

基于复杂排队系统的三值光学计算机服务模型与性能分析*

徐 群¹, 王先超²

(1. 上海大学 计算机工程与科学学院, 上海 200444;

2. 阜阳师范学院 计算机与信息工程学院, 安徽 阜阳 236037)

摘要:三值光学计算机的运算请求处理过程缺乏合理、系统的性能评价标准与体系。基于 $M/M/1$ 、 $M/M/n$ 、 $M^X/M/1$ 和 $M/M^B/1$ 构成的复杂排队系统, 构建三值光学计算机的四阶段服务模型, 同时建立立即调度和结束时调度两种策略和算法。基于不同排队系统讨论运算请求的接收时间、预处理时间、运算时间和发送时间的计算方法, 进而得到最终响应时间。通过仿真实验对两种策略的模型进行验证, 结果表明, 结束时调度策略明显优于立即调度策略。

关键词:三值光学计算机; 复杂排队系统; 立即调度; 结束时调度; 响应时间

中图分类号:TP301 **文献标志码:**A **文章编号:**1001-2486(2017)02-140-06

Service model and performance analysis of ternary optical computer based on complex queuing system

XU Qun¹, WANG Xianchao²

(1. School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China;

2. School of Computer and Information Engineering, Fuyang Normal University, Fuyang 236037, China)

Abstract: The request processing of ternary optical computer lacked reasonable systematic standard and system for performance evaluation. Therefore, a four-stage service model was built based on complex queuing systems including $M/M/1$, $M/M/n$, $M^X/M/1$ and $M/M^B/1$. Meanwhile, strategy and algorithm of instant-scheduling and end-scheduling were proposed in detail. Based on different queuing systems, the calculation method of receiving time, preprocessing time, operating time and transmitting time of the request were discussed, and the final response time was obtained. Finally, the models of two strategies were verified by simulation experiments. Results show that the end-scheduling strategy is superior to the instant-scheduling strategy.

Key words: ternary optical computer; complex queuing system; instant-scheduling; end-scheduling; response time

排队论作为随机运筹学与应用概率论中的重要分支学科, 理论体系发展成熟, 研究成果已广泛应用于军事、运输、生产等领域^[1-2]。近年来, 国内外学者运用排队论对一些并行计算平台的系统性能进行了研究。Khzaei 等基于 $M/G/m/m+r$ 系统和近似 Markov 链构建了云计算服务性能的近似分析模型^[3]; Cao 等基于 $M/M/m$ 系统对多核服务器进行建模, 优化了能量分配与负载分布问题^[4]; Bai 等构建了由两个串联排队系统组成的复杂排队模型以评价异构数据中心的性能^[5]。由此可见, 排队论提供了一个有组织的、结构的、简洁的框架去研究、模拟和评价并行计算平台的服务过程。

三值光学计算机 (Ternary Optical Computer, TOC) 的研究源于 2003 年^[6-7], 经过十余年的探讨, 在理论和实验上取得了一系列成果^[8-9], 而关于 TOC 服务质量的研究近乎空白。

TOC 作为一种新型计算模式, 所提供的服务类似于云计算, 与此同时, 该服务又有不同于云计算等并行平台的特色: 第一, 主要直接或间接处理二元三值逻辑运算^[10]; 第二, 巨位性和并行性^[11]; 第三, 运算请求的多样性、处理器的动态重构性和负载的时间依赖性^[12]。这些特性决定了不能直接将研究云计算性能的方法照搬到 TOC 上, 故基于排队网络研究 TOC 的服务性能, 以确保提供更好的服务质量。

* 收稿日期: 2016-10-10

基金项目: 国家自然科学基金资助项目(61073049, 61572305, 61672006); 安徽省教育厅重点资助项目(KJ2015A182, KJ2015A191)

作者简介: 徐群(1985—), 女, 山东青岛人, 博士研究生, E-mail: 775088224@qq.com;

