



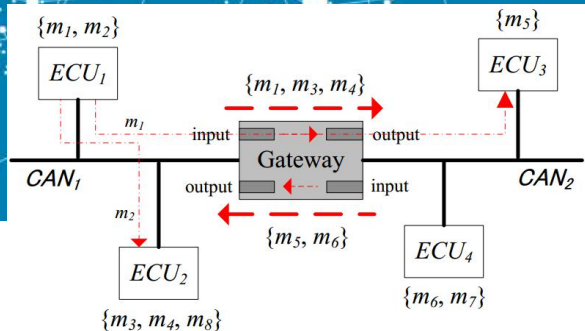
新一代汽车异构网络时间特性分析

宋金林 2017/7/4 导师：李仁发教授

当前进度

- 完成了CAN网络的时间分析（同构网络）
 - 充分分析和精确分析（基于忙周期理论）
 - 单个CAN以及网关互连CAN
- 异构分布式系统中能耗约束下的DAG调度算法研究
 - 完成一篇小论文（待投）

CAN网络时间分析



计算公式:

$$w_{s,i}^{n+1} = B_{s,i} + \sum_{j \in shp(i)} \left\lfloor \frac{w_{s,i}^n}{T_j} \right\rfloor \cdot C_j + \sum_{k \in dhp_{GW}(i)} \left\lfloor \frac{w_{s,i}^n + r_{s,k} - C_k - ADC_k}{T_k} \right\rfloor \cdot C_k$$

测试用例实验结果:

消息	不考虑间隔			考虑时间间隔		
	sou	des	e2e	sou	des	e2e
m1	4	4	8	4	4	8
m2	5	0	0	5	0	0
m3	7	6	13	7	6	13
m4	7	6	13	7	6	13
m5	7	8	15	7	8	15
m6	11	10	21	9	10	19
m7	16	0	0	12	0	0
m8	12	0	0	12	0	0

这个结果还有继续优化的空间（和真实的WCRT相比还是悲观了），但怎么优化是个难题。

DAG调度

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

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

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

- 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。

相关模型

○应用模型：DAG

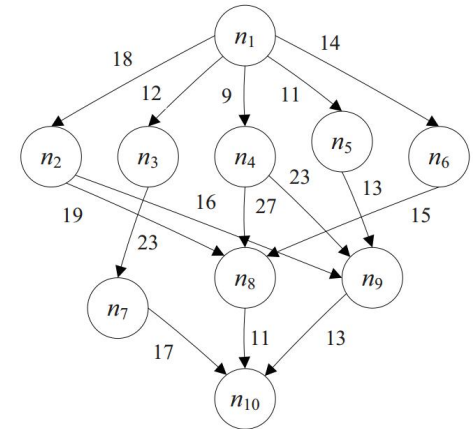
○任务优先级：向上排序值确定 (*Upward rank value*)

○能量模型：

$$P(f) = P_s + h(P_{\text{ind}} + P_d) = P_s + h(P_{\text{ind}} + C_{\text{ef}} f^m)$$

$$f_{\text{ee}} = \sqrt[m]{\frac{P_{\text{ind}}}{(m-1)C_{\text{ef}}}}$$

$$E(n_i, u_k, f_{k,h}) = (P_{k,\text{ind}} + C_{k,\text{ef}} \times f_{k,h}^{m_k}) \times \frac{w_{i,k} \times f_{k,\text{max}}}{f_{k,h}}$$



Task	u_1	u_2	u_3	$rank_u$
n_1	14	16	9	108.000
n_2	13	19	18	77.000
n_3	11	13	19	80.000
n_4	13	8	17	80.000
n_5	12	13	10	69.000
n_6	13	16	9	63.333
n_7	7	15	11	42.667
n_8	5	11	14	35.667
n_9	18	12	20	44.333
n_{10}	21	7	16	14.667

[1]G. Xie, J. Jiang, Y. Liu, R. Li, and K. Li, "Minimizing energy consumption of real-time parallel applications using downward and upward approaches on heterogeneous systems," *IEEE Transactions on Industrial Informatics*, 2017.

[2]B. Zhao, H. Aydin, and D. Zhu, "On maximizing reliability of real-time embedded applications under hard energy constraint," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 316–328, 2010.

[3]D. Zhu and H. Aydin, "Reliability-aware energy management for periodic real-time tasks," *IEEE Transactions on Computers*, vol. 58, no. 10, pp. 1382–1397, 2009.

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。

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

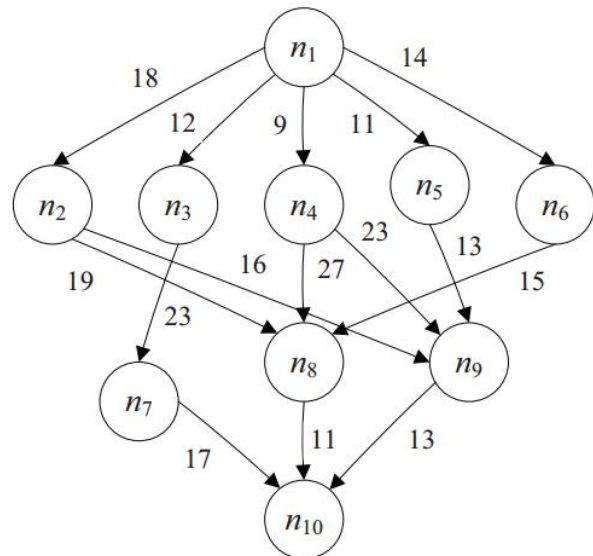
- 应用的能耗约束转化成任务的能耗约束

$$E_{\text{given}}(n_{s(j)}) = E_{\text{given}}(G) - \sum_{i=1}^{j-1} E(n_{s(i)}, u_{pr(s(i))}, f_{pr(s(i))}, hz(s(i))) - \sum_{i=j+1}^{|N|} E_{\text{pre}}(n_{s(i)}).$$

