

# SID



سرویس های ویژه



سرویس ترجمه تخصصی



کارگاه های آموزشی



بلاگ مرکز اطلاعات علمی



عضویت در خبرنامه



فیلم های آموزشی

## کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی



مباحث پیشرفته یادگیری عمیق؛ شبکه های توجه گرافی (GAN)

مباحث پیشرفته یادگیری عمیق؛  
شبکه های توجه گرافی  
(Graph Attention Networks)



آموزش استفاده از وب آو ساینس

کارگاه آنلاین آموزش استفاده از  
وب آو ساینس



کارگاه آنلاین مکالمه روزمره انگلیسی

## مدل سازی توزیع وظایف و محاسبه قابلیت اطمینان در سرویس های گرید دارای توپولوژی ستاره با استفاده از شبکه های پتری رنگی زمانی

رضا انتظاری ملکی

دانشجوی کارشناسی ارشد مهندسی نرم افزار دانشگاه علم و صنعت ایران

r\_entezari@comp.iust.ac.ir

محمد عبداللهی ازگمی

استادیار و عضو هیئت علمی دانشکده کامپیوتر دانشگاه علم و صنعت ایران

azgomi@iust.ac.ir

است. اشتراک منابع در گرید، توسط سیستم مدیریت منابع (Resource Management System) کنترل می شود. درخواستی را که کاربر به RMS تحویل می دهد، وظیفه (Task) می نامند. RMS باید وظیفه های تحویل گرفته شده را به زیروظایف (Subtask) تقسیم نموده و آن را برای حل به منابع موجود در شبکه بفرستد [3].

هدف اساسی در گرید محاسباتی، استفاده از قدرت پردازش کامپیوترهای موجود در شبکه برای حل مسائل پیچیده و زمان بر است. برای رسیدن به هدف گرید و حداکثر استفاده از منابع موجود در محیط گرید، نحوه توزیع زیروظایف بین منابع و زمان بندی آنها از اهمیت ویژه ای برخوردار است. این زمان بندی با در نظر گرفتن کیفیت سرویس (Quality of Service) انجام می شود و سعی بر این است که در حین زمان بندی، زیروظایف به گونه ای بین منابع پخش شوند که حداکثر QoS حاصل گردد [4,5,6]. به عنوان یکی از معیارهای مهم QoS در محیط گرید، قابلیت اطمینان (Reliability) در سرویس های گرید، باید مطالعه شود. چرا که مطالعه آن می تواند به طراحان سیستم ها در جهت تولید سیستم های بهتر و با قابلیت اطمینان بیشتر و به کاربران در جهت استفاده ای معقول و منطقی از محیط گرید کمک کند [7].

زمان اجرای (Execution time) یک وظیفه به عنوان عامل مورد نظر برای محاسبه قابلیت اطمینان در محیط گرید در نظر گرفته می شود [8,9,10,11] و همیشه سعی بر این است که یک وظیفه در زمانی کمتر از زمان تعیین شده برای آن، اجرا شود. معمولاً، تکنیکی که در محیط گرید برای افزایش قابلیت اطمینان استفاده می شود، تکنیک افزونگی است. به این ترتیب که RMS پس از شکستن وظیفه به زیروظایف، بجای اینکه هر وظیفه را به یک منبع منتسب کند، آن را به چندین منبع می دهد تا همه ای آنها بصورت همزمان اقدام به پردازش زیروظیفه ای مربوطه بکنند و اگر یکی از منابع طی انجام آن دچار خرابی شد، منابع دیگر بتوانند خرابی را جبران کنند.

روش های زمان بندی مختلفی با در نظر گرفتن قابلیت اطمینان ارائه شده اند که هدف آن ها برقراری حد تعادل بین اجرای وظایف در کمترین زمان ممکن و با حداکثر قابلیت اطمینان است. ولی مدل سازی این زمان بندی و توزیع زیروظایف در محیط گرید به همراه افزایش قابلیت اطمینان، خیلی مورد توجه قرار نگرفته و صرفاً چند مدل ساده

**چکیده:** گرید محاسباتی تکنولوژی جدیدی است که با هدف به اشتراک گذاشتن منابع و همکاری در سطح وسیع پدید آمده است. زمان بندی وظایف برای رسیدن به سطح کیفیت مطلوب، از جمله زمینه های مهم و مطرح در محیط گرید است. هر چند کارهای زیادی در مورد نحوه زمان بندی و توزیع زیروظایف در محیط گرید و با هدف افزایش قابلیت اطمینان و کارایی آن انجام شده، ولی در مورد ارائه تعریف صوری کارهای زیادی صورت نگرفته و کارهای موجود هر کدام به جنبه ای خاصی از توزیع زیروظایف اشاره کرده یا در مورد مثالی خاص ارائه شده اند. ما، در این مقاله، قابلیت اطمینان در سرویس های گرید را بررسی کرده و با استفاده از شبکه های پتری رنگی، مدلی برای محاسبه آن ارائه نموده ایم.

در این مطالعه اساس کار گرید بر عهده سیستم مدیریت منابع بوده و این سیستم وظایف را از کاربران گرفته و آنها را به زیروظایف تقسیم می کند، سپس هر زیروظیفه را به یک یا چندین منبع در دسترس فرستاده و در نهایت پس از اجرا شدن آنها توسط منابع متصل، خروجی ها را جمع کرده و به وظیفه ای درخواستی کاربر پاسخ می دهد. عملیات فوق با استفاده از شبکه های پتری رنگی زمانی شبیه سازی شده و محاسبات قابلیت اطمینان برای آن انجام گرفته است.

**واژه های کلیدی:** گرید محاسباتی، سیستم مدیریت منابع، توزیع زیروظایف، قابلیت اطمینان، زمان اجرا، شبکه های پتری رنگی زمانی.

### ۱- مقدمه

گرید محاسباتی، فراساختار سخت افزاری و نرم افزاری است که دسترسی ارزان، فراگیر و پایدار به منابع محاسباتی موجود در شبکه را فراهم می آورد [1]. در تعاریفی که بعداً و توسط افراد مختلف برای گرید ارائه گردید، مسئله اشتراک منابع و حل مسائل در گرید نیز مد نظر قرار گرفت [1,2]. مهمترین مسأله ای که در اینجا مطرح است، مسأله ای به اشتراک گذاشتن منابع است. البته باید توجه داشت که منظور از اشتراک منابع، مبادله مستقیم فایل نیست بلکه امکان دسترسی آسان به کامپیوترهای موجود در شبکه و استفاده از قدرت محاسباتی و سایر امکاناتی که آنها خود مایل به اشتراک گذاشتن آن هستند، مد نظر

صوری و سطح بالاتری ارائه کرده‌اند و در نهایت محاسبات قابلیت اطمینان را برای آن انجام داده‌اند. Han در [17] مدل‌سازی خود را براساس یک وظیفه‌ی خاص، که در ابتدای مقاله توضیح داده، انجام می‌دهد و شبکه‌ی پتری حاصل جامعیت لازم را ندارد، ولی نکته قابل توجه و مهم در این مقاله در نظر گرفتن محدودیت‌های اولویتی و وابستگی داده‌ای بین زیروظایف است که در مورد همان مثال خاص در نظر گرفته شده است. در این مقاله در نهایت تحلیلی ساده از بدست آوردن کمترین زمان اجرا در سرویس‌های گرید ارائه شده است.

