Information Processing Letters 135 (2018) 41–46

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Vacationing server model for M/G/1 queues for rebuild processing in RAID5 and threshold scheduling for readers and writers

Alexander Thomasian [1]

American University of Armenia, College of Science and Engineering, Yerevan 0019, Armenia

ARTICLE INFO

Article history:
Received 8 October 2016
Received in revised form 6 February 2018
Accepted 10 February 2018
Available online 13 February 2018
Communicated by P. Wong

Keywords:
Performance evaluation
Queueing theory
M/G/1 queues
Vacationing server model
Busy period
Delay cycle
Disk arrays
RAID5
Rebuild time
Readers and writers problem
Threshold scheduling

ABSTRACT

We present two methods to obtain the mean delay cycle for the M/G/1 queueing system with the Vacationing Server Model – VSM, which starts with an arrival during a vacation and ends when the queue is emptied and a vacation starts. In the case of VSM with multiple vacations the server returning from a vacation takes another vacation if the queue is empty, otherwise it starts serving requests. The M/G/1 queue has arrivals with rate λ, mean service time $\overline{x}$, so that its utilization factor is $\rho = \lambda\overline{x}$, and its mean busy period is $\overline{g} = \overline{x}/(1-\rho)$. The VSM delay cycle starts with a requested whose mean service time is augmented by the mean residual vacation time: $\overline{y} = \overline{x} + \overline{v}_r$, so that the mean delay cycle is $\overline{d}_v = \overline{y}/(1-\rho)$. This is the method used to determine rebuild time in RAID5 disk arrays. In a second study which deals with threshold scheduling of readers and writers $\overline{d}_v$ is obtained as the product of the mean number of requests arriving during the residual vacation time plus one (the request starting the residual vacation time) times $\overline{g}$, which yields $\overline{d}_v - \overline{v}_r = (1 + \lambda\overline{v}_r) \times \overline{g}$ as before. The analysis of VSM for rebuild processing in RAID5 and threshold scheduling of readers and writers is provided.

1. Introduction

We consider the *Vacationing Server Model – VSM* in an M/G/1 queueing system with Poisson arrivals and general service times [9], where an idle server takes multiple vacations when the request queue is emptied [9,19]. The server returning from a vacation takes another vacation if the queue is empty, but otherwise it starts serving requests again. Of interest is the mean delay cycle, denoted by $\overline{d}_v$, which is the time since the first arrival during the vacation period up to the time the queue is emptied [10].

VSM was applied in the context of RAID5 disk arrays with dedicated and distributed sparing in [22] and [24], respectively. We are interested in the effect of rebuild reads on the mean waiting time and the effect of the external requests on rebuild time.

VSM was applied to threshold scheduling of readers and writers in [20,21]. Readers are processed concurrently at a maximum degree of concurrency M, while writers cannot be processed concurrently with readers and each other. Writers arrive according to a Poisson process and when the processing of writers is completed the system starts processing readers during its vacation. There are M readers in a closed system and a completed reader is immediately replaced by a new reader. The processing of readers is stopped when the number of enqueued writers reaches the threshold K and resumes when the delay cycle for processing writers ends.

E-mail address: alexthomasian@gmail.com.
[1] Permanent address: Thomasian & Associates, 17 Meadowbrook Rd, Pleasantville, NY 10570, USA.

https://doi.org/10.1016/j.ipl.2018.02.011
0020-0190/

Provided writer service times are exponentially distributed the aforementioned system can be modeled as a two-dimensional *Continuous Time Markov Chain – CTMC*, where one dimension has M states to specify the number of active readers and the other an infinite number of states for the number of writers. The expression obtained for the mean number of writers ($\overline{N}_w$), obtained by solving the CTMC, which is given by Eq. (A.8) in [21] is different from the one obtained using VSM (see Eq. (3.8)), but both methods yield the same numerical value. VSM is applicable to the case where writers have a general service time, i.e., an M/G/1 queueing system as discussed in Section 4.

Different methods are used in [22] and [21] to determine the mean duration of the delay cycle ($\overline{d}_v$) and it is shown in Section 2 that both methods yield the same result.

Perhaps the earliest study of VSM is [1] where five models: A–E are proposed. We are considering Model E with vacations starting when there are no more requests to serve. An early application of VSM in the context of a paging drum with FIFO scheduling, which relies on the analysis in [18] is [7]. VSM is dealt with as a single homework problem in [9], but there are numerous variations which are reviewed in [19]. A recent text devoted solely to the theory and applications of VSMs is [31]. VSM is discussed in operations research texts on stochastic systems and queueing theory, such as Section 9.4 in [13].