- 根据任务的最早结束时间 (EFT) 为任务选择相应的处理器进行调度, 实现最终的调度长度最小化。

实验结果对比 (例子)

n_i	$E_{given}(n_i)$	$u_{pr(i)}$	$f_{pr(i),n_i}(i)$	$E(n_i, pr(i), f_{pr(i),n_i}(i))$	$AST(n_i)$	$AFT(n_i)$
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
$E(G) = 80.98 \leq E_{given}(G) = 80.9939, SL(G) = AFT(n_{10})=129.3660$						



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。

(贪心算法)

(E=80.9939, SL=129.366)

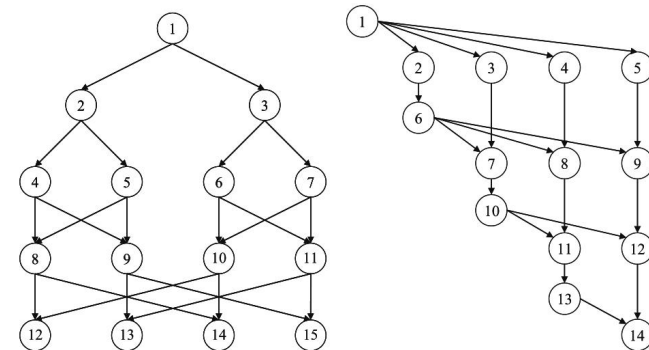
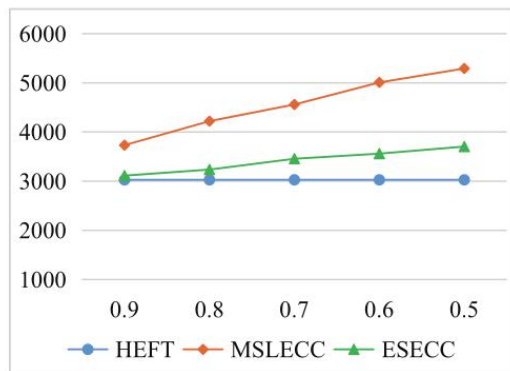
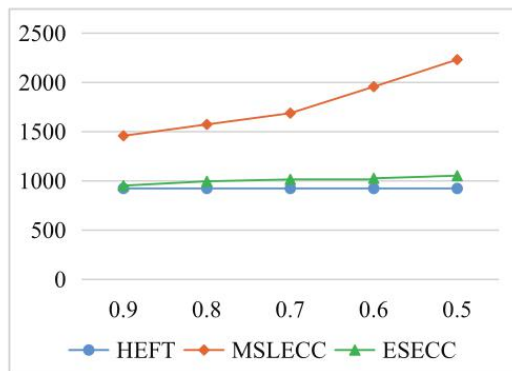
修改后的能耗分配更加均衡，结果也得到改善。

(E=74.6252, SL=84.0330)

n_i	E_{given}	upr	f	E	AST	AFT
n_1	8.5499	3	0.91	8.5051	0.0000	9.8901
n_3	8.0628	1	0.93	8.0214	21.8901	33.7181
n_4	8.1888	2	1.00	6.7200	18.8901	26.8901
n_2	9.8414	3	0.56	9.7932	9.8901	42.0330
n_5	8.2436	2	0.81	8.2236	26.8901	42.9395
n_6	8.3925	1	0.86	8.2622	33.7181	48.8344
n_9	9.3174	2	0.94	9.2597	58.0330	70.7990
n_7	7.3667	1	1.00	5.8100	48.8344	55.8344
n_8	8.5112	1	1.00	4.1500	61.0332	66.0332
n_{10}	12.2487	2	1.00	5.8800	77.0332	84.0332
				E=	74.6252	SL= 84.0332

FFT和GE实验

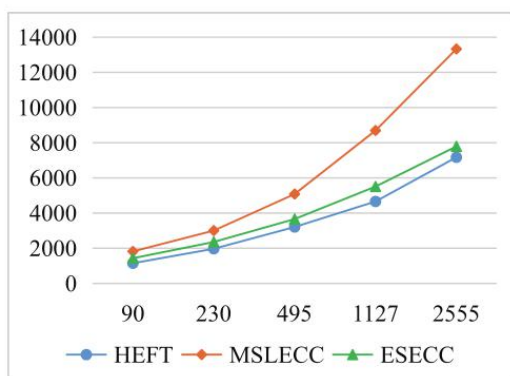
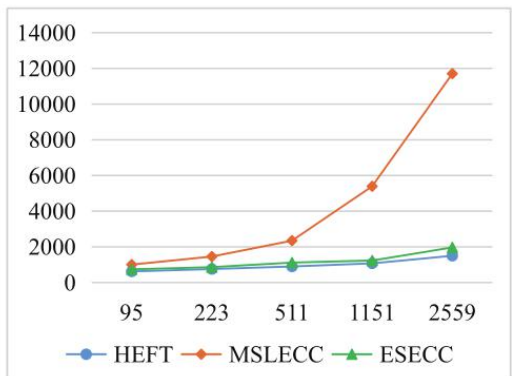
不同能耗约束下的调度长度



(a) Example of fast Fourier transform parallel applications with $\rho=4$. (b) Example of Gaussian elimination parallel applications with $\rho=5$.

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

不同应用规模下的调度长度



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

ESECC在任何情况下都比MSLECC要好。
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。

小结以及下一步计划

- 完成一篇小论文，待投。
- 下阶段找第二个创新点（还是能耗相关的DAG调度问题，考虑能耗约束下的费用最小化研究）。