دانشکده علوم ریاضی

پایان نامه کارشناسی ارشد

الگوریتم‌های سری و موازی برای شبیه‌سازی سیستم صف‌های سری

احمد مهدیلوزاد

استاد راهنما

دکتر حامدرضا طارقیان

استاد مشاور

دکتر محمد امینی

شهریور ۱۳۸۸

چکیده

سیستم‌های صف به‌واسطه توانایی، و در عین حال طبیعت ساده‌شان در نواحی کاربردی مختلف ابزار مهمی به‌حساب می‌آیند. اگرچه برای یک مدل صف غالبا می‌توان نتایج تحلیلی ارائه داد با این وجود شبیه-سازی سیستم‌های صف به‌عنوان یک تکنیک مهم در زمینه ارزیابی کارایی باقی می‌ماند. در این تحقیق، یک سیستم صف سری به دو روش زمان‌بندی پیشامد و سریع، شبیه‌سازی شده است. به‌منظور تسریع اجراهای شبیه‌سازی سیستم صف، الگوریتم‌های سری و موازی برای شبیه‌سازی سیستم‌های صف سری با بافرهای نامتناهی و متناهی ارائه شده‌اند. الگوریتم‌ها یک روش محاسباتی ساده بر اساس روابط بازگشتی به‌عنوان یک نمایش از دینامیک‌های سیستم ارائه می‌دهند. همچنین، تحلیل مختصری از کارایی الگوریتم‌ها ارائه شده است تا نشان دهد که الگوریتم‌ها به زمان کم و حافظه پایین نیاز دارند.

فهرست مطالب

عنوان **صفحه**

فصل ۱ مقدمه ۱

فصل ۲ بررسی منابع و پیشینه موضوع ۵

فصل ۳ شبیه‌سازی سیستم صف‌های سری ۲۴

۳-۱- کلیات سیستم‌های صف ۲۵

۳-۲- مدل‌های پایه عبارت‌های بازگشتی سیستم‌های صف ۲۶

۳-۳-۱- صف $G / G / 1$ ۲۵

۳-۲-۲- سیستم صف‌های سری تک سرویس‌دهنده ۲۷

۳-۲-۲-۱- سیستم‌های سری بسته ۲۹

۳-۲-۲-۲- صف‌های سری با بافرهای متناهی ۲۹

۳-۳- معیارهای کارایی ۳۲

۳-۳-۱- شبکه‌ها با گره‌های تک سرویس‌دهنده ۳۲

۳-۳-۲- سیستم صف‌های سری ۳۳

۳-۴- شبیه‌سازی سیستم‌های گسسته پیشامد ۳۳

۳-۵- الگوریتم‌های شبیه‌سازی سریع ۴۴

۳-۶- نتایج شبیه‌سازی به روش سنتی و سریع ۴۸

فصل ۴ الگوریتم‌های موازی کارآمد برای شبیه‌سازی صف‌های سری ۶۹

۴-۱- الگوریتم‌های موازی برای شبیه‌سازی سیستم صف‌های سری با بافرهای نامتناهی و متناهی ۷۰

۴-۱-۱- شبیه‌سازی صف‌های سری با بافرهای نامتناهی ۷۰

۴-۱-۱-۱- مطالعه کارایی الگوریتم شبیه‌سازی موازی ۷۳

۴-۱-۲- شبیه‌سازی صف‌های سری با بافرهای متناهی ۷۶

۴-۲- ارزیابی معیار کارایی ۷۹

فصل ۵ نتیجه‌گیری و پیشنهاد ۸۴

۵-۱- پیشنهادات ۸۵

منابع، مراجع و مآخذ ۸۶

پیوست‌ها

فصل ۱

مقدمه

انتظار در صف هر چند ناخوشایند است، اما متأسفانه بخشی از واقعیت اجتناب‌ناپذیر زندگی را تشکیل می‌دهد. انسان‌ها، در زندگی روزمره خود با انواع مختلف صف، که به از بین رفتن وقت، نیرو و سرمایه آن‌ها می‌انجامد، روبه‌رو می‌شوند. اوقاتی که در صف‌های اتوبوس، ناهارخوری، خرید و نظایر آن‌ها به هدر می‌رود، نمونه‌های ملموسی از این نوع اتلاف‌ها در زندگی است. البته در جوامع امروزی، صف‌های مهمتری وجود دارد، که هزینه‌های اقتصادی، و اجتماعی آن‌ها به مراتب بیشتر از نمونه‌های ساده فوق است. از آن جمله می‌توان صف‌های حاصل از ترافیک شهری، و نیز صف‌هایی که در فرودگاه‌ها، بنادر، موسسات مخابراتی و در پشت فرایندهای تولید تشکیل می‌شود را نام برد. در مجموع، شاید بتوان گفت که انتظار در صف دیگر استثناء نیست و به صورت قاعده در آمده است.

