سیزدهمین کنفرانس ملی انجمن کامپیوتر ایران
جزیره کیش، خلیج فارس، ایران ۱۹ الی ۲۱ اسفند ۱۳۸۶
انجمن کامپیوتر ایران
Computer Society of Iran
دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر
مقاله کامل

روش جدیدی برای زمانبندی اقتصادی گردش کارها در گریدهای محاسباتی

حمید محمدی فرد
دانشگاه آزاد اسلامی واحد مشهد
hamid_m_f@yahoo.com

حسین دلداری
دانشگاه فردوسی مشهد
hdeldari@yahoo.com

چکیده: گریدها به عنوان یک راه حل در کاربردهای با حجم بالای محاسبات مطرح می باشند. در رویکرد جدیدی از گرید که به رویکرد اقتصادی معروف می باشد، کاربران منابع بایستی هزینه استفده از منابع را به مالکین منابع پرداخت نمایند. در رویکردهای سنتی زمانبندی گرید، مدت زمان کامل شدن زمانبندی کارها به عنوان مهمترین پارامتر مد نظر قرار می گیرد در حالیکه در زمانبندهای اقتصادی بایستی علاوه بر پارامتر زمان اجرای کارها، هزینه استفاده از منابع نیز در نظر گرفته شود. در این مقاله الگوریتم جدیدی برای زمانبندی اقتصادی کارها در گریدهای محاسباتی پیشنهاد شده است که بر اساس یک رویکرد جدید ابتکار لیست، زمانبندی گردش کارهای مدل شده به صورت گرافهای بدون چرخه جهت دار (DAG) را انجام می دهد. در روش پیشنهادی بر اساس یک الگوریتم حریصانه، هزینه استفاده از منابع و مدت زمان تکمیل زمانبندی، در یک مرحله و بدون نیاز به زمانبندی مجدد بهینه سازی می شود.

واژه‌های کلیدی: زمانبندی گریدهای محاسباتی، زمانبندی اقتصادی، گردش کار،DAG

۱- مقدمه

در سالهای اخیر تکنولوژی گرید با مفهومی تحت عنوان گرید اقتصادی در حال تجربه ورود به عرصه تجارت می باشد. در گرید اقتصادی فراهم کنندگان منابع و سرویسها در قبال ارائه خدمات خود از مشتریانشان هزینه استفاده از سرویسها را دریافت نموده و مشتریان نیز برای استفاده از سرویسهای عرضه شده ملزم به پرداخت هزینه سرویسها می باشند [2].

در گریدها منابع تحت حوزه های مدیریتی مختلف با سیاستهای خاص هر حوزه، مدیریت می شوند. به دلیل گستردگی حوزه گرید از حیث جغرافیایی و تعدد منابع و سرویسها و همچنین فراهم کنندگان منابع، کاربران با شرایط متنوعی جهت انتخاب مواجه خواهند بود.

یک زمانبند در محیط گرید بایستی بدون در گیر کردن کاربر در پیچیدگی های ذاتی چنین محیطی با استفاده از اطلاعات موجود در GIS[1] ها و پارامترهای مد نظر کاربر یک زمانبندی کارآ با زمان و هزینه مناسب برای کاربر ارائه دهد.

فرق اصلی زمانبندهای سنتی و اقتصادی در این است که در زمانبندهای اقتصادی واگذاری منابع تنها بر اساس در دسترس بودن ظرفیت منابع صورت نمی گیرد بلکه به استطاعت و توانایی مالی مصرف کننده جهت استفاده از منابع نیز بستگی مستقیم دارد.

در گرید اقتصادی هر کدام از فراهم کنندگان و مصرف کنندگان منابع اهداف خاصی را دنبال می کنند که معمولا این اهداف در مقابل همدیگر قرار می گیرند. فراهم کنندگان منابع به دنبال واگذاری هر چه بیشتر منابع و کسب سود بیشتر و کاربران به دنبال به دست آوردن منابعی با سرعت اجرای بالاتر و هزینه پایین تر می باشند.

