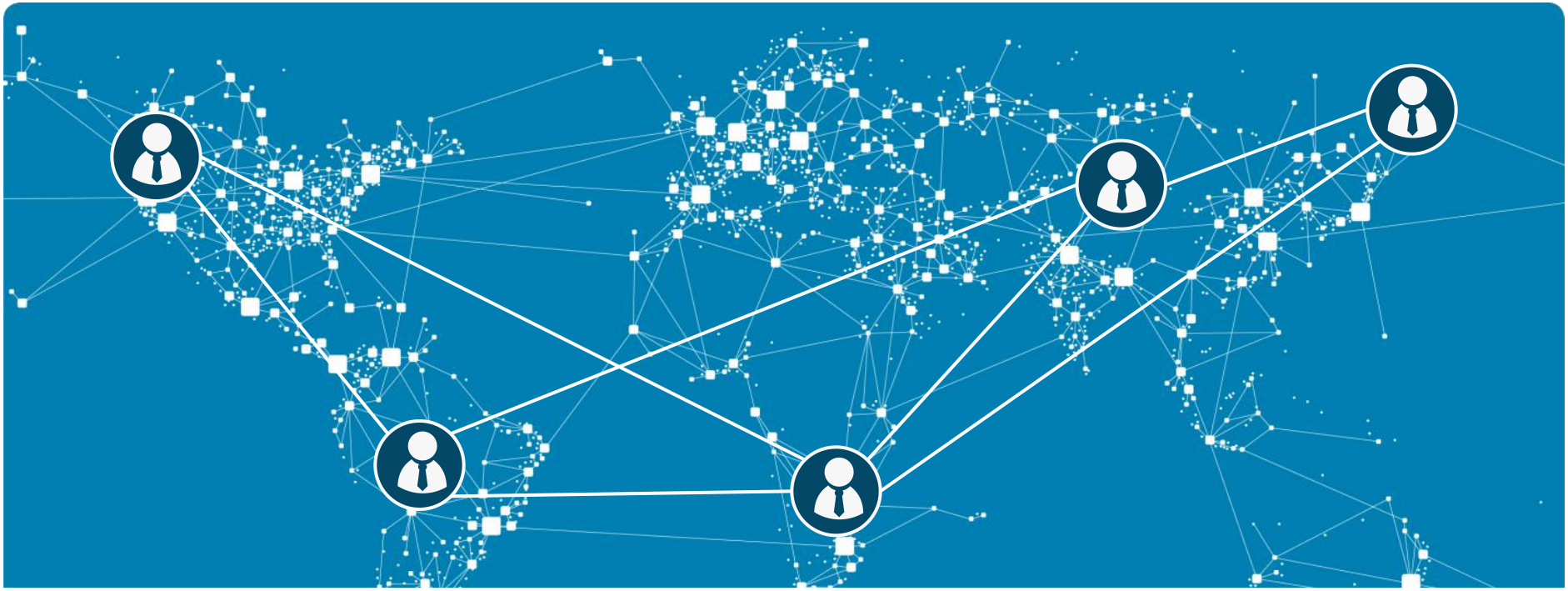


学期总结

宋金林　2017/7/13　导师：李仁发教授

学期工作总结

- CAN网络的时间分析
  - 单个CAN充分分析和精确分析
  - 网关互连CAN-CAN网络时间分析
- 异构分布式系统中能耗约束下的DAG调度算法研究
  - 改进现有算法，完成一篇小论文

CAN网络时间分析

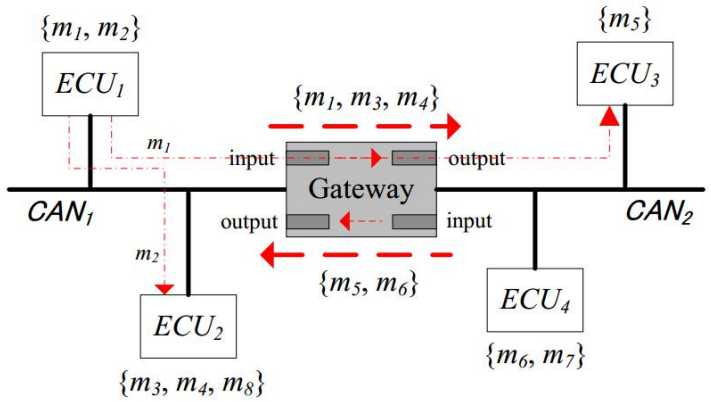
一个基本的模型如右图所示

工作：编程实现了

1、单个CAN网络充分分析和精确分析算法

2、网关互连CAN-CAN网络的时间分析算法

改进：未完成改进，即未能分析出更精确的WCRT。



[1] Davis R I, Kollmann S, Pollex V, et al. Controller Area Network (CAN) Schedulability Analysis with FIFO Queues[J]. Real-Time Systems, 2007, 35(3):239-272.