王先超(通信作者), 男, 副教授, 博士, 硕士生导师, E-mail: wxcdx@126.com

1 TOC 服务模型

TOC 服务模型如图 1 所示。用户以高级语言的文本形式向 TOC 的接收服务器 (Receiving Server, RS) 提交运算请求。假设队列容量足够大,当请求到达时,按先到先服务 (First Come First Service, FCFS) 策略排队。而后数据预处理服务器 (Pre Processing Server, PPS) 对各请求进行预处理,计算每个运算的运算量,将数据转换成三值数据^[13],处理后的运算请求同样按 FCFS 策略排

队。调度器 (Scheduler, S) 按 FCFS 策略同时调度若干预处理后的请求,资源分配器 (Allocator, A) 分配光学处理器 (Optical Processor, OP) 资源,同时将分配结果和重构码发送至 OP。OP 的重构部件完成重构后处理运算请求,解码服务器 (Decoder Server, DS) 对处理输出进行解码以生成运算结果。最后,发送服务器 (Transmitting Server, TS) 将运算结果发送至对应的客户端。图 1 中的 Stage 1 ~ 4 构成了四个阶段的复杂排队系统,即串行排队网络^[14]。

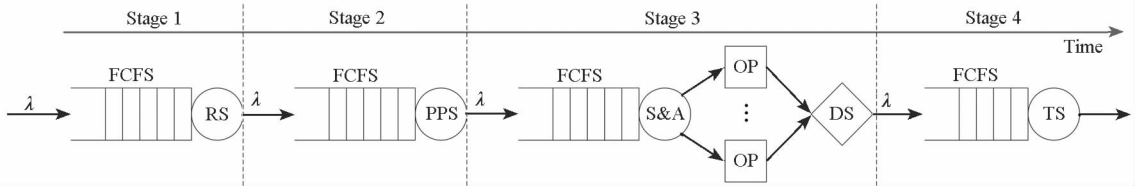


图 1 三值光学计算机服务模型

Fig. 1 Service model of ternary optical computer

2 调度算法和分配策略

OP 的应用特色在于众位数和可重构性,能在数据位的不同区段上构造出不同的逻辑运算器,并行处理多个运算要求。因此,采用不同的运算请求调度算法和数据位数分配策略,直接影响着系统的资源利用率及运算时间。

2.1 立即调度算法

算法 1 立即调度算法 (Instant Scheduling, IS) 将 OP 的数据位平均分成 n 部分,当运算请求到来时,若有小光学处理器空闲,则调度器 S 立即按 FCFS 策略将其调度到 TOC,否则它将排队等候,直到有小光学处理器空闲为止。

算法 1 立即调度算法

Alg. 1 Instant scheduling algorithm

已知: n 个小光学处理器,调度队列 Q

1. 初始化。待处理任务数 $N_{\text{Proc}} = 0$, Q 的长度 $L_Q = 0$;
2. 当任务到达 Q 时, L_Q 增 1, 转步骤 3;
3. 判断 N_{Proc} 是否为 n , 若是, 转至步骤 5; 否则判断 L_Q 是否为 0, 若为 0, 则转至步骤 5; 否则从 Q 中调度一个任务, L_Q 减 1, N_{Proc} 增 1;
4. S 接收到“任务完成”信号时, N_{Proc} 减 1, 转至步骤 3;
5. 算法结束。

调度完成后,资源分配器 A 按照算法 2 的按比例分配策略将一个小光学处理器的数据位分配给已调度任务的各运算。

算法 2 立即调度分配算法

Alg. 2 Instant scheduling allocation algorithm

已知: 光学处理器的数据位总数 N , 每个小光学处理器的数据位数 $N_{\text{DB}} = N/n$, 任务中二元三值逻辑运算个数 N_{Log}

1. 初始化。 $i = 1$, 任务运算量 $C = 0$;
2. 判断 i 是否大于 N_{Log} 。若是, 转至步骤 3; 否则 $C = C + C_i$ (C_i 表示 PPS 计算出的第 i 个运算的运算量), i 增 1, 转至步骤 2;
3. $i = 1$;
4. 判断 i 是否大于 N_{Log} 。若是, 转至步骤 5; 否则按比例分配数据位, 即 $N_i = \lfloor \frac{C_i}{C} \times N_{\text{DB}} \rfloor$, i 增 1, 转至步骤 4;
5. 算法结束。