Zh. Shan [18]، به مدل‌سازی و تحلیل کارایی در سیستم گرید با در نظر گرفتن صف‌های اولویت و لحاظ کردن اولویت برای وظایف و زیروظایف پرداخته است. این مدل بر اساس شبکه‌های پتری تصادفی ساخته شده و تحلیل کارایی برای آن با استفاده از خواص شبکه‌های پتری صورت گرفته است. E. Caron در [19] به مدل‌سازی و شبیه‌سازی نحوه زمانبندی وظایف در محیط گرید برای رسیدن به حداکثر توان عملیاتی پرداخته است. این مطالعه بر اساس سیستم‌های صف بوده و فاقد تعریف صوری است و در نهایت نیز شبیه‌سازی انجام شده است.

همه کارهای فوق به نحوی یا اقدام به مدل‌سازی زمانبندی، توزیع و گردش کار در محیط گرید نموده‌اند یا محاسبات مربوط به قابلیت اطمینان یا دیگر پارامترهای QoS را در این محیط به انجام رسانیده‌اند. ولی اکثر محاسبات صرفاً به صورت فرمول‌های ریاضی بوده و از یک مدل صوری سطح بالا استفاده نشده است، علاوه بر آن نمی‌توانند نحوه گردش کار در محیط گرید را به نحوی مطلوب بیان کنند. کارهای مدل‌سازی نیز یا وابسته به یک نرم‌افزار خاص هستند، یا اینکه مدل‌سازی را در سطح یک وظیفه و کار خاص توضیح داده‌اند و عمومیت لازم را ندارند. علاوه بر مشکل فوق، هیچکدام از مدل‌های ارائه شده اصل افزونگی که در محیط گرید بسیار مرسوم بوده و برای افزایش قابلیت اطمینان به کار برده می‌شود را در نظر نگرفته‌اند.

### ۳- محیط گرید و توپولوژی‌های آن

مفاهیم و تکنولوژی‌های موجود در زمینه‌ی گرید بسیار جدید هستند. گرید برای اولین بار در سال 1998 توسط Foster و Kesselman مطرح شد. قبل از آن، تلاش‌های زیادی برای هماهنگ کردن منابع توزیع شده در سطح شبکه و استفاده از آنها برای انجام وظایف صورت گرفته بود ولی گرید مفاهیم جدید و تازه‌ای را - مخصوصاً از لحاظ مدیریت وظایف و تمرکز بر روی اجزاء مرکزی - مطرح کرد. در سال 1998، Foster و Kesselman در [1] تعریف زیر را از گرید ارائه کردند:

"گرید محاسباتی، فراساختاری سخت افزاری و نرم افزاری است که دسترسی اتکاء‌پذیر، سازگار، ارزان، فراگیر و پایدار به منابع محاسباتی موجود در شبکه را فراهم می‌آورد."

با استفاده از شبکه‌های پتری، سیستم‌های صف و نمودارهای UML برای آن طراحی شده است.

در این مقاله مدلی سطح بالا برای تعریف نحوه‌ی توزیع زیروظایف و محاسبه قابلیت اطمینان در محیط گرید و با استفاده از شبکه‌های پتری رنگی زمانی ارائه شده است. ساختار مقاله به این ترتیب است که؛ در بخش دوم کارهای انجام شده در این زمینه و دلیل انجام کار جدید ارائه خواهد شد، در بخش سوم محیط گرید و توپولوژی‌های آن را بحث خواهیم کرد. تعریف و نحوه‌ی استفاده از شبکه‌های پتری و بسط‌های سطح بالای آن - از جمله شبکه‌های پتری رنگی - موضوع بخش چهارم است. در بخش پنجم مدل پیشنهادی آمده است و در بخش ششم مثالی در این مورد ارائه شده است. بخش هفتم نیز به نتیجه‌گیری و کارهای آتی اختصاص یافته است.

### ۲- کارهای مرتبط و انگیزه انجام کار جدید

کارهای زیادی در مورد محاسبه‌ی قابلیت اطمینان [3,7,8,9,10,11] و مدل‌سازی نحوه‌ی گردش کار و انجام وظایف [12,13,14,15] در محیط گرید انجام شده است. اما مدل‌سازی به همراه محاسبه قابلیت اطمینان یا کارایی، کمتر مورد مطالعه قرار گرفته و صرفاً چند مطالعه‌ی موردی و خاص روی آن انجام شده است [16,17,18,19].

Y-Sh. Dai و همکارانش در [3,7,9,10] و G. Levitin و همکارانش در [8,11] به محاسبه قابلیت اطمینان در سرویس‌های گرید پرداخته‌اند. این مطالعات، محاسبات مربوط به قابلیت اطمینان و کارایی را در سرویس‌های گرید انجام داده و صرفاً راه‌حلی برای بدست آوردن و بیشینه کردن آنها ارائه کرده‌اند و هیچ مدل صوری از آن‌ها ارائه نکرده‌اند.

نحوه‌ی گردش کار در محیط گرید، موضوع [12] است که در آن Y. B. Hlaoui و همکارانش به مدل‌سازی گردش وظایف در محیط گرید مبتنی بر سرویس و با استفاده از نمودارهای UML پرداخته‌اند. مدل‌سازی گردش کار در محیط گرید، با استفاده از یک شبکه پتری ساده و در نرم‌افزار Fraunhofer Resource Grid (FhRG) موضوع مورد بحث در [13] است. در این مقاله شبکه‌ی پتری حاصل ارائه شده است ولی این شبکه بسیار ساده بوده و خاص نرم‌افزار مذکور است. Y. Han و همکاران [14,17] نیز با استفاده از شبکه‌های پتری به بیان چگونگی توزیع وظایف در محیط گرید پرداخته‌اند. J. Xu و همکاران در [15] با استفاده از شبکه‌ی پتری توزیع شده به مدل‌سازی نحوه‌ی استدلال در گرید دانش (Knowledge Base Grid) پرداخته‌اند. شبکه‌ی پتری توزیع شده با استفاده از تکنیکی جالب و با توجه به چگونگی مکان‌یابی منابع در محیط گرید ساخته شده است.