برای زمانبندی اقتصادی تا کنون روشهای کمی پیشنهاد شده است که تعدادی از آنها را می توان در [2, 3, 8, 12] مطالعه نمود در روشهای رایج از جمله الگوریتم های ارایه شده توسط Zhao [8] و Buyya [12] کاربر بایستی یکی از محدودیتهای زمان یا هزینه را مشخص نماید که این مطلب اگرچه از جهاتی مفید است ولی از این حیث که کاربر بدون آگاهی از وضعیت بازار قادر به ارائه محدوده های مناسبی نمی باشد ایجاد مشکل می کند. چنانچه در همه این الگوریتم ها در انتهای الگوریتم بخشی برای کنترل وجود یا عدم وجود جواب و زمانبندی مجدد در نظر گرفته شده است.

در روش پیشنهادی در این مقاله یک روش مبتنی بر DAG (که از روشهای رایج ساختار گراف کارها برای زمانبندی می باشد [1, 11, 7] [6]) برای زمانبندی گرید اقتصادی با رویکرد بهینه سازی توام زمان و هزینه پیشنهاد گردیده است که در این روش الگوریتم ارائه شده با یک رویکرد سودجویانه۲ در هر زمان سعی در پیدا کردن بهترین انتخاب از لحاظ هزینه و زمان اجرای کار می نماید.

در ادامه مطالب، در فصل دوم تئوری های مقدماتی و مفاهیم لازم برای زمانبندیهای مبتنی بر DAG بیان شده است. در فصل سوم به شرح جزئیات الگوریتم پیشنهادی پرداخته شده است. فصل چهارم به بررسی و ارزیابی و مقایسه روش پیشنهادی با چند الگوریتم دیگر اختصاص داده شده است و در نهایت در فصل پنجم نتیجه گیری کل مطالب خلاصه و بیان شده است.

۲- مقدمات وتعریف زمانبندی مبتنی بر DAG

DAG یک گراف $G\langle V,E\rangle$ می باشد که V مجموعه گره های DAG می باشد به طوری که $V=\{V_i; i=1..n\}$. هر گره در گراف نمایشگر یک کار می باشد. E مجموعه لبه های گراف بوده و نمایشگر وابستگی کارها در گراف می باشد و داریم $E=\{(V_i,V_j); i,j=1..n \;\&\; \exists! LinkforV_i,V_j\}$. هر لبه (V_i,V_j) در گراف به این معناست که به منظور اجرای V_j بایستی اجرای V_i به پایان رسیده باشد. درحقیقت این رابطه، یک رابطه با ترتیب جزئی می باشد. هر لبه (V_i,V_j) در DAG دارای وزنی معادل $D_{i,j}$ می باشد که نمایشگر زمان لازم برای انتقال داده هایی است که بایستی جهت اجرای V_j از V_i به آن انتقال یابد و همچنین لازم به ذکر است که قبل از عدم اتمام V_i این داده قابل انتقال به V_j نخواهد بود.

در هر DAG یک گره ورودی یا V_{entry} و یک گره خروجی یا V_{exit} وجود دارد که مسئله زمانبندی با تخصیص منبع به گره ورودی آغاز شده و با اتمام تخصیص منبع به گره خروجی و اجرای آن به پایان می رسد (این فرض خللی در کلیت مسئله وارد نمی کند، زیرا در صورت وجود بیش از یک گره ورودی یا خروجی، یک گره فرضی ورودی یا خروجی با لبه هایی با وزن صفر به DAG اضافه می شود).

زودترین زمان اجرای گره V_i روی ماشین M_j یا $EST(V_i,M_j)$، زودترین زمانی است که مقدمات اجرای V_i و داده های مورد نیاز آن برای M_j فراهم شده و کار می تواند در آن زمان شروع شود. زودترین زمان اتمام یا $EFT(V_i,M_j)$، زمانی است که اجرای V_i روی M_j به پایان خواهد رسید. بنا به توضیحات داده شده $EST(V_i,M_j)$ و $EFT(V_i,M_j)$ را برای هر گره می توان از روابط (۱) و (۲) محاسبه نمود.

$$EST(V_i,M_j)=\max\{Available[j],\max\{AFT(V_t)+D_{ij}; V_t\in \Pr ed(V_i)\}\} \quad (۱)$$

$$EFT(V_i,M_j)=T_{ij}+EST(V_i,M_j) \quad (۲)$$

برای گره ورودی خواهیم داشت : $EST(V_{entry},M_j)=0$.