Reader–writer queues with Poisson arrivals, with general i.i.d. service times, and alternating exhaustive priorities are considered in [15]. Readers can be processed simultaneously, while writers are processed one at a time. The system switches from readers to writers when there are more readers to be processed and vice-versa. This system is analyzed to produce the stability condition and the LST for the steady state queueing time of readers and writers. A queueing system with reader preference is analyzed in [14] using M/G/∞ busy period to model readers and a modified M/G/1 queue to model the entire system. A reader–writer queue under the following five priority disciplines, where a strong priority implies preemption, is considered in [16]: strong reader preference (SRP), reader preference (RP), alternating exhaustive priority (AEP), writer preference (WP), and strong writer preference (SWP). Stability condition and the means for the steady-state reader and writer queueing times are obtained. The operation of VSM with both writer and reader arrivals is shown in Fig. 1 in [25], which uses simulation to study the effect of separate thresholds for readers and writers to balance writer and reader mean response times. An approximate solution to the nonsaturated readers and writers problem appears in [33]. Section 5.1 on updating mirrored disks and Section 5.2 on synchronizing reads on updating replicated databases in [29] are relevant to this paper.

This paper is organized as follows. Section 2 is the core section of the paper, which provides an analysis of VSM in the context of the M/G/1 queueing system. Section 3 describes RAID5 disk arrays and provides the analysis used in [22] to obtain the rebuild time in RAID5 disk arrays and the effect of rebuild on the response time of disk requests. Section 4 describes the application of VSM to threshold scheduling of readers and writers. We conclude with Section 5.

2. Analysis of M/G/1 queues with VSM

We consider an M/G/1 queueing system with Poisson arrivals with for requests rate λ and exponentially distributed interarrival time with mean $\overline{t} = 1/\lambda$. Requests served in FCFS order have a general service time with *probability density function – pdf* $b(x)$, *Laplace–Stieltjes Transform (LST)* $B^*(s) = \int_{x=0^-}^{\infty} b(x)e^{-sx}dx$. The ith moment $\overline{x^i}$, the mean $\overline{x}$, and the mean residual service time $\overline{x}_r = \overline{x^2}/(2\overline{x})$. The utilization factor of the server is: $\rho = \lambda\overline{x}$ with the condition $\rho < 1$ to ensure a finite queue length [9].

An M/G/1 queueing system alternates between busy and idle periods. Noting that the fraction of time the server is busy is: $\rho = \overline{g}/(\overline{g} + \overline{t})$, we have $\overline{g} = \overline{x}/(1-\rho)$. The sum of these two periods is referred to as a cycle and has a mean: $(\overline{c}) = \overline{g} + \overline{t}$.

After completing each vacation the server checks if the queue is empty and if so takes another vacation and otherwise starts serving requests. The pdf of vacation time is $v(x)$, its LST $V^*(s) = \int_{0^-}^{\infty} v(x)e^{-sx}dx$, its ith moment $\overline{v^i}$, and the mean residual vacation time $\overline{v}_r = \overline{v^2}/(2\overline{v})$. In Section 3 we consider a VSM with multiple vacations and in Section 4 a VSM with a single vacation.

The Pollaczek–Khinchin formula for mean waiting time in M/G/1 queueing systems with FCFS scheduling is given as [9]:

$$W_{M/G/1} = \frac{\lambda\overline{x^2}}{2(1-\rho)}.$$

This formula is extended below to VSM by noting that *Poisson Arrivals See Time Averages – PASTA* [32]. The mean waiting time encountered by an arriving request with VSM in effect (W_{VSM}) is the mean delay due to service time of requests ahead of it in the queue: the mean queue length ($\overline{N}_q$) times the mean service time ($\overline{x}$), plus the mean residual service time ($\overline{x}_r$) if the server is busy (P[busy] $= \rho$) and the mean residual vacation time ($\overline{v}_r$) if the server is idle (P[idle] $= 1-\rho$).

$$W_{VSM} = \overline{N}_q\overline{x} + \rho\overline{x}_r + (1-\rho)\overline{v}_r.$$

It follows from Little's result [9] that $\overline{N}_{queue} = \lambda W_{VSM}$, hence:

$$W_{VSM} = \frac{\lambda\overline{x^2}}{2(1-\rho)} + \frac{\overline{v^2}}{2\overline{v}} = W_{M/G/1} + \frac{\overline{v^2}}{2\overline{v}}. \tag{1}$$

W_{VSM} is increased by the mean residual time of rebuild reads, which corresponds to Eq. (2.14a) in Chapter 2 in [19]. As an aside, the VSM analysis in [22] was adopted to the analysis of rebuild in RAID1 (mirrored disks) in [2], which uses Eq. (2.40a) in Chapter 2 in [19], which is for the case when the first request has an exceptional service time, but this is not the case here.