L. Li و همکاران [16] با استفاده از سیستم‌های صف اقدام به مدل‌سازی زمانبندی وظایف در محیط گرید نموده و سپس با استفاده از یک شبکه پتری تصادفی (Stochastic Petri Net) آن را به شکل

#### ۴- شبکه‌های پتری رنگی زمانی

شبکه‌های پتری (Petri Nets) ابزار گرافیکی برای توصیف صوری سیستم‌های پویایی هستند که با مفاهیمی همچون همروندی، همگام‌سازی و انحصار متقابل سروکار دارند و عموماً بصورت محیط‌های توزیع شده هستند. یک مدل پتری بصورت گرافیکی با گراف دو قسمتی جهت‌دار نمایش داده می‌شود که در آن دو نوع گره وجود دارد؛ مکان‌ها که بصورت دایره نشان داده می‌شوند و گذرها که بصورت مستطیل نمایش داده می‌شوند [22]. به صورت صوری و بر اساس [23] می‌توان تعریف 4.1 را ارائه کرد.

**تعریف 4.1.** یک شبکه‌ی پتری پنج‌تایی به فرم  $PN = (P, T, F, W, M_0)$  است که:

1.  $P = \{p_1, p_2, \dots, p_m\}$  مجموعه متناهی از مکان‌ها،
2.  $T = \{t_1, t_2, \dots, t_n\}$  مجموعه متناهی از گذرها،
3.  $F \subseteq (P \times T) \cup (T \times P)$  مجموعه کمان‌ها،
4.  $W : F \rightarrow \{1, 2, 3, \dots\}$  تابع وزن (که با استفاده از آن می‌توان ضریب کمان را وارد شبکه‌ی پتری نمود) و
5.  $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$  نشانه‌گذاری اولیه است.

در این تعریف همه گذرها یکسان بوده و زمانی به آنها منتسب نشده است. هر کدام از گذرها به محض فعال شدن می‌توانند شلیک کنند و هیچ ترتیبی بین آنها نیست. یا به عبارتی دیگر، شلیک کردن گذرها غیرقطعی است.

برای فراهم آوردن امکان ارزیابی کمی در رفتار سیستم، نیاز به بسطی از شبکه‌های پتری است که عنصر زمان را وارد محاسبه کند. برای این منظور شبکه‌های پتری زمانی (Timed Petri Nets) تعریف شده‌اند. اساساً دو روش برای اختصاص زمان به شبکه‌های پتری وجود دارد؛ انتساب زمان به گذرها و انتساب زمان به مکان‌ها [24]. در این مقاله از شبکه‌های پتری که زمان را به گذرها اختصاص می‌دهد، استفاده خواهیم کرد. تعریف 4.2 شبکه‌های پتری زمانی را ارائه می‌کند.

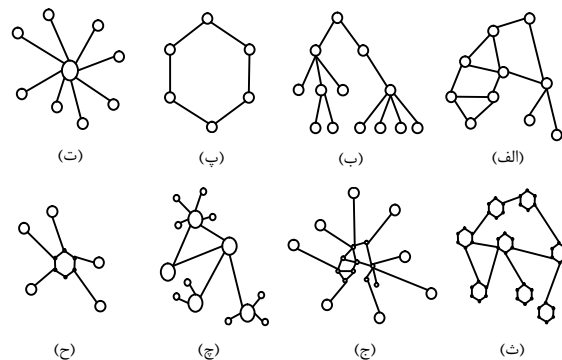
**تعریف 4.2.** یک شبکه‌ی پتری زمانی، شش تایی است به فرم  $PN = (P, T, F, W, M_0, Is)$  که  $PN = (P, T, F, W, M_0, Is)$  مطابق تعریف 4.1 بوده و

$$Is^2 : T \rightarrow Q^+ \times (Q^+ \cup \{\infty\})$$

که  $Is$  در ارتباط با هر گذر زمانی است [25,26].

با استفاده از شبکه‌های پتری زمانی می‌توان توزیع زیروظایف و گردش‌کار در محیط گرید را مدل کرد، ولی هدفی که ما دنبال می‌کنیم، مدل کردن توزیع زیروظایف همراه با محاسبه قابلیت اطمینان است. برای این منظور نیاز به بسط دیگری از شبکه‌های پتری داریم که بتوان بوسیله‌ی آن انواع مختلف زیروظایف به همراه ویژگی‌های آنها را مدل کرد. شبکه‌های پتری رنگی (Colored Petri Nets) برای برآوردن این هدف مناسب هستند. در واقع شبکه‌های پتری رنگی،

در محیط گرید، وظایف بصورت مجرد و بر روی یک سیستم اجرا نمی‌شوند، بلکه این وظایف به زیروظایف شکسته شده و هر کدام از آنها برای اجرا به منابع عضو گرید فرستاده می‌شوند. منابع موجود در شبکه با استفاده از لینک‌های ارتباطی به یکدیگر متصل هستند. یک لینک، امکان ارتباط و تبادل اطلاعات بین دو کامپیوتر مرتبط را فراهم می‌آورد. توپولوژی لینک، ساختار اتصال بین کامپیوترها را معین می‌سازد. چندین توپولوژی لینک در سیستم‌های گرید مطرح است که هر کدام از آنها دارای مزایا و معایبی هستند و برای فراهم آوردن امکاناتی در نظر گرفته شده‌اند. از جمله توپولوژی‌های مطرح می‌توان به توپولوژی ستاره، حلقه، درختی، گراف و مدل‌های ترکیبی اشاره کرد. این توپولوژی‌ها در شکل (1) نشان داده شده‌اند.



شکل 1 - توپولوژی‌های گرید [20]

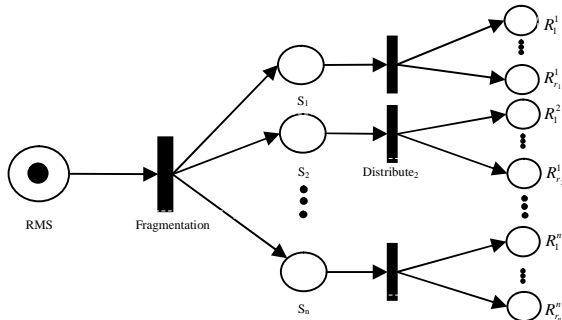
شکل 1- (ت) توپولوژی ستاره‌ی مربوط به سیستم گرید را نشان می‌دهد. در توپولوژی ستاره، RMS در ارتباط با همه کامپیوترهای موجود در شبکه است و عمل پخش زیروظایف بین کامپیوترها و دریافت نتایج از آنها، همگی به عهده‌ی RMS است. توپولوژی ستاره بدلیل ارائه مدلی انتزاعی از سیستم گرید و عدم وجود خطاهای دارای منبع مشترک در آن مورد توجه است.