در گرید مفروض، M مجموعه ماشینهای موجود جهت اجرای کارهای می باشد و داریم $M=\{M_j; j=1..m\}$. ماتریس $T_{m\times n}$ ماتریس زمان اجرای گره های گراف بر روی m ماشین موجود می باشد به طوری که $T_{i,j}$ زمان اجرای V_i روی M_j می باشد و ماتریس $C_{m\times n}$ ماتریس هزینه اجرای گره ها بر روی ماشینها می باشد به طوری که C_{ij} نشان دهنده هزینه اجرای V_i روی M_j می باشد.

$Available[j]$ زودترین زمانی است که ماشین M_j آماده اجرای کار می باشد و $\Pr ed(V_i)$ مجموعه گره های پیشین بلافاصله گره V_i می باشد. رابطه max موجود در معادله EST زمان آمادگی را برمی گرداند یعنی زمانی که همه داده ها ی مورد نیاز V_i به M_j رسیده است. بعد از اینکه V_t به ماشین M_j تخصیص داده شد، زودترین زمان شروع و زودترین زمان اتمام V_t روی ماشین M_j معادل با زمان شروع واقعی یا $AST(V_t)$ و زمان اتمام واقعی یا $AFT(V_t)$ مربوط به V_t خواهند بود. بعد از اتمام زمانبندی همه گره های گراف، $AFT(V_{exit})$ نشان دهنده پایان زمانبندی بوده و مجموع هزینه های تخصیص کارها به ماشینها برابر با هزینه کل زمانبندی خواهد بود. با تعاریف انجام شده توابع هدف زمانبندی روش پیشنهادی در (۳) و (۴) بیان شده است.

$$ScheduleLenght=\max\{AFT(V_{exit})\} \quad (۳)$$

$$ScheduleCost=\sum_{i=1}^{n}\sum_{j=\{1} C_{ij} \quad (۴)$$

۳- الگوریتم زمانبندی پیشنهادی با عنوان TCO (Time and Cost Optimization)

الگوریتم پیشنهادی TCO شامل سه مرحله می باشد. این مراحل به ترتیب، مرتب سازی سطوح، اولویت بندی کارها و در نهایت انتخاب ماشین می باشند. در ادامه جزئیات کامل هر مرحله تشریح شده است.

۳-۱- مرحله مرتب سازی سطوح

در این مرحله به منظور گروه بندی کارهای مستقل از هم، DAG در یک حالت بالا به پایین پیمایش می شود. کارهای گروه بندی شده در یک گروه قابلیت اجرای موازی با همدیگر را خوهند داشت.

در این مرحله روش کار به این صورت می باشد که در یک DAG داده شده $G\langle V,E\rangle$ سطح ۱ فقط شامل گره ورودی می باشد. سطح i شامل همه گره های V_k می باشد به طوری که به ازای همه لبه های (V_j,V_k) گره V_j در سطحی کمتر از i بوده و حداقل یک لبه (V_j,V_k) وجود داشته باشد که V_j در سطح i-1 باشد.

برای مثال برای گراف شکل (۱) ما ۵ سطح خواهیم داشت. سطح ۱ شامل گره ۱(گره ورودی)، سطح ۲ شامل گره های ۲،۳،۴ و سطح ۳ شامل گره های ۵،۶،۷،۸ و سطح ۴ شامل گره های ۹،۱۰ و سطح ۵ شامل گره ۱۱(گره خروجی).

۳-۲- مرحله اولویت بندی گره ها

در این مرحله اولویت اختصاص داده شده به هر گره بر اساس روش موجود در الگوریتم PETS [5] محاسبه می شود که شرح کامل این مرحله در ادامه بیان شده است.

برای تخصیص اولویت به یک گره سه پارامتر به نامهای ACC [۳] (میانگین زمان محاسبات) ، DTC [۴] (زمان انتقال داده)، RPT [۵] (درجه گره های پیشین) تعریف می شود. ACC یک گره میانگین زمان محاسبات گره روی m ماشین بوده و از رابطه (۵) محاسبه می گردد.

$$ACC(V_i) = \sum_{j=1}^{m} T_{ij} / m \qquad (۵)$$

DTC یک گره V_i مقدار زمان ارتباطات مربوط به انتقال داده V_i به همه گره های بعدی بلافاصله اش بوده و در سطح ۱ بر اساس معادله (۶) محاسبه می گردد.

$$DTC(V_i) = \sum_{j=1}^{n} C_{ij}; i < j \qquad (۶)$$

در این معادله n تعداد گره های سطح بعد می باشد و برای گره خروجی $DTC = 0$ می باشد.