The mean busy period ($\overline{g}$) can be determined from its LST derived based on Takacs' analysis in [9]. Given

$$G^*(s) = B^*(s + \lambda - \lambda G^*(s)), \tag{2}$$

the mean busy period is $\overline{g} = -dG^*(s)/ds|_{s=0} = \overline{x}/(1-\rho)$.

In the case of VSM it is as if the first request has an exceptional service time, known as the initial delay Y_0, we have a delay cycle Y_c, which starts with the arrival of the first request at an empty M/G/1 queue and ends when the system is empty again. As shown in Fig. 3.1 in [10]) $Y_c = Y_0 + Y_b$, where Y_b is the delay busy period during which requests are served. Denoting the LSTs of Y_c, Y_0, and Y_b with $G_c^*(s)$, $G_0^*(s)$, and $G_b^*(s)$, it is shown in Section 3.3 in [10] that we have the following relationship among LSTs:

$$G_c^*(s) = G_0^*(s + \lambda - \lambda G^*(s)) \tag{3}$$

where $G^*(s)$ was given by Eq. (2).

In the case of VSM the mean service time of the first request is augmented with the mean residual delay: $E[Y_0] = \overline{x} + \overline{v}_r$. The mean value delay cycle is $E[Y_c] = -dG_c^*(s)/ds|_{s=0} = E[Y_0]/(1-\rho)$ and the delay busy period are $E[Y_b] = \rho E[Y_0]/(1-\rho)$ [11]. We use an argument similar to the one used in obtaining $\overline{g}$ to obtain the mean of the delay busy period: $E[Y_b] = \overline{d}_v - \overline{v}_r$:

$$\rho = \frac{\overline{d}_v - \overline{v}_r}{\overline{d}_v + 1/\lambda} \implies \overline{d}_v = \frac{\overline{x} + \overline{v}_r}{1-\rho}. \tag{4}$$

The analysis of a threshold scheduling policy in the context of the classical readers and writers problem uses a different method to obtain $\overline{d}_v$ by multiplying the mean duration of an ordinary M/G/1 busy period, i.e., $\overline{g} = \overline{x}/(1-\rho)$ by the number of requests $K + \overline{J}$ yields the delay cycle minus $\overline{v}_r$.

It can be shown using a method similar to the one described in [9] that the z-transform for the number of arrivals during a service time as given by Eq. (5.45) in [9], the number of arrivals during a vacation is $\alpha(z) = V^*(\lambda - \lambda z)$. The z-transform for the number of arrivals during the residual vacation time is [6]:

$$\beta(z) = \frac{1 - \alpha(z)}{\alpha^{(1)}(1)(1-z)}. \tag{5}$$

The mean number of arrivals during residual vacation time is then:

$$\overline{J} = \frac{\alpha^{(2)}(1)}{2\alpha^{(1)}(1)}. \tag{6}$$

If the vacations time is exponentially distributed with parameter μ, $V^*(s) = \mu/(s+\mu)$ and $\alpha(z) = \mu/d(z)$ with $d(z) = \lambda - \lambda z + \mu$. Given $\alpha(z)$ the first two moments of the number of arrivals can be obtained as follows:

$$\alpha(z) = \frac{\mu}{d(z)}, \quad \frac{d\alpha(z)}{dz}|_{z=1} = \frac{\lambda\nu}{d^2(z)}|_{z=1} = \frac{\lambda}{\mu},$$

$$\frac{d^2\alpha(z)}{dz^2}|_{z=1} = \frac{2\lambda^2\mu}{d^3(z)}|_{z=1} = \frac{2\lambda^2}{\mu^2},$$

where we have used Eq. (6) to obtain $\overline{J} = \lambda/\mu$, since $\overline{v}_r = \overline{v} = 1/\mu$ due to the memoryless property of the exponential distribution [9]. One is added to $\overline{J}$ to take into account the first request starting the residual vacation time. An alternative derivation of Eq. (4) is then:

$$\overline{d}_v = (1 + \lambda/\mu) \times \overline{x}/(1-\rho) + v_r.$$

For a general vacation time distribution:

$$\overline{J} = \frac{\alpha^{(2)}(1)}{2\alpha^{(1)}(1)} = \lambda\overline{v}_r. \tag{7}$$