کاهش اثرات نامطلوب انتظار در صف بدون شناخت خصائص این پدیده امکان‌پذیر نیست. نظریه صف که به مطالعه صف‌ها از دیدگاه ریاضی می‌پردازد، تأثیر عوامل تشکیل‌دهنده صف و راه‌های منطقی کاهش زمان انتظار را بررسی می‌کند.

اما اصولا چرا صف تشکیل می‌شود؟ جواب این سوال بسیار ساده است. علت این امر وجود تقاضای بیشتر برای خدمات این سرویس‌دهنده‌ها اعم از تجهیزات یا نیروی انسانی در مقایسه با توان خدمت‌رسانی

این سرویس‌دهنده‌ها می‌باشد. علت این که چرا خدمات ارائه شده توسط این تجهیزات کمتر از تقاضا است را باید در تعداد کم سرویس‌دهنده، محدودیت فضای موردنیاز در مقایسه با خدمت ارائه شده، و... جستجو نمود. این‌گونه محدودیت‌ها می‌تواند با سرمایه‌گذاری بیشتر روی سیستم از بین برود. به منظور سرمایه‌گذاری صحیح نخست می‌بایست به سوالاتی از این قبیل « هر مشتری چقدر منتظر خواهد ماند؟ » ، « معمولا چند نفر در صف می‌باشند؟ » و « چه تعداد سرویس‌دهنده لازم است؟ » ... پاسخ داده شود. نظریه صف پاسخ به این سوالات را با استفاده از تجزیه و تحلیل مدل‌های ریاضی ارائه شده به دست می‌دهد .

اما در برخی موارد گسترش مدل‌های تحلیلی برای سیستم‌های صف‌بندی ممکن نیست. این موضوع می‌تواند معلول مشخصه‌های ورودی یا مکانیزم‌های سرویس، پیچیدگی طرح سیستم، طبیعت نظم صف، یا ترکیبی از این‌ها باشد. به عنوان مثال، بررسی سیستم‌های صف‌بندی در شرایط گذرا، تحلیل مدل‌های صف-بندی G/G/c و یا صف‌بندی غیرمارکوفی به صورت دقیق، دشوار و عملا غیر ممکن است[۱].

در چنین شرایطی شبیه‌سازی می‌تواند برای تحلیل بسیاری از سیستم‌های صف‌بندی پیچیده‌ای که عملا با آن‌ها مواجه هستیم ابزار بسیار مهم و اغلب تنها شیوه باشد[۱]. در شبیه‌سازی کامپیوتری سیستم، مدلی از سیستم طراحی می‌گردد . با اجرای کامپیوتری مدل، رفتار سیستم در دسترس تحلیل‌گر سیستم قرار می‌گیرد. تحلیل‌گر در هر زمان می‌تواند تغییرات لازم را در مدل سیستم ایجاد نماید و نتیجه آن را بعد از اجرای مدل مشاهده کند. شناخت به‌دست‌آمده از سیستم با این روش، به هنگام پیشنهاد یا انجام تغییرات در سیستم، بسیار سودمند است. با ایجاد تغییر در ورودی‌های شبیه‌سازی و بررسی خروجی‌های به دست آمده، اطلاعات مفیدی درباره رفتار متغیرهای مهم سیستم به دست می‌آید. تحلیل‌گر با استفاده از مدلسازی سیستم، قبل از پیاده‌سازی سیستم جدید و صرف هزینه، به عملکرد سیستم صف واقف می‌شود و در صورت برخورد با مسائل، قبل از طراحی به رفع آن می‌پردازد[۳,۲,۱] .

علی‌رغم این که شبیه‌سازی برای تحلیل سیستم‌های صف‌بندی مفید و در شرایطی تنها رویکرد است برای نتیجه‌گیری دقیق، به تعداد تکرار بسیاری نیاز می‌باشد که این مساله هم زمان‌بر و هم هزینه‌بر است.

یک راه‌حل برای رفع این مشکل استفاده از کامپیوترهایی با چند پردازنده می‌باشد که می‌توان اجرای یک برنامه شبیه‌سازی را به صورت $SIMD$[1] (چند جریان داده و یک دستور) یا $MIMD$[2] (چند جریان داده و چند دستور) روی چند پردازنده توزیع کرد[۱۰,۱۳-۱۶,۲۷] و با به دست آوردن سریع نتایج، این ایراد شبیه‌سازی را برطرف کرد. برای استفاده از چنین کامپیوترهایی، باید الگوریتم‌هایی طراحی شود که قابلیت توزیع روی چند پردازنده را داشته باشد، این الگوریتم‌ها را می‌توان به دو دسته تقسیم کرد : دسته اول، الگوریتم‌های شبیه‌سازی که خود خاصیت توازی دارند و دسته دوم، الگوریتم‌هایی که قابلیت توازی ندارند و به شبیه*سازی موازی و توزیع شده*[3] معروف هستند (به عنوان مثال، شبیه‌سازی ترافیک هوایی[۱۴]).