2.2 任务结束时调度算法

当任务到达率和运算量都较小时,可能造成某些小光学处理器空闲,致使整个光学处理器的利用率降低,从而影响运算时间,因此提出任务结束时调度 (End Scheduling, ES) 策略的算法 3 和算法 4。

算法 2 和算法 4 中的光学处理器资源分配算法称为按比例分配算法,二者的区别在于:算法 2 每次分配一个小光学处理器的资源,而算法 4 每次都分配所有的光学处理器资源。显然,对于运算量相同的第一个任务,ES 策略下的处理时间约是 IS 策略下的 $1/n$ 。

算法 3 任务结束时调度算法

Alg.3 End scheduling algorithm

- 已知:TOC 最多可同时处理的任务数 n , 调度队列 Q
1. 参数初始化。已调度的任务数 $N_{Sched} = 0$;
 2. 第一个任务到达 Q 时, 调度器 S 采用立即调度策略调度它, N_{Sched} 增 1;
 3. S 接收到“任务结束”信号时, $N_{Sched} = 0$;
 4. 判断 Q 中的任务数 N_Q 是否为 0。若是, 则转步骤 6, 否则从 Q 中调度一个任务, N_{Sched} 增 1;
 5. 判断 N_{Sched} 是否大于 n 。若是, 将 N_{Sched} 发送至资源分配器 A, 而后转步骤 6, 否则转步骤 4;
 6. 算法结束。

算法 4 任务结束时调度分配算法

Alg.4 End scheduling allocation algorithm

- 已知:第 i 个任务的二元三值逻辑运算个数 N_{iLog} , OP 的数据位总数 N , 需调度的任务数 N_{Sched}
1. 参数初始化。 $i = 1, j = 1, N_{Sched}$ 个任务的总运算量 $C = 0$;
 2. 判断 i 是否大于 N_{Sched} 。若是, 转至步骤 5; 否则转至步骤 3;
 3. 判断 j 是否大于 N_{iLog} 。若是, 转至步骤 4; 否则 $C = C_{ij} + C, j$ 增 1, 转至步骤 3;
 4. i 增 1, 转至步骤 2;
 5. $i = 1, j = 1$;
 6. 判断 i 是否大于 N_{Sched} 。若是, 转至步骤 9; 否则转至步骤 7;
 7. 判断 j 是否大于 N_{iLog} 。若是, 转至步骤 8; 否则按比例第 i 个任务的第 j 个运算分配数据位 $P_{ij} = \lfloor \frac{C_{ij}}{C} \times N \rfloor, j$ 增 1, 转至步骤 7;
 8. i 增 1, 转至步骤 6;
 9. 算法结束。

3 响应时间建模

TOC 系统达到平衡状态时, 响应时间 T 的计算公式为:

$$T = T_R + T_p + T_C + T_T \quad (1)$$

其中: T_R 表示接收运算请求时间, 指从用户提交运算请求到接收完运算请求的时间; T_p 表示数据预处理时间; T_C 表示运算时间, 指运算请求从被调度到完成解码的时间; T_T 表示发送时间, 指将运算结果发送至用户客户端的时间。同时假设它们彼此统计独立。

3.1 接收时间

假设 RS 接收请求的到达时间间隔服从参数为 λ 的负指数分布, RS 服务时间服从参数为 μ_R 的负指数分布, μ_R 与网络传输速度 ω 和各运算请求平均传输量 η 有关, 满足 $\mu_R = \omega/\eta$ 。用 M/M/1 排队系统对 Stage 1 建模, RS 服务模型的时齐 (Continuous-Time Markov Chain, CTMC) 状态转移图如图 2 所示。