[2] X. Yong, Z. Gang, C. Yang, R. Kurachi, H. Takada, and L. Renfa, "Worst case response time analysis for messages in controller area network with gateway," IEICE TRANSACTIONS on Information and Systems, vol. 96, no. 7, pp. 1467–1477, Jul. 2013.

DAG能耗调度

- 问题描述：对于异构分布式系统中的并行应用，给定能耗($E_{given}(G)$)约束下，让调度长度($SL(G)$)最短。

$$E(G) = \sum_{i=1}^{|N|} E(n_i, u_{pr(i)}, f_{pr(i),hz(i)}) \leq E_{\text{given}}(G).$$

$$SL(G) = \min_{u_k \in U} \{ \min_{f_{k,h} \in [f_{k,\text{low}}, f_{k,\text{max}}]} \{EFT(n_{\text{exit}}, u_k, f_{k,h})\}\}.$$

- 该问题此前的研究是采用一种预分配的机制，给待分配的任务分配该任务在所有处理器上运行的最低能耗值，以保证最终的能耗不超过给定的总能耗约束。
- 缺陷：高优先级任务获得较多能量，低优先级任务获得较少能耗，在给定的总能量较少时，调度结果很悲观。（能量分配不均）

[1] X. Xiao, G. Xie, R. Li, and K. Li, "Minimizing schedule length of energy consumption constrained parallel applications on heterogeneous distributed systems," in Trustcom/BigDataSE/ISPA, 2016 IEEE. IEEE, 2016, pp. 1471–1476.

改进

- idea：均衡分配系统给定的能量，并且保证应用的总能耗不超过给定的约束值。

$$\begin{cases} \Delta E_{\mathrm{ae}}(G) = E_{\mathrm{given}}(G) - E_{\mathrm{min}}(G). \\ E_{\mathrm{aa}}(n_i) = E_{\mathrm{min}}(n_i) + \Delta E_{\mathrm{ae}}(G)/|N|. \\ E_{\mathrm{pre}}(n_i) = \min\{E_{\mathrm{aa}}(n_i), E_{\mathrm{max}}(n_i)\}. \end{cases}$$

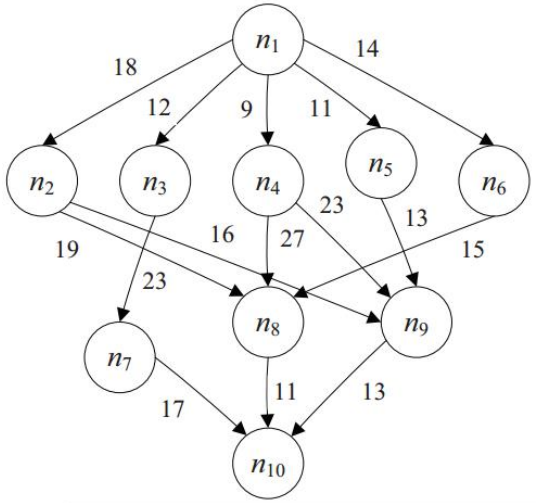
$$E_{s(j)}(G) = \sum_{i=1}^{j-1} E(n_{s(x)}, u_{pr(s(x))}, f_{pr(s(x)),hz(s(x))}) + E(n_{s(j)}, u_k, f_{k,h}) + \sum_{i=j+1}^{|N|} E_{\mathrm{min}}(n_{s(y)})$$

$$E_{s(j)}(G) = \sum_{i=1}^{j-1} E(n_{s(x)}, u_{pr(s(x))}, f_{pr(s(x)),hz(s(x))}) + E(n_{s(j)}, u_k, f_{k,h}) + \sum_{i=j+1}^{|N|} E_{\mathrm{pre}}(n_{s(y)})$$

应用数学归纳法证明了该预设值的可行性，即最终能耗不会超过给定的约束值。

实验结果对比（例子）

n_i	$E_{given}(n_i)$	$u_{pr(i)}$	$f_{pr(i),hz(i)}$	$E(n_i, pr(i), f_{pr(i),hz(i)})$	$AST(n_i)$	$AFT(n_i)$
n_1	13.44	u_3	1.0	9.63	0	12
n_3	20.33	u_3	1.0	20.33	9	28
n_4	18.19	u_2	1.0	6.72	18	26
n_2	19.26	u_1	1.0	10.79	27	40
n_5	10.92	u_3	1.0	10.7	28	38
n_6	13.44	u_2	1.0	13.44	26	42
n_9	5.4385	u_2	0.61	5.3606	56	75.67
n_7	1.3188	u_1	0.33	1.3177	51	72.2121
n_8	0.8874	u_1	0.26	0.8863	72.2121	91.4429
n_{10}	1.8204	u_2	0.26	1.8193	102.4429	129.3660
$E(G) = 80.98 \leqslant E_{\text{given}}(G) = 80.9939, SL(G) = AFT(n_{10})$=129.3660						