همانطور که اشاره شد، در محیط گرید به جای اینکه کاربردها در یک وظیفه‌ی مجرد اجرا شوند، به مجموعه‌ای از چندین وظیفه‌ی وابسته به هم یا مستقل از هم تجزیه شده و اجرا می‌شوند. بنابراین بسیاری از کاربردهای گرید به دسته‌ای از کاربردها به نام گردش‌کار (Workflow) متعلق هستند. در اینجا منظور از کاربردهای گردش‌کار، کاربردهایی هستند که نیاز است زمانبندی شده و در روی منابع مختلف اجرا شوند تا نتیجه نهایی حاصل گردد. برای مدل‌سازی و بصری‌سازی این گردش‌کار راه‌حل‌های مختلفی وجود دارد. از آن جمله می‌توان به گراف وظایف اشاره کرد که نوعی گردش‌کار انتزاعی است که یک کاربرد را بصورت مدل عمومی گراف جهت‌دار بدون حلقه نشان می‌دهد [21]. ولی مشکلی که گراف وظایف دارد این است که صرفاً یک مدل گرافیکی است و یک مدل صوری نیست. علاوه بر آن با استفاده از گراف وظایف نمی‌توان محاسبات مربوط به قابلیت اطمینان در سرویس‌های گرید را انجام داد.

اگر حجم کل محاسبات لازم برای اجرای وظیفه‌ی تسلیمی به RMS برابر C باشد و تعداد کل منابع متصل به RMS در حال حاضر برابر r باشد، وظیفه‌ی S را به زیروظایف  $S_1, S_2, \dots, S_n$  تقسیم می‌کند که هر کدام از زیروظایف، مطابق فرمول (1) حجم محاسباتی به اندازه‌ی  $c_j$  خواهند داشت.

$$\sum_{j=1}^n c_j = C \quad (1)$$

که در فرمول (1)، n برابر تعداد زیروظایف است و اگر افزونگی نداشته باشیم، خواهیم داشت:  $n=r$ . ولی در صورت داشتن افزونگی،  $n < r$  خواهد بود. با توجه به اینکه شکستن وظایف به زیروظایف و توزیع آنها در RMS صورت می‌گیرد و بنا به فرض، RMS یک کامپیوتر بسیار سریع و به دور از خطا است، پس این عملیات را می‌توان بوسیله گذرهای فوری شبیه‌سازی کرد. شکل (2) عملیات فوق را نشان می‌دهد.



شکل 2- شکستن وظایف و توزیع زیروظایف در RMS

در شکل (2) گذرهای Fragmentation و Distribute گذرهای فوری بوده و هیچ زمانی را به یال‌ها نمی‌افزایند. پس از شکسته شدن وظیفه به زیروظایف هر کدام از زیروظایف به مجموعه‌ای از منابع منتسب می‌شوند، بطوریکه  $S_1$  به مجموعه‌ی  $\{R_1^1, R_1^2, \dots, R_1^n\}$ ،  $S_2$  به مجموعه‌ی  $\{R_2^1, R_2^2, \dots, R_2^n\}$  و به همین ترتیب  $S_n$  به مجموعه‌ی  $\{R_n^1, R_n^2, \dots, R_n^n\}$  منتسب می‌شوند که  $\sum_{j=1}^n r_j = r$  و برابر تعداد منابع در دسترس است.

اگر داده‌ی لازم جهت مبادله بین منبع k و RMS برای اجرای زیروظیفه‌ی j را برابر  $a_j$  فرض کنیم و سرعت انتقال اطلاعات در لینک ارتباطی بین این منبع و RMS را برابر  $s_k$  بگیریم، زمان لازم برای انتقال داده و اطلاعات از طریق فرمول (2) محاسبه خواهد شد:

$$t_{kj} = \frac{a_j}{s_k} \quad (2)$$

همانطور که اشاره شد این زمان، زمان لازم برای انتقال داده‌ی خام از RMS به منابع و نتایج خروجی از منابع به RMS است. بنابراین:

قدرت شبکه‌های پتری را با زبان‌های برنامه‌سازی سطح بالا تلفیق کرده و اجازه‌ی استفاده از انواع داده و محاسبات بر روی آنها را فراهم می‌آورند [24].

**تعریف 4.3.** شبکه‌های پتری رنگی به کلاسی از شبکه‌های سطح بالا تعلق دارند که شبکه‌های پتری معمولی را به این صورت گسترش می‌دهند: نشانه‌ها می‌توانند با خود داده‌های پیچیده حمل کنند، کمان‌ها شامل توضیحات ورودی - که در شلیک کردن گذرها نقش دارند - و توضیحات خروجی - که در تولید نشانه‌ها پس از شلیک کردن گذرها مؤثرند - هستند [27].

در تعریف شبکه‌های پتری رنگی، نحو و روش خاصی برای مدل‌سازی بیان نشده است. تعاریف و توضیحات شبکه می‌توانند به روشهای مختلف و توسط نمادهای ریاضی استاندارد یا زبان‌های برنامه‌نویسی سطح بالا بیان شوند. CPN Tools [28] یکی از ابزارهایی است که مدل‌سازی، اعتبارسنجی و تحلیل شبکه‌های پتری رنگی را امکان‌پذیر می‌سازد. CPN Tools امکان استفاده از شبکه‌های پتری رنگی زمانی (Timed Colored Petri Nets) را نیز فراهم می‌آورد. در TCPN علاوه بر اینکه گذرها دارای زمان بوده، نشانه‌ها نیز دارای مُهر زمانی (Timestamp) هستند. مُهر زمانی مربوط به نشانه‌ها در زمان تعریف نشانه‌ها تولید می‌شوند [27].

#### 5- مدل پیشنهادی

در این بخش مدل پیشنهادی با توجه به تعریف مسئله و مفروضات آن ارائه می‌شود. فرض می‌کنیم که محیط گرید دارای توپولوژی ستاره است و در نتیجه RMS در ارتباط با همه‌ی منابع موجود در گرید است. RMS پس از دریافت وظیفه از کاربر، آن را به زیروظایف تقسیم می‌کند. با توجه به اینکه از تکنیک افزونگی در انتساب منابع به زیروظایف استفاده شده است، این تقسیم باید بگونه‌ای باشد که تعداد زیروظایف (n) از تعداد منابع در دسترس (r) کمتر باشد. پس از این تقسیم، RMS هر زیروظیفه را به بیش از یک منبع منتسب می‌کند. پس هر زیروظیفه به دو منبع یا بیشتر منتسب می‌شود ولی هر منبع فقط یک زیروظیفه را پردازش می‌کند. برای استفاده‌ی حداکثری از منابع، RMS از همه‌ی منابع در دسترس برای اجرای وظیفه بهره می‌گیرد. وابستگی بین داده‌ای در بین زیروظایف وجود ندارد و هیچ‌کدام از آنها برای اجرا شدن نیازی به اجرای زیروظیفه‌ی دیگر ندارند و به محض دریافت اطلاعات لازم از RMS، توسط منابع، پردازش زیروظیفه صورت می‌گیرد.