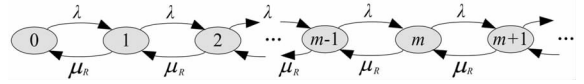


图 2 RS 的 M/M/1 模型状态转移图

Fig.2 State transition diagram of M/M/1 model in RS

当 $\rho_R = \lambda/\mu_R < 1$ 时, 存在平稳分布^[15-16]。记第 m 个状态的概率为 $P_m (m = 0, 1, 2, \dots)$, 由 Kolmogorov-chapman 方程, 可得:

$$\begin{cases} \lambda P_0 = \mu_R P_1 \\ (\lambda + \mu_R) P_m = \lambda P_{m-1} + \mu_R P_{m+1}, m \geq 1 \end{cases} \quad (2)$$

解得 $P_m = \rho_R^m P_0, m \geq 1$, 结合正则性 $\sum_{m=0}^{\infty} P_m = 1$, 得 RS 的空闲概率 $P_0 = 1 - \rho_R$ 。于是, RS 的平均运算请求数为:

$$N_R = \sum_{i=0}^{\infty} iP_i = \rho_R(1 - \rho_R) \sum_{i=0}^{\infty} i\rho_R^{i-1} = \frac{\lambda}{\mu_R - \lambda} \quad (3)$$

由 Little 公式, 得接收运算请求的平均时间:

$$T_R = \frac{N_R}{\lambda} = \frac{1}{\mu_R - \lambda} \quad (4)$$

3.2 预处理时间

由文献[15-16]可知, 运算请求到达 PPS 的时间间隔仍服从参数为 λ 的负指数分布。假设 PPS 将运算转换成三值逻辑运算的运算量为 C , 将数据转化成控制内码的速率为 ν , 则其服务速率 $\mu_p = \nu/C$ 。PPS 处理的运算请求都是 RS 接收的, 故建模系统可使用 3.1 节的结论, 得:

$$T_p = \frac{1}{\mu_p - \lambda} \quad (5)$$

3.3 运算时间

3.3.1 立即调度策略的运算时间

采用 IS 策略时, Stage 3 可用 M/M/n 排队系统进行建模, 时齐 CTMC 状态转移图如图 3 所示。其中 τ 表示整个 OP 的运算速度, 整个 OP 的平均服务强度 $\mu = \tau/C$, 每个小光学处理器及相应解码器的服务强度 $\mu_c = \mu/n$ 。

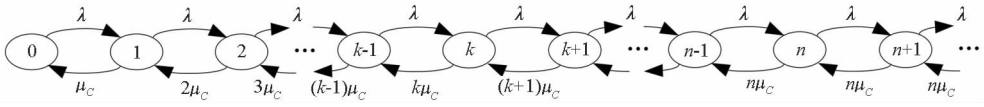


图 3 IS 策略下 OP 服务模型的状态转移图

Fig. 3 State transition diagram of OP service model under IS strategy

该策略下 Stage 3 的运算请求到达率也为 λ 。

记 $\rho_{c1} = \frac{\lambda}{\mu_c} = \frac{n\lambda}{\mu}$, $\rho_c = \frac{\rho_{c1}}{n} = \frac{\lambda}{\mu}$ 。当 $\rho_c < 1$ 时,系统存在平稳状态。由图 3 可列出平衡条件下的 Kolmogorov-chapman 方程组(6),再结合正则性,得到空闲概率 P_0 :

$$P_k = \begin{cases} \frac{\rho_{c1}^k P_0}{k!} = \frac{(n\rho_c)^k P_0}{k!}, & 0 \leq k < n \\ \frac{\rho_{c1}^k}{n!} P_0 = \frac{n^n \rho_c^k P_0}{n!}, & n \leq k \end{cases} \quad (6)$$