Multiplying by $\overline{J}+1$ to take into account the arrival, which started the residual vacation time we obtain $\overline{d}_v$ previously given by Eq. (4).

$$\overline{d}_v = (\lambda\overline{v}_r + 1) \times \frac{\overline{x}}{1-\rho} + \overline{v}_r = \frac{\overline{x} + \overline{v}_r}{1-\rho}. \tag{8}$$

3. RAID5 disk array organization operation

Redundant Arrays of Independent Disks – RAID level 5 (RAID5) attains load balancing via striping, which partitions large files into strips placed in round-robin manner across the N disks in the array [3]. Erasure coding in RAID5 dedicates the capacity of one out of N disks to parity. One strip per stripe holds the *eXclusive-OR (XOR)* of the $N-1$ data strips in the stripe. Parity strips are placed in repeating left to right diagonals according to the left symmetric organization to balance disk loads for updating parity blocks [3].

The updating of small data blocks on disk by *Online Transaction Processing – OLTP* applications has a significant impact on performance due to the *Small Write Penalty (SWP)*, since it entails two disk accesses to read the old data and old parity blocks, unless they are cached, computes the parity, and uses two more disk accesses to write them.

After a single disk failure RAID5 disk arrays continue their operation in degraded mode. Each access to the failed disk requires the reading of corresponding blocks from all surviving disks, which are then XORed to reconstruct the missing block on demand. The doubling of the read load on surviving disks is reduced by using the *Clustered RAID – CRAID)* paradigm and setting the parity group size G to less than N, so that the load increase is $\alpha = (G-1)/(N-1) < 1$ [17].

Two methods to implement CRAID he *Balanced Incomplete Block Designs (BIBD)* and *Nearly random Permutations – NRP* are described in [28]. The load increase for a mixture of read and write requests in clustered RAID5 and RAID6 is quantified in [26]. The mean response time degradation in tolerating one and two disk failures in RAID5 and RAID6 disk arrays are quantified in [27].

Requests for on demand reconstruction of blocks on failed disks or unreadable sectors are modeled as *Fork–Join – F/J* requests. Techniques to determine the mean response time of F/J requests based on [22,24] and other studies are reviewed in [29]. The overall mean response time of disk requests is a weighted sum of ordinary and F/J requests according to their frequency. Eq. (1) can be used to determine the mean response of disk requests in RAID5. For this we need the first two moments of disk service time, which is the sum of seek time, rotational latency, and transfer time [27].

In dedicated sparing rebuild processing is a systematic reconstruction of the blocks of a failed disk on a spare disk [22], while distributed sparing allocates adequate empty space across disks to reconstruct the contents of a failed disk [24]. *Rebuild Units – RUs* are fixed size blocks serving as the units of reconstruction. In the case of a RAID5 with N disks $N-1$ RUs in a stripe are XORed to reconstruct the RU on the failed disk. We only consider dedicated sparing as in [22], since otherwise in distributed sparing disks need to be engaged in both reading and writing RUs [24]. An iterative solution for this purpose was developed and validated in [24]. The RU used in [22–24] was a track, which had a fixed size in disks without *Zoned Bit Recording – ZBR* [8]. An advantage of having tracks as the RU is that no rotational latency is incurred and the reading of the track can be started at any sector, which is the smallest unit of data storage. In the case of disks with ZBR the linear recording density is maintained at approximately same level, which results in tracks whose capacity is roughly proportional to the diameter of the track. Variable RU sizes complicates buffer management for RUs read to be XORed. The variability in track sizes can be taken into account by analyzing rebuild in multiple stages, which is used to take into account the fraction of materialized data blocks on the spare disk due to read redirection and updates to that disk [22].

Ordinary disk accesses are affected by the rebuild accesses, although rebuild accesses are processed at a lower priority. This is because disk requests are not generally preemptible. Partially preemptible rebuild accesses in the form of the split-seek option is considered and analyzed in [22], i.e., a track is not read after a seek to a track, if an arrival occurs while the seek is in progress. The effect of rebuild preemption during latency and transfer phases is investigated in [23]. Preemption lowers the response time of external requests that it is blocking, but more external requests are affected by the increased rebuild time, since the same track has to be visited several times to complete its reading, so that more external requests are processed in degraded and affected by rebuild processing [23].

Read requests to the failed disk entail higher response time. For example if disk response times are exponentially distributed with mean R, the mean response time of an $N-1$-way F/J request to reconstruct a block on a failed disk is $R^{F/J}_{N-1} = RH_{N-1}$, where the Harmonic sum $H_{N-1} = \sum_{n=1}^{N-1} 1/n$. Simply prioritizing the components of F/J requests results in a much lower mean response time for $N-1$-way F/J requests, but disk response times can be balanced by conditionally prioritizing tardy components of F/J requests [30].