RPT یک گره V_i بالاترین درجه همه گره های پیشین بلافاصله بوده و از رابطه (۷) محاسبه می گردد.

$$PRT(V_i) = \max\{rank(V_1) + rank(V_2) + ... + rank(V_h)\} \qquad (۷)$$

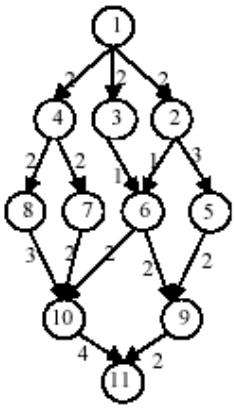
در رابطه ۷ ، $V_1, V_2, ..., V_h$ گره های پیشین بلافاصله V_i بوده و برای گره ورودی $RPT = 0$ می باشد. درجه برای هر گره V_i بر اساس مقادیر ACC و DTC و RPT محاسبه می گردد و ما حداکثر درجه گره های پیشین V_i به عنوان یکی از پارامترهای محاسبه درجه V_i به کار برده می شود که از رابطه (۸) به دست می آید.

$$rank(V_i) = round(ACC(V_i) + DTC(V_i) + RPT(V_i)) \qquad (۸)$$

همه گره های در سطح ۱ بر اساس مقدار درجه محاسبه شده اولویت بندی می شوند. در هر سطح گره با با لاترین درجه بالاترین اولویت را خواهد داشت و به همین ترتیب بقیه گره های هر سطح مرتب می شوند و اگر دو گره درجه های مساوی داشته باشند گره با ACC کمتر اولویت بیشتری خواهد داشت و برای مثال برای گراف شکل (۱) مقادیر ACC ، DTC ، RPT و درجه اولویت در ادامه محاسبه شده اند.

جدول (۱): ماتریس زمان ارتباطات (T) داده شده گراف شکل (۱)

Task V_i	M_1	M_2	M_3
1	4	4	4
2	5	5	5
3	4	6	4
4	3	3	3
5	3	5	3
6	3	7	2
7	5	8	5
8	2	4	5
9	5	6	7
10	3	7	5
11	5	6	7



شکل (۱): یک DAG نمونه

جدول (۲): محاسبه پارامترهای تعریف شده برای گراف شکل (۱)

Level	Task	DTC	ACC	RPT	rank	Priority
1	1	6	4	0	10	1
2	2	4	5	10	19	1
2	3	1	5.25	10	16	3
2	4	4	3	10	17	2
3	5	2	3.75	19	25	2
3	6	4	3.5	19	27	1
3	7	2	5.75	17	25	3
3	8	3	3.5	17	24	4
4	9	2	5.75	26.5	34	2
4	10	4	4.25	26.5	35	1
5	11	0	6.5	34.75	41	1

برای گره V_1 سه گره بعدی بلافاصله V_2, V_3, V_4 وجود دارد و زمان ارتباطات میان V_1 و این گره ها ۲،۲،۲ می باشد. در اینجا DTC گره ۱ برابر ۶ (۲+۲+۲) می باشد ومقدار RPT گره V_1 صفر می باشد. مقدار ACC گره V_1، ۴ و مقدار درجه گره V_1، ۱۰ (۴+۶+۰) می باشد. اولویت گره V_1، ۱ می باشد. به طریق مشابه پارامترهای بقیه گره ها در سطوح مختلف محاسبه می شوند این مقادیر در جدول (۲) نمایش داده شده است.

جدول (۳): هزینه ماشینها برای گراف (۱)

M_j	$Cost_j$
1	180$
2	120$
3	100$

۳-۳- مرحله انتخاب ماشین

این مرحله به عنوان مرحله اصلی و تکنیک اصلی روش پیشنهادی می باشد که به تفصیل در ادامه تشریح می گردد. درمرحله انتخاب ماشین هدف انتخاب ماشینهایی برای اجرای کارها می باشد که هم از حیث زمان و هم از حیث هزینه مقدار مناسبی را ارائه دهند.