平均运算请求数 N_c 与运算时间 T_c 为:

$$N_c = \frac{\rho_c \rho_{c1} P_0}{n! (1 - \rho_c)^2} + \rho_{c1} \quad (7)$$

$$T_c = \frac{N_c}{\lambda} \quad (8)$$

3.3.2 任务结束时调度策略的运算时间

采用 ES 策略时,若 Q 中存在 $i(0 \leq i \leq n)$ 个任务,调度后整个 OP 被分成 i 个小光学处理器,其服务速率为 $\mu_i = \mu/i$ 。采用 $M/M^B/1$ 系统进行建模^[15], B 为批服务最大运算请求数(即 TOC 最多可同时处理的任务数 n),状态转移图如图 4 所示。

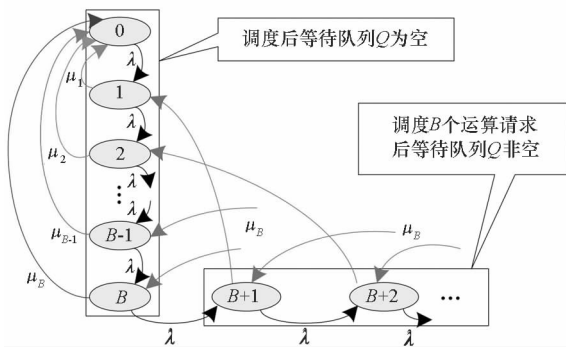


图 4 ES 策略下 OP 服务模型的状态转移图

Fig. 4 State transition diagram of OP service model under ES strategy

根据图 4 可得平衡条件下的 Kolmogorov-chapman 方程组:

$$\begin{cases} -\lambda P_0 + \mu(P_1 + P_2 + \dots + P_B) = 0 \\ -(\lambda + \mu)P_k + \lambda P_{k-1} + \mu P_{k+B} = 0, k \geq 1 \end{cases} \quad (9)$$

用算子法^[15]改写式(9)得:

$$[\mu D^{B+1} - (\lambda + \mu)D + \lambda]P_k = 0, k \geq 0 \quad (10)$$

令特征方程 $f(x) = \mu x^{B+1} - (\lambda + \mu)x + \lambda = 0$,该方程在 $(0, 1)$ 上存在唯一的根 x_0 ^[15],得:

$$P_k = (1 - x_0)x_0^k, k \geq 0 \quad (11)$$

$$T_c = \frac{x_0}{\lambda(1 - x_0)} \quad (12)$$

3.4 发送时间

假设运算结果平均大小为 R ,则 TS 的平均服务速率 $\mu_T = \omega/R$,分两种情况讨论发送时间。

3.4.1 立即调度策略的发送时间

IS 策略下,运算结果按到达率 λ 一个个地到达 TS,用 $M/M/1$ 排队系统建模,得:

$$T_T = \frac{1}{\mu_T - \lambda} \quad (13)$$

3.4.2 任务结束时调度策略的发送时间

ES 策略下,运算结果成批地到达 TS,故用 $M^X/M/1$ 排队系统建模^[15], X 表示每批到达的运算请求结果数。TS 模型的状态转移图如图 5 所示,其中 $X \in \{1, 2, 3, 4\}$, $\lambda_i(i=1, 2, \dots, B)$ 表示每批到达 i 个运算请求的速率, $\lambda = \sum_{i=1}^B \lambda_i$,每批到达运算请求的批大小为 i 的概率 $P_{Bi} = \lambda_i/\lambda$,可得:

$$\begin{cases} -\lambda P_0 + \mu_T P_1 = 0 \\ -(\lambda + \mu_T)P_k + \mu_T P_{k+1} + \lambda \sum_{i=1}^k P_{k-i} P_{Bi} = 0, k \geq 1 \end{cases} \quad (14)$$

当 $k > B$ 时, $P_{Bi} = 0$ 。采用生成函数法^[15]求解式(14),得到每批到达运算请求的均值 $E(X)$:

$$E(X) = \lim_{z \rightarrow 1} \sum_{i=1}^{\infty} iP_{Bi}z^{i-1}, |z| \leq 1 \quad (15)$$