3.1. Estimating rebuild time in RAID5 disk arrays

Rebuild time in RAID5 can be estimated using the analysis in Section 2. Let $\overline{K}$ denote the mean number of vacations taken per cycle or the number or read tracks. The mean duration of a cycle is the sum of a delay cycle and mean interarrival time: $\overline{c} = \overline{d}_v + \overline{t}$. Given that T denotes the number of disk tracks then the time to read a disk is $\overline{c} \times (T/\overline{K})$. Mean rebuild time can be approximated by the reading time of a single disk, since due to the load balance resulting from striping it takes about the same time to read all surviving disks (see Fig. 4 in [24] gives the coefficient of variation of rebuild time versus the arrival rate of external requests for distributed sparing). There is also the pipelining of rebuild writes with reads, so that the last rebuild write completes closely after the last rebuild read.

We next consider VSM with multiple vacations with different types. After a busy period corresponding to the processing of disk requests is completed, we have a sequence of vacations $V_i, i \geq 1$, e.g., the server returning from V_1 starts V_2 if there are no arrivals, i.e., the queue is empty, etc.

Our analysis is based on [5], which is repeated in [19]. The distribution of a type i vacations is $V_i(t)$, their LST $\mathcal{V}^*_i(s)$, and their jth moment $v_{i,j} = \int_0^\infty x^j dV_i(x)$. The probability of an arrival during the jth vacation is:

$$p_j = \left[1 - \int_0^\infty e^{-\lambda t} dV_j(t)\right] \prod_{k=1}^{j-1} \int_0^\infty e^{-\lambda t} dV_k(t)$$
$$= \left[1 - V^*_j(\lambda)\right] \prod_{k=1}^{j-1} V^*_k(\lambda).$$

The fraction of type i vacations is $q_i = \sum_{j=i}^\infty p_j/\overline{K}$, where $\overline{K}$ was defined above. The distribution of a typical vacation is $V(t) = \sum_{i=1}^\infty q_i V_i(t)$ and its mean is $v_j = \sum_{i=1}^\infty q_i v_{i,j}$.

The analysis in [22] considers three types of vacations, but only two types of vacations are considered here for brevity. Type 1 vacations involve a disk seek to the next track to be read and a full disk rotation to read a track (assuming the rebuild unit is a track). Type 2 vacations involve the reading of successive tracks without incurring a seek and are repeated until an arrival occurs. Type 2 vacations are not taken if an arrival occurs during the first (type 1) vacation. When there are no external requests all vacations will be of type 2 and rebuild time equals the number of tracks times disk rotation time. The probability of an arrival during the first and the jth vacation is as follows:

$$p_1 = 1 - V^*_1(\lambda),$$
$$p_j = [1 - V^*_2(\lambda)]V^*_1(\lambda)V^*_2(\lambda)]^{j-2}, \quad j \geq 2.$$

The mean number of vacations in this case is:

$$\overline{K} = 1 + \frac{V^*_1(\lambda)}{1 - V^*_2(\lambda)}$$

and the probability of the two types of vacations is

$$q_1 = \frac{1 - V^*_2(\lambda)}{1 + V^*_1(\lambda) - V^*_2(\lambda)}, \quad q_2 = 1 - q_1.$$

The ith moment of vacation time is:

$$\overline{v^i} = \sum_{j=1}^\infty q_j v_{i,j} = \frac{(1 - V^*_2(\lambda))v_{i,1} + V^*_1(\lambda)v_{i,2}}{1 - V^*_2(\lambda) + V^*_1(\lambda)}.$$

The mean residual vacation time is $\overline{v}_r = \overline{v^2}/(2\overline{v})$.

4. Threshold scheduling of readers and writers

Writers are mutually exclusive with one another and readers, so that writers are processed singly. while readers can process can be processed concurrently with a maximum degree of concurrency up to M, which is the number of servers.

Writers arriving according to a Poisson process with rate λ are processed until the writer queues is emptied, after which the processing of readers is started. The M readers are processed in a closed system, so that a completed reader is immediately replaced by a new reader. No new readers are activated when the number of enqueued writers exceeds the threshold K. The processing of in-progress readers continues until they are all processed and this is the start of vacations with K initial writers. We are interested in of additional writers enqueued in this period while emptying is in progress ($\overline{J}$). The processing of writers is started, after the processing of the last active reader is completed, and continues until there are no more writers to process. This constitutes a delay busy period, since the processing of the first writer is delayed.