بدین منظور ابتدا پارامتری با عنوان متوسط هزینه هر واحد زمانی استفاده از کل ماشینها از رابطه (۹) محاسبه می گردد.

$$C_{avg} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} T_{ij}} \qquad (۹)$$

سپس بر اساس اولویتهای مشخص شده در مرحله قبلی، EST و EFT هر گره V_i بر روی ماشینهای موجود $M_1, M_2, ...M_m$ محاسبه می شود. فرض می شود برای هر گره V_i کمترین مقدار مربوط به EFT بر روی ماشین با شماره a اتفاق بیفتد، بر این اساس برای گره V_i اختلاف زمان اجرا بر روی ماشینهای M_j و M_a از رابطه (۱۰) به دست می آید.

$$EFT_Diff(i,j) = EFT(i,j) - EFT(i,a) \qquad (۱۰)$$

و تفاوت هزینه اجرای روی ماشین M_j و M_a از رابطه (۱۱) به دست می آید.

$$Cost_Diff(i,j) = C_{i,a} - C_{i,j} \qquad (۱۱)$$

حال بر اساس اختلاف موجود در هزینه ها و زودترین زمان پایان بر روی ماشین های مختلف بر اساس رویکرد زیر اقدام به انتخاب بهترین گزینه می شود.

ماشینهایی که مقدار Cost_Diff آنها منفی باشد انتخاب نامناسبی بوده و کنار گذاشته می شوند، زیرا این حالت به این معناست که در ماشین j با صرف هزینه بیشتر کار مربوطه نیز دیرتر به پایان رسیده است، که این حالت به طور مشخص غیر بهینه است. در ماشینهای باقیمانده یعنی آنهایی که Cost_Diff آنها مثبت است با مقایسه ذیل بررسی می شود که کدام ماشین گزینه مناسبتری جهت انتخاب خواهد بود.

بدین منظور تفاوت هزینه ای که انتخاب a و انتخاب j در اجرای V_i دارد محاسبه نموده و با مقایسه با متوسط ضرری که از انتخاب a به عوض j حاصل می شود، مشخص می شود که کدام ماشین انتخاب مناسب تری می باشد به طور خلاصه می توان گفت ماشینی که کمترین مقدار $Main_Diff$ را بر اساس رابطه (۱۲) بدهد انتخاب می گردد.

$$Main_Diff = [C_{i,j} - C_{i,a}] - [EFT(i,j) - EFT(i,a)] * C_{avg} \qquad (۱۲)$$

جهت خلاصه سازی، تابعی که بر اساس روش بالا ماشین مناسب را انتخاب می کند $MachineSelect(V_i)$ نامگذاری می شود که این تابع مناسب ترین ماشین را برای تخصیص هر گره V_i مشخص می کند.

مراحل این زمانبندی در قالب الگوریتم زیر تنظیم و ارائه شده است.

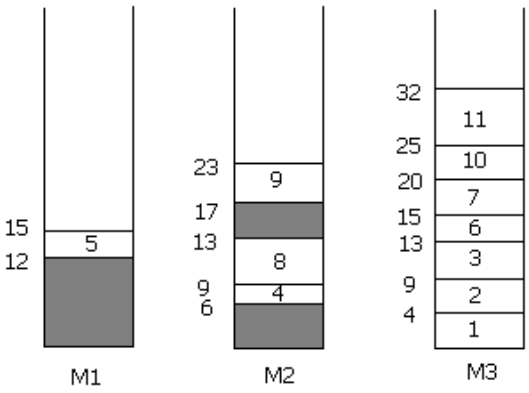
***Step 1**: read the DAG G<V,E>, and M,C,T,D Matrixes;*
***Step 2**:Leveling the DAG;*
***Step 3:** for* all tasks V_k at each level *Li* do
begin
compute *ACC(V_k),DTC(V_k) and RPT* (V_k);
rank(V_k) = ACC(V_k) + DTC(V_k) + RPT (V_k);
//tie, if any, is broken based on *ACC* value, the task with minimum *ACC* value receives the higher priority followed by the task with //next minimum *ACC* value and so on;
end;
Step 4: construct a priority queue using ranks;
***Step 5:** while* there are unscheduled task V_k in the queue *do*
MachineSelection(V_k)

مراحل انتخاب ماشین برای گراف شکل (۱) در جدول (۴) نمایش داده شده است.