برخی از این مفروضات طبیعی بوده و سازگار با ساختار سیستم گرید است ولی برخی دیگر جهت راحت کردن مسأله از مفروضات پیچیده است. در هر حال با اضافه کردن مفروضات جدید می‌توان مسأله را با محدودیت‌های دیگری نیز حل کرد. با توجه به مفروضات فوق و با استفاده از [8,9,10,11] می‌توان محاسبات حاصل را به شرح زیر انجام داد:



داده‌اند شناسایی کنیم. به همین دلیل بعد از ارسال نتایج به  $P_j$  در هر کدام از  $P_j$ ها اقدام به مینیمم‌گیری می‌کنیم.

همچنانکه در شکل (3) نیز دیده می‌شود، نشانه‌هایی که از مسیرهای مختلف آمده‌اند، با خود زمان‌های مربوط به فرمول‌های (2) و (3) را جمع زده‌اند و آورده‌اند. پس در این مرحله - که در RMS انجام می‌گیرد- از زمان منتسب به نشانه‌ها مینیمم می‌گیریم و در نتیجه نشانه‌هایی که کمترین زمان را داشته باشد به  $P_{collect}$  راه می‌یابند.

تا این قسمت همه‌ی زیروظایف انجام شده و زمان مربوط به زیروظیفه‌ای که کمتر از زیروظیفه‌های مشابه اجرا شده است را داریم. طبق تعریف، یک وظیفه در صورتی کامل می‌شود که تمام زیروظایف آن کامل شده و نتایج آنها به RMS برگشت داده شده باشند. همچنین زمان انجام یک وظیفه برابر است با بزرگترین زمان انجام زیروظایف. در واقع چون همه‌ی زیروظایف بصورت هم‌روند اجرا می‌شوند، زمان انجام مربوط به وقت‌گیرترین زیروظیفه برابر خواهد بود با زمان انجام کل وظیفه. وقت‌گیر بودن یک زیروظیفه صرفاً به معنای بزرگ بودن داشتن حجم محاسبات بالا نیست، بلکه عواملی مانند کُند بودن منبعی که آن زیروظیفه را پردازش کرده یا کم بودن پهنای باند لینک ارتباطی آن منبع با RMS نیز می‌توانند موثر باشند. عملیات ماکزیمم‌گیری در RMS صورت می‌گیرد و مطابق شکل (3)، توسط گذر فوری Max مدل‌سازی می‌شود.

کمانی که از مکان  $P_{collect}$  به گذر Max متصل است دارای ضریب است. این ضریب برابر تعداد زیروظایف بوده و بیانگر این مطلب است که گذر Max در صورتی شلیک می‌کند که همه زیروظایف یک وظیفه انجام شده باشند.

پس از شلیک گذر Max وظیفه‌ی مذکور تمام شده و وظیفه‌ی جدید می‌تواند وارد سیستم شود. مطابق شکل (3)، تعداد مکان‌ها برابر  $3 \times r + 2 \times n + 3$ ، گذرهای فوری برابر  $2 \times n + 2$  و گذرهای زمانی نیز برابر  $3 \times r$  خواهد بود.

پس اگر تعداد مکان‌ها را با  $P$  و تعداد گذرها را با  $T$  نشان دهیم، فرمول‌های (4) و (5) را خواهیم داشت:

$$P = 3 \times r + 2 \times n + 3 \quad (4)$$

$$T = 3 \times r + 2 \times n + 2 \quad (5)$$

شبکه‌ی پتری فوق، علاوه بر قابلیت مدل‌سازی گردش کار در سیستم گرید (با مفروضات گفته شده) توانایی محاسبه‌ی معیار قابلیت اطمینان را نیز دارد.

در کاربردهایی که زمان اجرای وظیفه، یک فاکتور مهم است، قابلیت اطمینان بصورت احتمال اجرای وظیفه ورودی در زمان کمتر از یک مقدار داده شده تعریف می‌شود. بنابراین می‌توان از روی مدل فوق و با توجه به زمان داده شده، احتمال‌های اجرای وظیفه‌ی ورودی در کمتر از این زمان داده شده، را با یکدیگر جمع کرده و در نهایت

این دو عملیات را می‌توان به‌صورت جدا از هم یا به‌صورت یکجا مدل کرد (که در مورد مدل پیشنهادی و مثال هر دو حالت بحث شده است). به طریقه‌ی مشابه با فرمول (1) می‌توان زمان لازم برای اجرای زیروظیفه‌ی  $j$  توسط منبع  $k$  را مانند فرمول (3) محاسبه کرد:

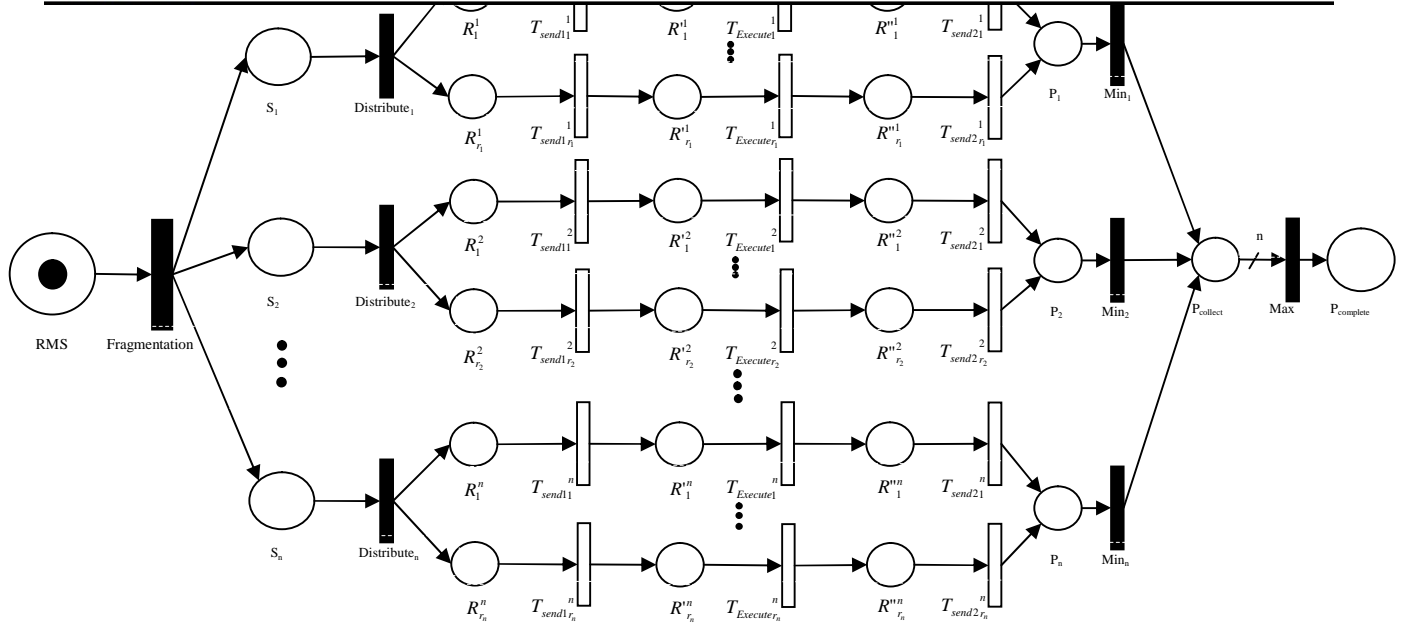
$$T_{kj} = \frac{c_j}{x_k} \quad (3)$$

در ادامه‌ی شکل (2) و مطابق با فرمول‌های (2) و (3) می‌توان مدل حاصل را کامل‌تر کرده و ارسال زیروظایف به منابع و اجرا شدن آن‌ها را نیز مدل کرد. به دلیل اختصار، از آوردن مرحله به مرحله‌ی مدل پیشنهادی خودداری شده و مدل نهایی حاصل در شکل (3) نشان داده شده و از این قسمت به بعد توضیحات روی شکل (3) ارائه خواهند شد.

مدل‌سازی ارسال و اجرای زیروظایف مطابق شکل (3) توسط گذرهای  $T_{send1j}$  و  $T_{executej}$ ، که گذرهای زمانی هستند، انجام شده است. قابل ذکر است که نشانه‌هایی که در مکان‌ها قرار می‌گیرند، رنگی هستند و می‌توانند داده را ذخیره کنند، بنابراین برای پشتیبانی از احتمالی نبودن انتخاب گذرها برای شلیک کردن در بین منابع مختلف و یا به‌عبارتی دیگر برای فراهم آوردن تصادفی شلیک کردن گذرهای مختلف، زمان را به یال‌ها نسبت داده‌ایم و وقتی نشانه‌ها از کمان‌ها عبور می‌کنند، زمان منتسب به آنها را با خود حمل کرده و بترتیب این زمان‌ها را جمع می‌زنند.

نشانه‌ها علاوه بر زمان، احتمالات را نیز با خود حمل می‌کنند. اگر احتمال اینکه لینک ارتباطی بین منبع  $j$  و RMS، در حین انتقال داده fail نکند را با  $q_j(\tau_{kj})$  نشان دهیم، این مقدار در حین گذر از  $T_{send1j}$  به نشانه‌ها افزوده می‌گردد و همچنین اگر احتمال اینکه منبع  $j$  در حین اجرای زیروظیفه دچار خرابی نشود را برابر  $p_j(T_{kj})$  بگیریم، این مقدار در انتقال از گذر  $T_{executej}$  به نشانه‌ها اضافه می‌شود. البته شایان ذکر است که زمان اجرا برای یک زیروظیفه برابر است با جمع دو زمان مذکور و احتمال نیز برابر حاصلضرب احتمالات است. نحوه‌ی محاسبه  $q_j(\tau_{kj})$  و  $p_j(T_{kj})$  در منابع [8,9,10] آمده است. نکته حائز اهمیت این است که زمان اجرای زیروظیفه‌ی  $j$  که به منبع  $k$  اختصاص داده شده است، با احتمال  $p_j(T_{kj}) * q_j(\tau_{kj})$  برابر  $T_{kj} + \tau_{kj}$  و با احتمال  $1 - [p_j(T_{kj}) * q_j(\tau_{kj})]$  برابر بینهایت است (یعنی منبع یا لینک ارتباطی fail کرده است).

پس از اجرای زیروظایف در منابع، نوبت به ارسال نتایج به RMS می‌رسد. در این مرحله هر مجموعه از منابع که زیروظیفه‌ی یکسانی را پردازش کرده‌اند، از طریق گذرهای زمانی مربوطه ( $T_{send2j}$ )، نتایج را به یک مکان می‌فرستند. دلیل استفاده از TCPN نیز در اینجا روشن می‌شود؛ می‌خواهیم منابعی که زیروظیفه‌ی مربوطه را سریعتر انجام



شکل 3- مدل پیشنهادی

با توجه به توضیحات فوق می‌توان مثال زیر را مطرح کرد:  
 سیستمی با چهار منبع را در نظر بگیرید؛ فرض کنید که وظیفه ورودی، توسط RMS به دو زیروظیفه تقسیم می‌شود و زیروظیفه اول به منابع 1 و 2 و زیروظیفه دوم به منابع 3 و 4 اختصاص داده می‌شود. مشخصات لازم برای حل مسئله در جداول (1) تا (3) ارائه شده است.

جدول 1- مشخصات منابع و لینک‌های ارتباطی (مثال اول)

	منبع 1	منبع 2	منبع 3	منبع 4
<b>x</b>	4	6	3	8
<b>s</b>	3	6	4	5

جدول 2- مشخصات زیروظایف (مثال اول)

حجم محاسبات زیروظایف	حجم داده مبادله شده
$c_1=12$	$a_1=6$
$c_2=24$	$a_2=20$

جدول 3- احتمالات خراب نشدن منابع و لینک‌های ارتباطی (مثال اول)

	منبع 1	منبع 2	منبع 3	منبع 4
	زیروظیفه 1	زیروظیفه 1	زیروظیفه 2	زیروظیفه 2
<b>p</b>	0.75	0.80	0.92	0.85
<b>q</b>	0.94	0.89	0.77	0.71

حال با توجه به روابط (6) و (7) می‌توان تعداد کل مکان‌ها و گذرهای حاصل را بدست آورد:

$$P = 16, T = 13$$

"قابلیت اطمینان" را بدست آورد. برای این کار در مدل ساخته شده از نشانه‌ها بهره می‌گیریم، زیرا هم زمان و هم احتمال اجرای زیروظایف و در نهایت وظیفه کلی را با خود دارند. در بخش بعد مثالی برای روشن شدن مطلب ارائه خواهد شد.

### ۶- مثال

در شبکه پتری نشان داده شده در شکل (3)، با توجه به اینکه حجم داده‌ی مبادله شده را بدون توجه به نوع داده (داده‌ی خام یا نتایج)، برابر  $a_j$  فرض کرده‌ایم، می‌توان از گذرهای  $T_{send2_j^i}$  صرفنظر کرده و آنها را با گذرهای  $T_{send1_j^i}$  ادغام کرد [17] و با  $T_{Transmit}$  نشان داد و در نتیجه از تعداد گذرها به اندازه  $r$  کاسته خواهد شد. با اعمال این تغییر، در قسمت ماکزیم‌گیری نیز به تعداد یک مکان از مجموعه‌ی مکان‌ها کاسته خواهد شد. همچنین با توجه به اینکه عملیات قسمت‌بندی (Fragmentation) و توزیع (Distribution) هر دو در RMS صورت گرفته و زمان مربوط به آنها ناچیز است - و با گذرهای فوری پیاده‌سازی شده است - می‌توان این گذرها را نیز با هم تلفیق کرده و با یک گذر نشان داد. با توجه به توضیحات قبل، به تعداد  $n$  گذر نیز از مجموعه گذرها کم خواهد شد.