Readers have exponentially distributed service times with parameter ν, which is required for mathematical tractability. Writers with Poisson arrivals with rate λ have a general service time distribution with LST $B^*(s)$, the moments of service time are $\overline{x^i}$, the mean $\overline{x}$. The fraction of time the systems processes writers is $\rho_w = \lambda\overline{x}$, which should be less than one to allow readers to be processed. The case when writer service times are also exponentially distributed is considered in [21].

Markovian Decision Processes – MDP is used in [4] to show that the *Threshold Fastest Emptying – TFE* policy optimizes the performance of the system under consideration. Higher values of K increases reader throughput, but this at the cost of mean writer response times, which is quantified in [21] by analyzing an M/G/1 VSM where readers are processed during vacations taken when the system is idle, having processed all writers. We are interested in the reader throughput (γ_K) and the mean number of writers in the system ($\overline{N}_w$), which can be used to obtain mean writer response time: $R_w = \overline{N}_w/\lambda$.

There are three processing phases if we start with a system with no writers.

Phase I: Readers are processed at a degree of concurrency M in a closed system. Each completed reader is immediately replaced by a new reader. No new readers are introduced when the writer queue length exceeds the threshold K. The duration of this phase is $T_I = K/\lambda$, which is the time that it takes for K writers to arrive. The number of readers processed is the duration of the interval multiplied by the processing rate: $N_I = T_I \times M\nu$.

Phase II: This is the reader emptying phase, which starts as soon as the Kth writer arrives. $N_{II} = M$ readers are processed in this phase. The duration of this phase is the maximum of M exponentials: $T_{II} = H_M/\nu$, where $H_M = \sum_{m=1}^{M} 1/m$ is the Harmonic sum. The mean number writers arriving during this phase is $\overline{J} = \lambda T_{II}$.

Phase III: The system emptied from readers starts processing writers, so that $N_{III} = 0$ readers are processed in this phase. The duration of the delay cycle starting with $K + \overline{J}$ writers is obtained by taking the derivative of $G^*(s) = B^*[s + \lambda - \lambda G^*(s)]^{(K+\overline{J})}$ to obtain the mean, which yields $T_{III} = (K + \overline{J})\overline{g}$, where $\overline{g} = \overline{x}/(1-\rho)$ is an ordinary busy period.

Reader throughput is the ratio of number of readers completed and the sum of durations of the three phases.

$$\gamma_K = \frac{N_I + N_{II} + N_{III}}{T_I + T_{II} + T_{III}} = (1-\rho_w)M\nu \frac{K + \frac{\lambda}{\nu}}{K + \frac{\lambda}{\nu}H_M}. \tag{9}$$

As $K \to \infty$ readers attain the maximum throughput, since they are processed at the maximum degree of concurrency without interruptions, so that $\gamma_\infty = (1-\rho_w)M\nu$. This will result in high delays in processing writers.

Our analysis of VSM with multiple identical vacations is based on problem 5.23 in [9], which is solved in [12]. The z-transform for the number of requests in M/G/1 with VSM, following the decomposition principle for VSM, is the sum of writers in an M/G/1 queue and the writers accumulated during vacations [6]: Hence we have the product of the respective z-transforms.

$$\begin{aligned} Q'(z) = \beta(z)Q(z) &= \frac{1-\alpha(z)}{\alpha^{(1)}(1)(1-z)} \cdot \frac{(1-\rho)(1-z)}{1 - z/\mathcal{B}^*(\lambda - \lambda z)} \\ &= \frac{(1-\alpha(z))(1-\rho)}{\alpha^{(1)}(1)(1 - z/\mathcal{B}^*(\lambda - \lambda z))}, \end{aligned} \tag{10}$$

where $Q(z) = [(1-\rho_w)(1-z)]/[1 - z/\mathcal{B}^*(\lambda - \lambda z)]$ is the z-transform of an M/G/1 queueing system and $\beta(z)$ is given by Eq. (5). In an M/G/1 queueing system with writers only,

$$\overline{N}_{M/G/1} = \frac{dQ(z)}{dz}|_{z=1} = \rho_w + \frac{\lambda^2\overline{x^2}}{2(1-\rho_w)}$$

and the mean number of requests arriving during the residual vacation time is given as follows (note we have applied L'Hospital's rule):

$$\begin{aligned} \tilde{N}_{vac} = \beta^{(1)}(z)|_{z=1} &= \frac{-\alpha^{(1)}(z)(1-z) + (1-\alpha(z))}{\alpha^{(1)}(1)(1-z)^2}|_{z=1} \\ &= \frac{0}{0} = \frac{-\alpha^{(2)}(z)(1-z) + \alpha^{(1)}(z) - \alpha^{(1)}(z)}{-2\alpha^{(1)}(1)(1-z)} \\ &= \frac{\alpha^{(2)}(1)}{2\alpha^{(1)}(1)}. \end{aligned} \tag{11}$$

