

گزارش فاز اول پروژه درس طراحی مدارهای دیجیتال کم‌توان

مهدی بهرامیان - ۴۰۱۱۷۱۵۹۳

عنوان مقاله : TMDS: Temperature aware Makespan Minimizing DAG Scheduler for Heterogeneous Distributed Systems

این مقاله یک الگوریتم زمانبندی مبتنی بر گراف تسک و با توجه به دما، روی پردازنده های ناهمگن ارائه میدهد. نکته تمایز این تحقیق از مقالات دیگر یکی همین ناهمگن بودن پردازنده هاست و دیگری توجه به دما و ویژگی های حرارتی پردازنده علاوه بر اهمیت زمان بر وظایف بیدرنگ است.

این مقاله با تعریف نحوه مدلسازی مساله شروع میکند :

- pow_t : توان مصرفی زمان t
- pow_d : توان مصرفی پویا زمان t
- pow_s : توان مصرفی ایستا زمان t
- $pow_t = pow_d + pow_s = C_0 f^3 + (C_1 f + C_2 f T_t)$
- $A = a(C_0 f^3 + C_1 f)$, $B = (b - aC_2 f)$
- $T_q = A/B + (T_0 - A/B)e^{(-B(q-0))}$: رابطه دما وقتی وظیفه بر روی پردازنده در حال اجراست

سپس هدف این بهینه سازی به صورت کمینه کردن طول زمان کل این زمانبندی به طوری که شرایط حرارتی حفظ شود تعریف می‌شود.

راهی که این مقاله برای این مساله ارائه میدهد به صورت ترکیب Γ تابع $TMDS(G, P)$ و $TETT(t, p, temp, T_est[t,p])$ است که اولی الگوریتم اصلی زمانبندی و دومی یک الگوریتم کمکی برای بررسی نتیجه زمانبندی وظیفه t روی پردازنده p در دمای $temp$ است.

این دو الگوریتم را در زیر مشاهده میکنید :

Algorithm 1: $TMDS(G, P)$

Input: Application DAG $G(V, E)$, processor set P

Output: A temperature-aware schedule that minimizes makespan

```
1  $\forall p_n \in P$ , Set  $avail[n] = 0$ ;  
2 Determine  $OFTs$ ,  $Ranks$  of all tasks using equations 17, 18;  
3 Initialize ready list  $taskList = \{\tau_{entry}\}$ ;  
4 // Let  $\Gamma_n^{cur}$  be a temporary variable which holds the temperature of processor  $p_n$  at any time during partial schedule generation  
5 while  $taskList \neq \phi$  do  
6    $\tau_j$  = Extract the task with highest  $Rank$  from ready list  $taskList$ ;  
7   for each processor  $p_n$  in  $P$  do  
8     Compute  $EST[\tau_j, p_n]$  using equation 19; // Start time of  $\tau_j$  if selected for execution on  $p_n$   
9      $\Gamma_{j,n}^{est} = T_{cool}(\Gamma^a, \Gamma_n^{cur}, EST[\tau_j, p_n] - avail[n])$ ; // Tentatively determine temperature of  $p_n$  at the instant  $EST[\tau_j, p_n]$  ( $\Gamma_{j,n}^{est}$ ), given: (i) its temperature at  $avail[n]$  ( $\Gamma_n^{cur}$ ), and (ii) that  $p_n$  idles in the interval  $[avail[n], EST[\tau_j, p_n]]$   $\langle EFT[\tau_j, p_n], Sch_{j,n}, \Gamma_n^{fin} \rangle = TETT(\tau_j, p_n, EST[\tau_j, p_n], \Gamma_{j,n}^{est})$ ; // The  $TETT()$  function returns: (i) effective finish time ( $EFT[\tau_j, p_n]$ ) of  $\tau_j$  on  $p_n$ , (ii) Schedule ( $Sch_{j,n}$ ) of execution and cooling intervals w.r.t. task  $\tau_j$  on  $p_n$  and (iii) temperature of  $p_n$  at  $EFT[\tau_j, p_n]$   
10     $SCT[\tau_j, p_n] = EFT[\tau_j, p_n] + OFT[\tau_j, p_n] - \omega_{j,n}$ ; // Estimate schedule completion time of  $\tau_j$  on  $p_n$   
11  Determine:  $p_{n'} = \min_{p_r \in P} SCT[\tau_j, p_r]$ ; // Find the processor  $p_{n'}$  for which  $SCT[\tau_j, p_{n'}]$  is minimum  
12  Assign  $\tau_j$  on processor  $p_{n'}$ ;  
13  Set  $AFT[\tau_j] = avail[n'] = EFT[\tau_j, p_{n'}]$ ,  $\Gamma_{n'}^{cur} = \Gamma_{n'}^{fin}$ ; // Update: (i) actual finish time of  $\tau_j$  ( $AFT[\tau_j]$ ), (ii)  $avail[n']$  and (iii) temperature  $\Gamma_{n'}^{cur}$  of  $p_{n'}$  at  $avail[n']$   
14  Insert all ready to execute successor tasks of  $\tau_j$  in  $taskList$ ;
```

		<i>TODS</i>	<i>TPSLs</i>	<i>TPPTS</i>	<i>TPEFT</i>	<i>THEFT</i>
<i>TMDS</i>	better	67.8%	66.7%	62.2%	54.0%	65.6%
	equal	1.3%	6.5%	1.3%	14.2%	1.3%
	worse	30.9%	26.8%	36.5%	31.8%	33.1%
<i>THEFT</i>	better	57.3%	43.7%	40.5%	39.3%	
	equal	1.5%	1.1%	1.7%	1.6%	
	worse	41.2%	55.2%	57.8%	59.1%	
<i>TPEFT</i>	better	64.1%	56.3%	54.9%		
	equal	2.1%	12.5%	1.8%		
	worse	33.8%	31.2%	43.3%		
<i>TPPTS</i>	better	60.3%	52.1%			
	equal	0.8%	1.4%			
	worse	38.9%	46.5%			
<i>TPSLs</i>	better	57.7%				
	equal	1.5%				
	worse	40.8%				