جدول (۴): محاسبه پارامترهای الگویتم برای گراف شکل (۱)

	M1		M2		M3		
Task	EST	EFT	EST	EFT	EST	EFT	Selected Processor
1	0	4	0	4	0	4	3
2	6	11	6	11	4	9	3
4	6	9	6	9	9	12	2
3	6	10	9	15	9	13	3
6	14	17	14	21	13	15	3
5	12	15	12	17	15	18	1
7	15	20	9	17	15	20	3
8	15	17	9	13	20	25	2
10	22	25	22	29	20	25	3
9	17	22	17	23	25	32	2
11	29	34	29	35	25	32	3

بر اساس مقادیر محاسبه شده در جدول (۴) زمانبندی کارهای گراف شکل (۱) به ۳ ماشین موجود در شکل (۲) مشاهده می شود.



شکل (۲): انتخاب ماشینها به وسیله الگوریتم TCO

٤- ارزیابی الگوریتم

الگوریتم پیشنهادی این مقاله در حقیقت یک الگوریتم سودجویانه است که در هر لحظه اقدام به انتخاب مناسب ترین ماشین از حیث زمان اجرا و هزینه اجرا برای هر کار می نماید که در نهایت منجر به زمانبندی مناسبی برای زمان و هزینه می گردد.

برای شبیه سازی الگوریتم زمانبند TCO و همچنین محیط گرید از شبیه ساز GridSim [4,9] استفاده شده است. محیط گرید شبیه سازی شده شامل چندین موجودیت کاربر و منبع با نیازمندیهای مختلف می باشد. کاربران شبیه سازی شده شامل ویژگیهای کاربرد و نیازمندیهای کیفیت سرویس(با استفاده از Gridlet) می باشد. به منظور شبیه سازی دو موجودیت کاربر و دلال، کلاس GridSim توسعه داده شده است.

در آزمایش مربوط به این تحقیق الگوریتم پیشنهادی TCO با دو الگوریتم معروف دیگر به نامهای HEFT [10] که یک الگوریتم کلاسیک در زمانبندی گرید می باشد و الگوریتم LOSS [17] که یک الگوریتم زمانبندی اقتصادی از دسته الگوریتم های DBC می باشد، مقایسه شده است.

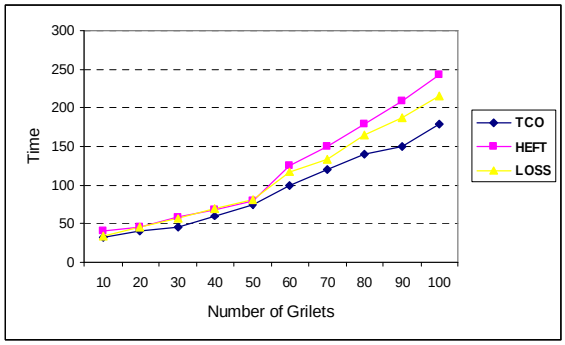
در آزمایشات صورت گرفته از توسعه کلاس GridSim برای دو موجودیت کاربر و دلال استفاده شده است. برای شبیه سازی منابع تعداد ۱۵ ماشین با محدوده سرعت ۱۰۰۰ تا ۴۰۰۰ MIPS استفاده شده است. برای مدلسازی کاربردهای کاربران فرض شده که ۱۰۰ وظیفه وجود دارد. هر کدام از این وظایف توسط یک نمونه از کلاس Gridlet تعریف شده اند. هر Gridlet شامل اطلاعات هر وظیفه از جمله طول وظیفه بر حسب MI و اندازه ورودی و خروجی بر حسب بایت می باشند. طول وظایف بر حسب زمان اجرای آنها روی یک PE استاندارد در نظر گرفته شده است. طول اجرای Gridlet ها بر اساس یک انتخاب تصادفی با حداقل ۱۰۰۰۰ MI و یک نرخ رشد ۱۰٪ انتخاب شده است.

جهت بررسی الگوریتم یک سری شبیه سازی بر اساس تغییر در تعداد کاربران، تعداد منابع و با در نظر گرفتن پخش بودن منابع در حوزه های جغرافیایی و مدیریتی مجزا استفاده شده است.

همچنین در این مدلسازی فرض شده است که هر ماشین یک PE دارد. شبیه سازی ۲۰۰ بار تکرار شده است و نتایج در نمودارهای موجود ارائه شده است. شبکه ارتباطی بین منابع از نوع mesh در نظر گرفته شده است. این توپولوژی به این دلیل استفاده شده است که شبیه سازی درگیر ساختار پیچیده شبکه نشود. پهنای باند شبکه از MB۱۶ تا MB۱۰۲۴ با نسبت افزایش MB۶۴ در نظر گرفته شده است.