The decomposition principle for VSM leads to:

$$\tilde{N}_w = \tilde{N}_{M/G/1} + \tilde{N}_{vac}. \tag{12}$$

Taking the mean and dividing both sides by λ yields:

$$R_w = R_{M/G/1} + \frac{\overline{\nu^2}}{2\overline{\nu}}.$$

Subtracting the mean service time ($\overline{x}$) from both sides yields Eq. (1).

Let $\tilde{J}$ denote the random variable for the number of writers accumulated when the system is in Phase II. Then there will be $K + \tilde{J}$ writers when vacations end. In the

case when reader processing times are exponentially distributed: $H(t) = 1 - e^{-\nu t}$ the residual lifetime is $F(t) = H(t)$

$$P[\tilde{J}] = \int_0^\infty \frac{(\lambda t)^j}{j!} e^{-\lambda t} d[F(t)^M]$$

The z-transform of $K + \tilde{J}$ is

$$\alpha(z) = M\nu z^K \sum_{m=0}^{M-1} (-1)^M \frac{\binom{M-1}{m}}{(M+1)\nu + \lambda(1-z)}$$

$$\bar{N}_w = Q^{\prime(1)}(1) = \frac{\lambda^2 \overline{x^2}}{2(1-\rho_w)} + \frac{\alpha^{(2)}(1)}{2\alpha^{(1)}(1)} \tag{13}$$

$$\alpha^{(1)}(1) = \frac{M}{\nu} \sum_{m=0}^{M-1} \frac{K(m+1)\nu + \lambda}{(m+1)^2}$$

$$\alpha^{(2)}(1) = \frac{M}{\nu^2} \sum_{m=0}^{M-1} (-1)^m \binom{M-1}{m} \times \frac{K(K-1)(m+1)^2\nu^2 + 2K(m+1)\nu\lambda + 2\lambda^2}{(m+1)^3}.$$

5. Conclusions

We have shown that two seemingly different methods yield the same result for the mean duration of the delay cycle ($E[Y_c]$). The two methods are utilized in analyzing the two systems described in Section 3, which deals with rebuild processing in RAID5 and Section 4, which deals with threshold scheduling of readers and writers. The first method is based on the standard queueing theory result for the mean duration of the delay cycle for VSM as the $E[Y_c] = (\overline{x} + \overline{v}_r)/(1-\rho)$, while the second method yields the delay busy period ($E[Y_b]$) by multiplying the number of arrivals during the mean residual vacation time plus one ($\lambda\overline{v}_r + 1$) by the mean duration of an ordinary busy period $\overline{g} = \overline{x}/(1-\rho)$.

Acknowledgements

Prof. Victor Nicola, currently at the University of Twente in the Netherlands collaborated with me at IBM Research Center at IBM Yorktown Heights in analyzing the readers and writers problem. Prof. Prudence Wong at the Computer Science Dept. at University of Liverpool successfully processed the paper.

