classes [Harrison 1988]
- Solution: server should serve last the class k for which μ_k is smallest, and that this class gets service only when the system is empty of all other classes
- Note that in the Brownian limit this corresponds to only one queue being in heavy traffic, and a SSC to 1 dimension.