در این آزمایش فرض بر این قرار گرفته که هزینه تابع خطی از سرعت می باشد که به صورت $Cost = K \times Speed$ می باشد که مقدار k به صورت تصادفی در هر مرحله انتخاب گردیده است.

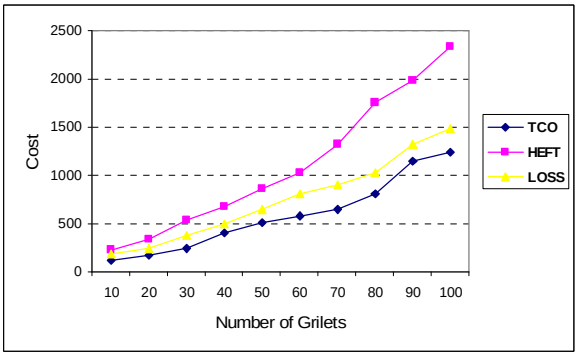
در اولین آزمایش makespan سه الگوریتم TCO، HEFT و LOSS مقایسه شده اند.



شکل (۳): نمودار مقایسه Makespan سه الگوریتم TCO، HEFT و LOSS

مشاهده می شود که زمان اجرای عملیات زمانبندی در الگوریتم TCO از هر دو الگوریتم HEFT و LOSS بهتر می باشد. دلیل این رخداد را می توان با توجه به اینکه در هر دو مرحله اول و دوم یعنی رتبه بندی منابع و انتخاب منابع، زمان اجرا در نظر گرفته می شود توجیه نمود.

در آزمایش بعدی هزینه مصرفی سه روش مقایسه شده اند که مشاهده می شود که الگوریتم TCO نتایجی به مراتب بهتر ارائه داده است.



شکل (۴): نمودار مقایسه هزینه زمانبندی در الگوریتم های TCO، HEFT و LOSS

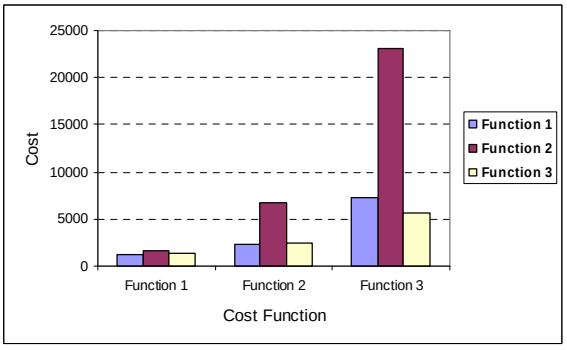
در آزماش بعدی با توجه به این مطلب که هزینه استفاده از ماشینها در نتایج زمانبندی تاثیر مستقیم دارد، سه رابطه بین هزینه و سرعت ماشینها لحاظ شده است. این سه تابع عبارتند از:

$$Cost = \sqrt{Speed} \quad \text{(تابع اول)}$$

$$Cost = Speed^2 \quad \text{(تابع دوم)}$$

$$Cost = e^{speed} \quad \text{(تابع سوم)}$$

همانطور که در روابط مشخص است فرض بر این بوده که هزینه استفاده از منابع به صورت یک تابع صعودی بر حسب افزایش سرعت ماشینها می باشد. که این فرض خللی به کارآیی مسئله وارد نمی کند، زیرا در دنیای واقعی نیز معمولا ماشینهای با سرعت بیشتر هزینه استفاده بیشتری نیز دارند. ولی نسبت نرخ رشد هر تابع هزینه بر حسب زمان بسیار متفاوت بوده به صورتیکه در تابع سوم این رشد به صورت نمایی فرض شده است. آزمایش بر اساس هر کدام از این توابع هزینه در سه رویکرد LOSS، HEFT و TCO مورد تست و بررسی قرار گرفتند که نتایج در نمودار شکل (۵) قابل رویت می باشد.