References

[1] B. Avi-Itzhak, P. Naor, Some queuing problems with the service station subject to breakdown, Oper. Res. 11 (3) (June 1963) 303–320.
[2] E. Bachmat, J. Schindler, Analysis of methods for scheduling low priority disk drive tasks, in: Proc. ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, Marina del Rey, June 2002, pp. 55–65.
[3] P.M. Chen, E.K. Lee, G.A. Gibson, R.H. Katz, D.A. Patterson, RAID: high-performance, reliable secondary storage, ACM Comput. Surv. 26 (2) (1994) 145–185.
[4] C.A. Courcoubetis, M.I. Reiman, Optimal control of a queueing system with simultaneous service requirements, IEEE Trans. Autom. Control AC-32 (8) (August 1987) 717–727.
[5] B.T. Doshi, An M/G/1 queue with variable vacations, in: Proc. Modeling Techniques and Tools for Performance Analysis'85, North-Holland, The Netherlands, 1985, pp. 67–81.
[6] S.W. Fuhrmann, A note on the M/G/1 queues with server vacations, Oper. Res. 32 (6) (November–December 1984) 1368–1373.
[7] S.H. Fuller, F. Baskett, An analysis of drum storage units, J. ACM 22 (1) (Jan. 1975) 83–105.
[8] B. Jacob, S.W. Ng, D.T. Wang, Memory Systems: Cache, DRAM, Disks, Morgan-Kauffman Publisher, 2008.
[9] L. Kleinrock, Queueing Systems, Vol. I: Theory, Wiley-Interscience, 1975.
[10] L. Kleinrock, Queueing Systems, Vol. II: Computer Applications, Wiley-Interscience, 1976.
[11] L. Kleinrock, R. Gail, Solution Manual for Queueing Systems: Volume 2: Computer Applications, Technology Transfer Institute, 1986.
[12] L. Kleinrock, R. Gail, Queueing Systems: Problems and Solutions, John Wiley and Sons, 1996.
[13] H. Kobayashi, B.L. Mark, System Modeling and Analysis: Foundations of System Performance Evaluation, Pearson, 2008.
[14] V.G. Kulkarni, L.C. Puryear, A reader–writer queue with reader preference, Queueing Syst. 15 (1:4) (1994) 81–97.
[15] V.G. Kulkarni, L.C. Puryear, Stability and queueing time analysis of a reader–writer queue with alternating exhaustive priorities, Queueing Syst. 19 (1–2) (March 1995) 81–103.
[16] L.C. Puryear, V.G. Kulkarni, Comparison of stability and queueing times for reader–writer queues, Perform. Eval. 30 (4) (October 1997) 195–215.
[17] R.R. Muntz, J.C.S. Lui, Performance analysis of disk arrays under failure, in: Proc. Very Large DataBase (VLDB) Conf., Brisbane, Queensland, Australia, August 1990, pp. 162–173.
[18] C.E. Skinner, Priority queueing systems with server walking times, Oper. Res. 15 (2) (March–April 1967) 278–285.
[19] H. Takagi, Queueing Analysis Volume 1: Vacations and Priority Systems, North-Holland – Elsevier, 1991.
[20] A. Thomasian, V.F. Nicola, Analysis of a threshold policy for scheduling readers and writers (extended abstract), in: Proc. ACM SIGMETRICS Int'l Conference on Measurement and Modeling of Computer Systems, Berkeley, CA, May 1989, p. 237.
[21] A. Thomasian, V.F. Nicola, Performance evaluation of a threshold policy for scheduling readers and writers, IEEE Trans. Comput. 42 (1) (January 1993) 83–98.
[22] A. Thomasian, J. Menon, Performance analysis of RAID5 disk arrays with a vacationing server model for rebuild mode operation, in: Proc. IEEE Int'l Conf. on Data Engineering, Houston, TX, February 1994, pp. 111–119.
[23] A. Thomasian, Rebuild options in RAID5 disk arrays, in: Proc. 7th IEEE Symp. on Parallel and Distributed Processing, SPDP 1995, San Antonio, TX, October 1995, pp. 511–518.
[24] A. Thomasian, J. Menon, RAID5 performance with distributed sparing, IEEE Trans. Parallel Distrib. Syst. – TPDS 8 (6) (June 1997) 640–657.
[25] A. Thomasian, A multithreshold scheduling policy for readers and writers, Inf. Sci. 104 (3–4) (1998) 157–180.
[26] A. Thomasian, Clustered RAID arrays and their access costs, Comput. J. 48 (6) (2005) 702–713.
[27] A. Thomasian, G. Fu, C. Han, Performance of two-disk failure-tolerant disk arrays, IEEE Trans. Comput. 56 (6) (June 2007) 799–814.
[28] A. Thomasian, M. Blaum, Higher reliability redundant disk arrays: organization, operation, and coding, ACM Trans. Storage (TOS) 5 (3) (2009), Article No. 7.
[29] A. Thomasian, Analysis of fork/join and related queueing systems, ACM Comput. Surv. 47 (2) (August 2014), 17:1–17:71.
[30] A. Thomasian, B. Liu, Y. Deng, Balancing disk access times in RAID5 disk arrays in degraded mode by conditionally prioritizing fork/join requests, ACM SIGARCH Comput. Archit. News (CAN) 42 (2) (2014) 15–19.
[31] N. Tian, Z.G. Zhang, Vacation Queueing Models: Theory and Application, Springer, 2006.
[32] R.W. Wolff, Poisson arrivals see time averages, Oper. Res. 30 (1982) 223–231.
[33] E. Xu, A.S. Alfa, A vacation model for the non-saturated readers and writers system with a threshold policy, Perform. Eval. 50 (4) (2002) 233–244.