برای اینکه شبکه پتری حاصل قابلیت کارکرد مجدد بعد از اتمام وظیفه را داشته باشد، پس از مکان  $P_{complete}$  یک گذر اضافه کرده و با شلیک کردن آن یک نشانه در مکان RMS گذاشته می‌شود. بنابراین فرمول‌های 4 و 5 به صورت زیر درخواهند آمد:

$$P = 3 \times r + n + 2 \quad (6)$$

$$T = 2 \times r + n + 3 \quad (7)$$

با استفاده از بسط‌های سطح بالایی از شبکه‌های پتری-که اصولاً برای مدل‌سازی سیستم‌های همروند طراحی شده‌اند- می‌توان نحوه اجرای وظایف در یک سیستم گرید را مدل‌سازی نمود. کاری که ما در این مقاله انجام دادیم، علاوه بر مدل‌سازی نحوه اجرای وظایف در سیستم گرید، محاسبه قابلیت اطمینان بود که با استفاده از ابزار CPN Tools صورت گرفت.

از جمله کارهای آینده که می‌توان در این زمینه انجام داد، می‌توان به موارد زیر اشاره کرد:

- برداشتن برخی از مفروضات از مسأله و دخیل کردن فرض‌های واقعی‌تر در حل مسأله.
- استفاده از مدل مذکور و تعمیم آن برای توپولوژی‌های دیگر گرید از جمله توپولوژی درختی و سلسله‌مراتبی.
- افزودن سیستم صف به هر کدام از منابع و مدل‌سازی اجرای کارهای محلی در هر کدام از منابع.
- لحاظ کردن وابستگی داده‌ای بین زیروظایف و انتظار برخی از زیروظایف‌ها جهت گرفتن داده مورد نیاز از زیروظایف دیگر.

### سپاسگزاری

بدین وسیله از "بنیاد ملی نخبگان" به‌عنوان یاری‌گر، تقدیر و تشکر می‌شود.

مدل حاصل با استفاده از ابزار CPN Tools شبیه‌سازی شده و نتایج حاصل از ارزیابی قابلیت اطمینان برای زمان کمتر از 10 در شکل (4) ارائه شده است.

خطوط زیر نیز توسط مونیتوری که با استفاده از CPN Tools نوشته شده، محاسبه و چاپ می‌گردد و در واقع ارزیابی قابلیت اطمینان این سیستم، با مشخصات داده شده است.

START

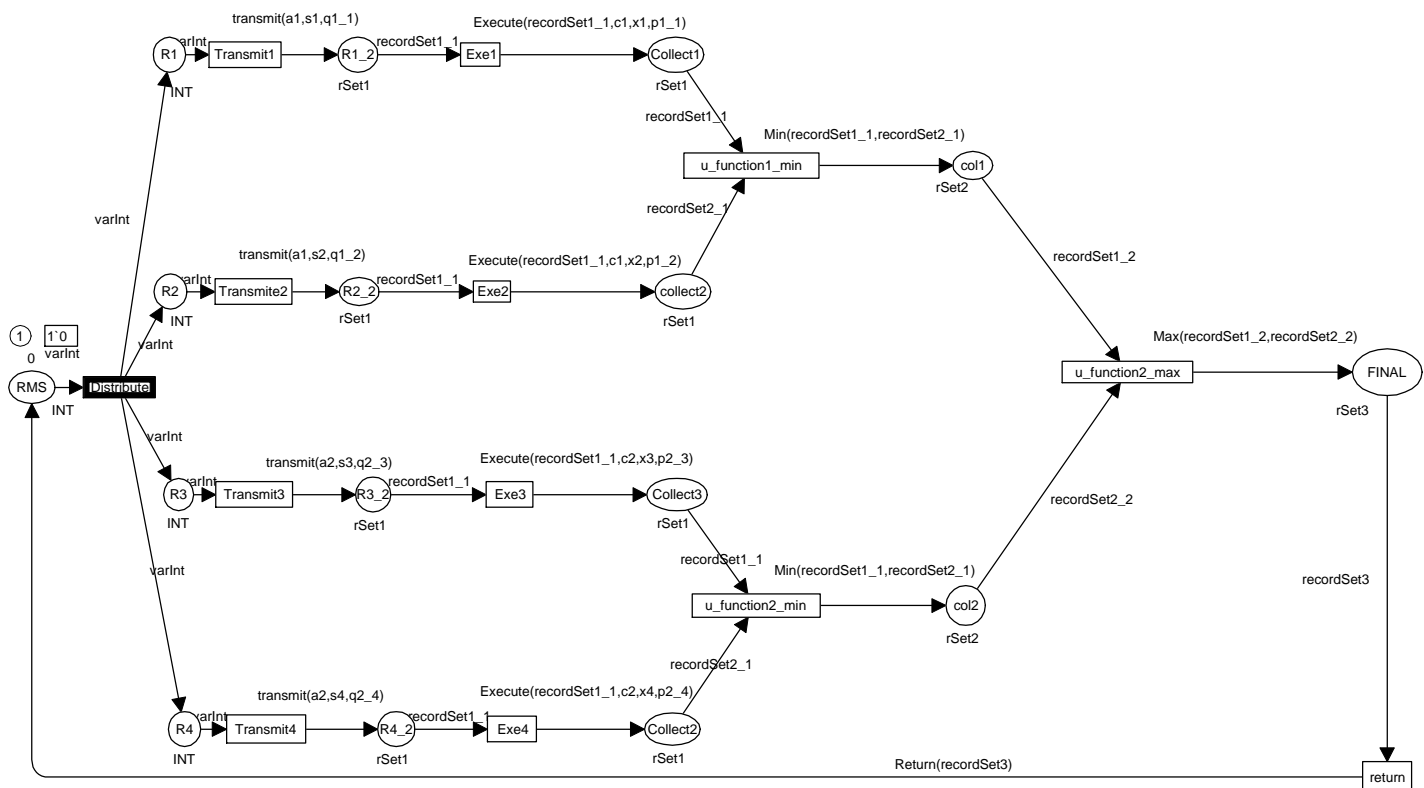
The input task is completed in 7 second with probability 0.54,  
 And completed in 13 second with probability 0.25 ,  
 And task fail with probability 0.21 .

Reliability or a probability that output is produced in time less than 10 is equal to : 0.54 .

END

### ۷- نتیجه گیری

تکنولوژی گرید که به اعتقاد برخی از کارشناسان، شانس دوم اینترنت نیز محسوب می‌شود، به سرعت در حال پیشرفت بوده و هر روز مباحث جدیدی در این زمینه مطرح شده و یا مباحث موجود بسط داده می‌شوند. اجرای وظایف تسلیمی به سیستم مدیریت منابع در گرید و نیز قابلیت اطمینان در این اجرا از جمله مباحث مطرح در این زمینه است که مقالات زیادی درباره این موضوع و موضوعات مشابه نوشته شده است.