شکل (۵): نمودار مقایسه makespan زمانبندی در الگوریتم های TCO، HEFT و LOSS بر اساس سه تابع هزینه پیشنهادی

۵- نتیجه و کارهای آینده

رویکرد اقتصادی زمانبندی کارها در محیط گرید علاوه بر پارامتر مدت زمان زمانبندی بایستی هزینه استفاده از منابع را نیز در فرآیند زمانبندی در نظر داشته باشد. در اغلب الگوریتم های زمانبندی اقتصادی، کاربر بایستی یکی از محدودیتهای زمان یا هزینه را برای زمانبند مشخص نماید که این مطلب اگرچه از جهاتی مفید است ولی از این حیث که کاربر بدون آگاهی از وضعیت بازار قادر به ارائه محدوده مناسبی نمی باشد ایجاد مشکل می کند. در این مقاله یک الگوریتم با رویکرد ابتکار لیست[۶] به نام TCO برای زمانبندی اقتصادی مبتنی بر DAG در محیط گرید پیشنهاد شده است. هدف این الگوریتم بهینه سازی توام زمان و هزینه هم چنین دور کردن کاربر از شرایط مربوط به محیط گرید می باشد و انتخاب بهترین منبع به زمانبند واگذار گردیده است.

به منظور کار آینده در ارتباط با این روش پیشنهادی می توان مسئله پویایی در محیط گرید (از قبیل بروز خطا در منابع تخصیص یافته) را به الگوریتم اضافه نمود.

مراجع

[1] J. Blythe, S. Jain, E. Deelman, Y. Gil, K. Vahi, A. Mandal, and K. Kennedy. *Resource Allocation Strategies forWorkflows in Grids* In IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005).

[2] R. Buyya. *Economic-based Distributed Resource Management and Scheduling for Grid Computing*. PhD thesis, Monash University, Melbourne, Australia, http://www.buyya.com/thesis, April 12 2002.

[3] R. Buyya, D. Abramson, and J. Giddy. *An economy grid architecture for service-oriented grid computing*. In 10th IEEE Heterogeneous ComputingWorkshop (HCW'01), San Fransisco, 2001.

[4] R. Buyya and M. Murshed, "GridSim: *A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing*" Concurrency and Computation: Practice and Experience, 14(13-15):1175-1220, Wiley Press, USA, 2002.

[5] E. Ilavarasan and P. Thambidurai. *Low Complexity Performance Effective Task Scheduling Algorithm for Heterogeneous Computing Environments*; Journal of Computer Sciences 3 (2): 94-103, 2007

[6] A. Mandal, K. Kennedy, C. Koelbel, G. Marin, J. Mellor-Crummey, B. Liu and L. Johnsson. *Scheduling Strategies for Mapping Application Workflows onto the Grid*. In IEEE International Symposium on High Performance Distributed Computing (HPDC 2005), 2005.

[7] R. Sakellariou and H. Zhao. *A low-cost rescheduling policy for efficient mapping of workflows on grid systems*. In Scientific Programming, volume 12(4), pages 253–262, December 2004.

[8] R. Sakellariou, H. Zhao, E. Tsiakkouri, M. D. Dikaiakos. "*Scheduling Workflows with Budget Constraints*". In S.Gorlatch, M.Danelutto (Eds.), Integrated Research in Grid Computing, CoreGrid series, Springer-Verlag, to appear,2005

[9] A. Sulistio and R. Buyya, "*A Grid Simulation Infrastructure Supporting Advance Reservation*", In 16th International Conference on Parallel and Distributed Computing and Systems (PDCS 2004), ACTA Press, Anaheim, California, November 9-11, 2004, MIT Cambridge, Boston, USA.

[10] H. Topcuoglu, S. Hariri, and M. Wu. *Performance-effective and low-complexity task scheduling for heterogeneous computing*. In IEEE Transactions on Parallel and Distributed Systems, volume 13(3), pages 260–274, March 2002.

[11] M. Wieczorek, R. Prodan and T. Fahringer. *Scheduling of Scientific Workflows in the ASKALON Grid Environment*. In SIGMOD Record, volume 34(3), September 2005.

[12] J. Yu,R. Buyya, C. Khong Tham . *Cost-based scheduling of scientific workflow applications on utility grids* , Vic., Australia;,e-Science and Grid Computing, 2005. First International Conference on, Dec. 2005.

زیرنویس‌ها

[1] Grid Information Server
[2] Greedy
[3] Average Computation Cost
[4] Data Transfer Cost
[5] Rank of Predecessor Tasks
[6] List Heuristic