前面的任务以最大功耗运行来降低调度长度，后面的任务由于能耗不够，在最低功耗运行。（贪心）

(E=80.9939，SL=129.366)

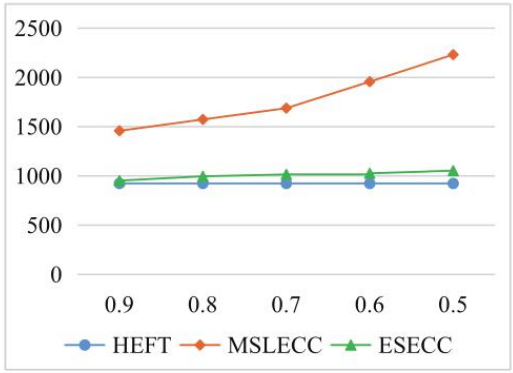
修改后的能耗分配更加均衡，结果也得到改善。

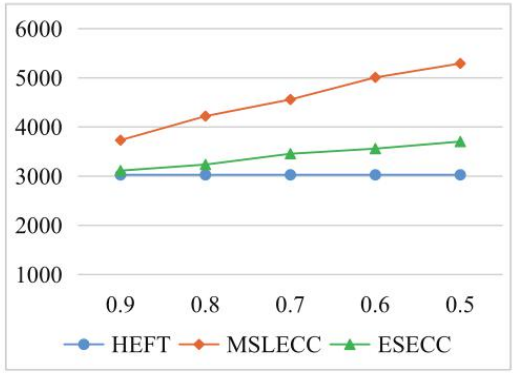
(E=74.6252，SL=84.0330)

ni	Egiven	upr	f	E	AST	AFT
n1	8.5499	3	0.91	**8.5051**	0.0000	9.8901
n3	8.0628	1	0.93	**8.0214**	21.8901	33.7181
n4	8.1888	2	1.00	**6.7200**	18.8901	26.8901
n2	9.8414	**3**	0.56	**9.7932**	9.8901	**42.0330**
n5	8.2436	2	0.81	**8.2236**	26.8901	42.9395
n6	8.3925	1	0.86	**8.2622**	33.7181	48.8344
n9	9.3174	2	0.94	**9.2597**	58.0330	70.7990
n7	7.3667	1	1.00	**5.8100**	48.8344	55.8344
n8	8.5112	**1**	1.00	**4.1500**	61.0332	**66.0332**
n10	12.2487	**2**	1.00	**5.8800**	77.0332	**84.0332**
			E=	74.6252	SL=	84.0332

FFT和GE实验

不同能耗约束（由多到少）下的调度长度

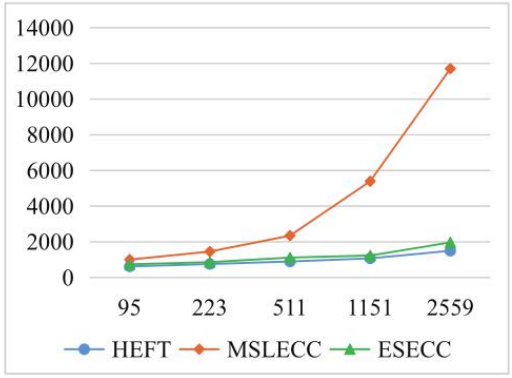


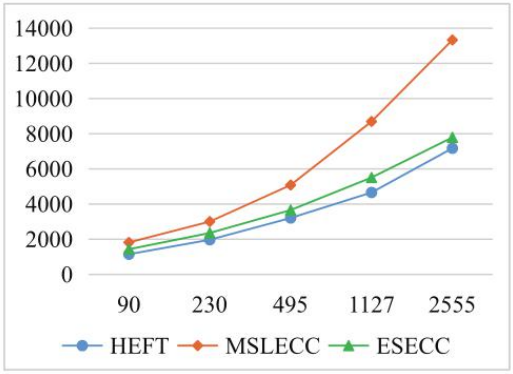


(a) Fast Fourier transform application. (b) Gaussian elimination application.

ESECC在任何情况下都比MSLECC要好。
特别地：在给定的能量比较少时，或者应用的规模比较大时，ESECC的优势更加明显。

不同应用规模（由小到大）下的调度长度





(a) Fast Fourier transform application. (b) Gaussian elimination application.

下一步计划

- 下阶段：找第二个创新点（DAG能耗调度相关）。

THANKS