令 $\rho_T = \frac{\lambda E(X)}{\mu_T}$,由 Little 公式得:

$$T_T = \frac{E(X) + E(X^2)}{2\mu_T(1 - \rho_T)} \quad (16)$$

最后,将式(4)、式(5)、式(8)、式(13)代入式(1)得 IS 策略的系统平均响应时间,将式(4)、式(5)、式(12)、式(16)代入式(1)得 ES 策略的系统平均响应时间。

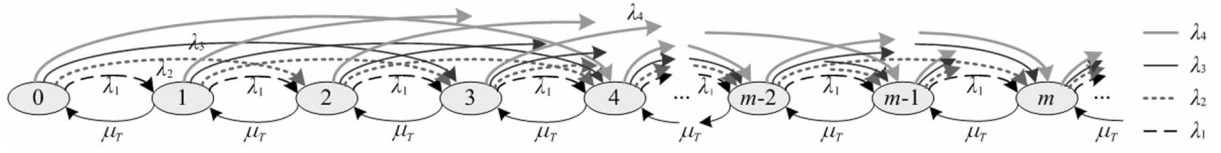


图 5 ES 策略下发送模型的状态转移图

Fig. 5 State transition diagram of transmitting model under ES policy

4 性能仿真

4.1 不同调度策略下的系统响应时间

本节通过仿真实验研究不同调度策略及参数对系统响应时间的影响。实验参数取值如下： $\lambda \in$ 整数区间 $[1, 60]$, $\omega = 20$ MB/s, $\eta = 0.01$ MB, $R = 0.05$ MB, $C = 50$ G, $v = 1$ GB/s, $\tau = 2$ GB/s, $\rho = \max\{\rho_R, \rho_P, \rho_C, \rho_T\}$, $\mathbf{Z} = [z_1 \ z_2 \ z_3 \ z_4]$ 表示不同 λ 的调度向量。OP 每批处理请求数最大值 $B=4$, 为便于比较两种策略同时也令 $n=4$ 。根据各参数绘制的 λ 对 T 的影响如图 6 和表 1 所示。

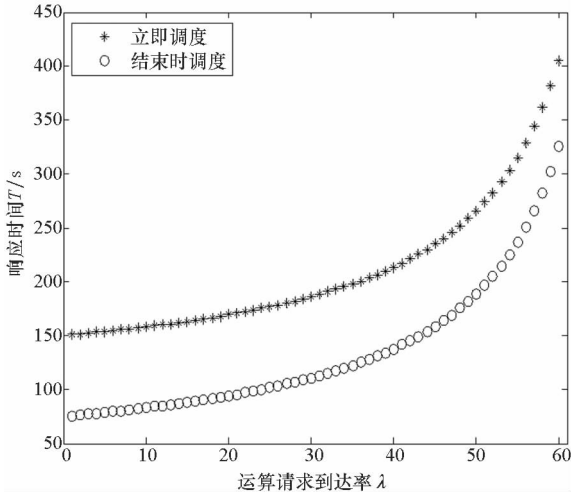


图 6 运算请求到达率对响应时间的影响

Fig. 6 Effect of request arrival rate on response time

表 1 运算请求到达率对响应时间的影响

Tab. 1 Effect of request arrival rate on response time

运算请求到达率 λ	响应时间		
	IS 策略/s	ES 策略/s	比值
1	150.705	75.705	1.991
2	151.429	76.430	1.981
3	152.175	77.176	1.972
...
58	361.353	282.440	1.279
59	381.432	302.229	1.262
60	404.826	325.322	1.244

可以看出:① T 随 λ 的增加而增加,任务呈现排队现象。②对于 λ 的任一取值,ES 策略的响应

时间明显优于 IS 策略,原因是当系统中运算请求数比较少时,IS 策略不能充分利用 OP 资源。③随着 λ 的增加,二者响应时间的比值呈减小趋势,原因是当 λ 较高时,IS 策略会提高 OP 的利用率,从而使二者差距变小。