شکل 4- شبکه پتری شبیه‌سازی شده با CPN Tools



*Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06)*, IEEE 2006.

- [17] Han Y., Luo X., "Modeling and Performance Analysis of Grid Task Scheduling Based on Composition and Reduction of Petri Nets", *Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06)*, IEEE 2006.
- [18] Shan Zh., Lin Ch., Ren F., Wei Y., "Modeling and Performance Analysis of a Multiserver Multiqueue System on the Grid", *Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03)*, 2003.
- [19] Caron E., Garonne V., Tsaregorodtsev A., "Defination, modeling and simulation of a grid computing scheduling system for high throughput computing", *Future Generation Computer Systems* 23, pp. 968-976, Elsevier, 2007.
- [20] Berman F., Hey A.J.G., Fox G.C., *Grid Computing: Making the Global Infrastructure a Reality*, Wiley Series in Communications Networking & Distributed Systems; Chapter: "Peer-To-Peer Grid Databases for Web Service Discovery", Chapter Author: Hoschek W., May 2003.
- [21] Lee J., Chin S., Lee H., Yoon T., Chung K., Yu H., *Adaptive Workflow Scheduling Strategy in Service-Based Grids*, Book Series: Lecture Notes in Computer Science, pp. 298-309, Springer, 2007.
- [22] Ajmone Marsan M., Balbo G., Conte G., Donatelli S., Franceschinis G., *Modeling with Generalized Stochastic Petri Nets*, Wiley Series in Parallel Computing, John Wiley and Sons, 1995.
- [23] Murata T., "Petri nets: Properties, analysis and applications", *Proceedings of the IEEE*, Vol. 77, pp. 541-580, April 1989.
- [24] Furfaro A., Nigro L., Pupo F., "Distributed Simulation of Timed Coloured Petri Nets", *Proceedings of the Sixth IEEE International Workshop on Distributed Simulation and Real-Time Applications*, pp. 159-166, 2002.
- [25] Kordic V., *Petri Net Theory and Applications*, Chapter: Model Checking of Time Petri Nets, Chapter Author: Boucheneb H., Hadjidj R., I-Tech Education and Publishing, Vienna, Austria, First Edition, February 2008.
- [26] Kordic V., *Petri Net Theory and Applications*, Chapter: From Time Petri Nets to Timed Automata, Chapter Author: Cassez F., Roux O.H., I-Tech Education and Publishing, Vienna, Austria, First Edition, February 2008.
- [27] Jensen K., "Coloured Petri Nets – Basic concepts, analysis methods and practical use", *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag Vol. 1, 2, 3, 1992-98.
- [28] Jensen K., Kristensen L.M., Wells L., "Coloured Petri Nets and CPN Tools for modeling and validation of concurrent systems", *International Journal on Software Tools for Technology Transfer (STTT)*, pp. 213-254, Springer Berlin / Heidelberg, 2007.
- [1] Foster I., Kesselman C., *The Grid: Blueprint for a New computing Infrastructure*, Morgan Kuffman, 1999.
- [2] Foster I., Kesselman C., *The Grid 2: Blueprint for a New Computing Infrastructure*, Los Alios, Morgan-Kuffman; 2003.
- [3] Dai Y.S., Levitin G., "Optimal Resource Allocation for Maximizing Performance and Reliability in Tree-Structured Grid Services", *IEEE Transaction on Reliability*, Vol. 56, No.3, September 2007.
- [4] Plestys R., Vilutis G., Sandonavicius D., "the Measurement of Grid QoS Parameters", *Proceedings of the ITI 2007; 29<sup>th</sup> Int. Conf. on Information Technology Interfaces*, Cavtat, Croatia, June 2007.
- [5] Sun X-He., Wu M., "Quality of Service of Grid Computing: Resource Sharing", *The Sixth International Conference on Grid and Cooperative Computing (GCC 2007)*.
- [6] Wang X., Luo J., *Architecture of Grid Resource Allocation Management Based on QoS*, Book chapter of "Grid and Cooperative Computing", Springer Berlin / Heidelberg, pp. 81-88, 2004.
- [7] Dai Y.S., Xie M., Poh K.I., "Reliability of grid service systems", *Computers & Industrial Engineering* 50, pp. 130-147, Elsevier, 2006.
- [8] Levitin G., Dai Y.S., "Service reliability and performance in grid system with star topology", *Reliability Engineering and System Safety* 92, pp. 40-46, Elsevier, 2007.
- [9] Levitin G., Dai Y.S., "Reliability and Performance of Tree-Structured Grid Services", *IEEE Transaction on Reliability*, Vol. 55, No. 2, June 2006
- [10] Dai Y.S., Levitin G., Trivedi K.S., "Performance and Reliability of Tree-Structured Grid Service Considering Data Dependence and Failure Correlation", *IEEE Transaction on Computer*, Vol.56, No.7, July 2007.
- [11] Levitin G., Dai Y.S., Ben-Haim H., "Reliability and Performance of Star Topology Grid Service With Precedence Constraints on Subtask Execution", *IEEE Transaction on Reliability*, Vol. 55, No. 3, September 2006.
- [12] BenDaly Hlaoui Y., Jemni BenAyed L.; "Toward an UML-based Composition of Grid Services Workflows", *Research Unit of Technologies of Information and Communication*, Tunisia, ACM, AUPC'08, July 2008.
- [13] Neubauer F., Hoheisel A., Geiler J., "Workflow-based Grid applications", *Future Generation Computer Systems* 22, pp. 6-15, Elsevier, 2006.
- [14] Han Y., Jiang Ch., Luo X., "Resource Scheduling Model for Grid Computing Based on Sharing Synthesis of Petri Net", *The 9<sup>th</sup> International Conference on Computer Supported Cooperative Work in Design Proceedings*, May 2005.
- [15] Xu J., Wu Zh., Chen H., "Distributed Petri Net for Knowledge Base Grid Reasoning", *IEEE International Conference on Systems, Man and Cybernetics*, Vol.1. pp. 593-597, Oct. 2003.
- [16] Li L., FangChun Y.; "Modeling and Performance Analysis of a Priority-based Scheduling Scheme in Service Grid",

مراجع

# SID



سرویس های ویژه



سرویس ترجمه تخصصی



کارگاه های آموزشی



بلاگ مرکز اطلاعات علمی



عضویت در خبرنامه



فیلم های آموزشی

## کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی



مباحث پیشرفته یادگیری عمیق؛  
شبکه های توجه گرافی  
(Graph Attention Networks)



کارگاه آنلاین آموزش استفاده از  
وب آوساینس



کارگاه آنلاین مقاله روزمره انگلیسی