运算量从 10 G 增至 50 G,其他条件不变,两种策略下系统响应时间如图 7 所示。容易看出,对于相同的运算请求到达率和运算量,ES 策略的响应时间都明显优于 IS 策略。

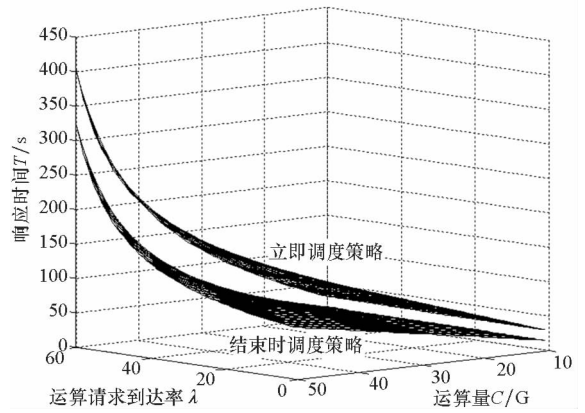


图 7 不同策略下 C 和 λ 对 T 的影响

Fig. 7 Effect of C and λ on T under different strategies

4.2 n 或 B 对响应时间的影响

在只改变 n 或 B 而其他参数不变的情况下,仿真实验得到的响应时间如图 8 和表 2 所示。

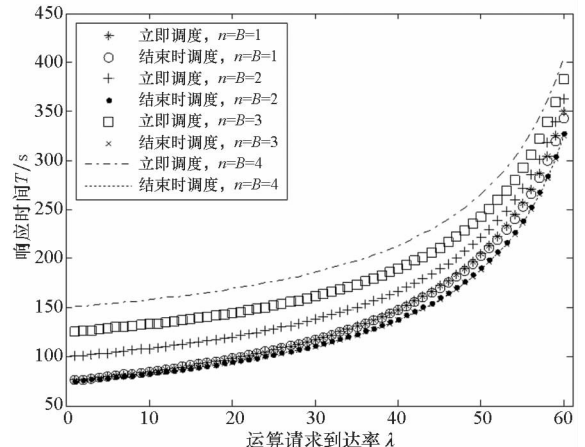


图 8 n 或 B 对不同调度策略下 T 的影响

Fig. 8 Influence of n or B on T under different strategies

容易得出:①当 n 或 B 为 1, $\lambda \leq 20$ 时, IS 策略的性能稍优于 ES 策略,但是当 $\lambda \geq 30$ 时, IS 策略在调度时将不断中断正在运行的任务,导致性能下降,而 ES 策略能有效缩短时间。②当 n 或 B 增加时,对于相同的 λ , 处理器资源利用率的减少导致

IS 策略的响应时间增加,而 ES 策略通过减少调度次数使系统性能表现更优。③将 1024 位光学处理器均分成 4 个小光学处理器,其性能基本达到最优。④因为 TS 待传输的运算结果的量很小,所以调度向量 Z 对响应时间的影响也很小。

表 2 n 或 B 对两种不同策略响应时间的影响Tab. 2 Influence of n or B on response time under two different strategies

调度策略	n 或 B	响应时间/s						
		$\lambda = 1$	$\lambda = 10$	$\lambda = 20$	$\lambda = 30$	$\lambda = 40$	$\lambda = 50$	$\lambda = 60$
IS 策略	1	75.847	84.682	98.008	117.418	148.239	205.067	349.691
	2	100.707	108.285	120.157	137.957	166.895	221.400	363.008
	3	125.705	133.098	144.487	161.566	189.539	242.773	382.668
	4	150.705	158.070	169.306	186.057	213.507	265.989	404.826
ES 策略	1	75.880	84.944	98.316	117.410	147.321	202.254	343.315
	2	75.706	83.185	94.626	111.537	138.878	190.691	327.840
	3	75.705	83.079	94.267	110.766	137.574	188.777	325.231
	4	75.705	83.079	94.267	110.752	137.571	188.776	325.209

5 结论

TOC 的性能分析与评价是其走向实际应用的重要一环,因此应用排队论建立 TOC 服务模型,以响应时间为指标评价其性能。基于 M/M/1 和 M/M/n 系统讨论了 IS 策略下 TOC 响应时间的计算方法;基于 M/M/1、M^x/M/1 和 M/M^B/1 系统探讨了 ES 策略下 TOC 响应时间的计算方法。仿真实验表明,ES 策略下 TOC 的性能明显优于 IS 策略。以后将基于排队论研究 TOC 的利用率、平均队长等性能指标。

参考文献 (References)

- [1] 俞政,周令毕. 一个输入依赖于系统状态的排队模型的瞬时队长分布[J]. 国防科技大学学报, 2005, 27(1): 106-110.
YU Zheng, ZHOU Lingbi. Distribution of the transient queue length in a queuing model with input dependent on the system state [J]. Journal of National University of Defense Technology, 2005, 27(1): 106-110. (in Chinese)
- [2] 俞政,刘再明,侯振挺. 一个具有相互独立、不同分布服务时间序列的排队模型[J]. 国防科技大学学报, 2002, 24(4): 80-85.
YU Zheng, LIU Zaiming, HOU Zhenting. A queuing model with mutually independent service times whose distributions are not necessarily identical [J]. Journal of National University of Defense Technology, 2002, 24(4): 80-85. (in Chinese)
- [3] Khazaei H, Mistic J, Mistic V B. Performance analysis of cloud computing centers using M/G/m/m + r queuing systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 23(5): 936-943.
- [4] Cao J W, Li K Q, Stojmenovic I. Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers [J]. IEEE

- Transactions on Computers, 2014, 63(1): 45-58.
- [5] Bai W H, Xi J Q, Zhu J X, et al. Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model [J]. Mathematical Problems in Engineering, 2015(1): 1-15.
- [6] Jin Y, He H C, Lu Y T. Ternary optical computer principle[J]. Science in China Series F: Information Sciences, 2003, 46(2): 145-150.
- [7] Jin Y, He H C, Lu Y T. Ternary optical computer architecture[J]. Physica Scripta, 2005, T118: 98-101.
- [8] Jin Y, Wang H J, Ouyang S, et al. Principles, structures and implementation of reconfigurable ternary optical processors[J]. Science China Information Sciences, 2011, 54(11): 2236-2246.
- [9] Peng J J, Shen R, Jin Y, et al. Design and implementation of modified signed-digit adder [J]. IEEE Transactions on Computers, 2014, 63(5): 1134-1143.
- [10] Jin Y, Shen Y F, Peng J J, et al. Principles and construction of MSD adder in ternary optical computer[J]. Science China Information Sciences, 2011, 53(11): 2159-2168.
- [11] Song K, Yan L P. Reconfigurable ternary optical processor based on row operation unit [J]. Optics Communications, 2015, 350: 6-12.
- [12] Yan J Y, Jin Y, Zuo K Z. Decrease-radix design principle for carrying/borrowing free multi-valued calculator and application in ternary optical computer [J]. Science in China Series F: Information Sciences, 2008, 51(10): 1415-1426.
- [13] Wang X C, Yao Y F, Wang C S, et al. Architecture of the monitor system in ternary optical computer [J]. Advanced Materials Research, 2013(616/617/618): 2158-2161.
- [14] Stewart W J. Probability, markov chains, queues and simulation: the mathematical basis of performance modeling [M]. USA: Princeton University Press, 2009: 559-610.
- [15] Gross D, Shortie J F, Thompson J M, et al. Fundamentals of queuing theory [M]. 4nd ed. New Jersey, USA: John Wiley & Sons, 2008.
- [16] Kleinrock L. Theory, volume 1, queuing systems [M]. New York, USA: Wiley-Interscience, 1975.