در شبیه‌سازی سیستم صف‌های سری می‌توان از الگوریتم‌های هر دو دسته استفاده کرد [۱۰,۱۲,۱۳,۲۴,۲۷,۲۹] به علت عدم دسترسی آسان به کامپیوترهای با چند پردازنده و با توجه به اینکه در منابع [۱۲,۱۳] الگوریتم‌های شبیه‌سازی ارائه شده است که خود خاصیت توازی دارند ما نیز در این تحقیق از الگوریتم‌های دسته اول استفاده می‌کنیم .

پردازش موازی به ویژه، برای شبیه‌سازی سیستمی، با ساختاری مشخص و از پیش تعیین‌شده، برخلاف سیستمی پیچیده با وابستگی‌های فراوان، می‌تواند مفید باشد. در این خصوص، یک سیستم خوب، سیستمی است که متغیرهای خروجی می‌تواند توسط مجموعه‌هایی از روابط بازگشتی تولید شود. یک نمونه از سیستم‌هایی که از روابط بازگشتی استفاده می‌کنند شبکه‌های صف می‌باشند. در این موارد، مشکل، تعیین یک مجموعه موثر از روابط بازگشتی برای نمایش سیستم (معمولا انتخاب‌های زیادی وجود دارد) و آن‌گاه تعیین چگونگی اجرای این روابط در حالت موازی می‌باشد. به طور نمونه یک درجه بالا از توازی، الگوریتم-های جزئی شده نیاز دارد، و شبیه‌سازی‌های پیچیده خانواده‌ای از این الگوریتم‌های هدف خاص را شامل می-شود[۱۰].

[1] Single Insruction Stream-Multiple Data Stream
[2] Multiple Instruction Stream-Multiple Data Stream
[3] Parallel and Distributed Simulation

در این پایان‌نامه، در فصل ۲ تاریخچه مختصری از کارها و تحقیق‌های انجام شده در زمینه شبیه‌سازی سیستم‌های صف و پردازش موازی آن‌ها بیان شده است. در فصل ۳، ابتدا به معرفی سیستم صف، رفتار صف-های سری با مدل‌های صف‌بندی G/G/1 و ظرفیت‌های نامتناهی و متناهی و در نهایت به شبیه‌سازی می-پردازیم. برای شبیه‌سازی صف‌های سری از روش شبیه‌سازی سنتی مبتنی بر رهیافت زمان‌بندی پیشامدها و همچنین شبیه‌سازی با استفاده از الگوریتم‌های بازگشتی با عنوان شبیه‌سازی سریع استفاده کرده‌ایم و نتایج شبیه‌سازی به دست آمده از برنامه‌های شبیه‌سازی با الگوریتم‌های بازگشتی و الگوریتم‌های شبیه‌سازی به شیوه سنتی و نتایج تحلیلی را مورد مقایسه قرار داده‌ایم و در پایان، زمان اجرای دو شبیه‌ساز، مورد بررسی و مقایسه قرار گرفته است. در فصل ۴ رفتار الگوریتم‌های موازی برای شبیه‌سازی صف‌های سری مورد بررسی قرار گرفته است و در فصل ۵ جمع‌بندی، نتیجه‌گیری و پیشنهاداتی برای تحقیق‌های آینده ارائه شده است.

فصل ۲

بررسی منابع و پیشینه موضوع

در فصل قبل به‌طور مختصر به مسائلی که در تحقیق حاضر مورد بررسی قرار گرفته‌اند اشاره شد، در این فصل قصد داریم تا پیشینه مختصری از مباحث بررسی شده در این تحقیق را ارائه دهیم. بخشی از تحقیق و پژوهش‌های انجام شده که مفید و در عین حال جالب بوده است در این فصل آورده شده است.

نظریه صف با رویکردی مبتنی بر مفاهیم آمار و احتمال، حل مسائل صف‌بندی، و همچنین تجزیه و تحلیل رفتار این دسته از سیستم‌ها را بر عهده دارد. نظریه صف‌بندی به‌منظور تهیه مدل‌هایی برای پیش‌بینی رفتار سیستم‌هایی تکامل یافته است که سعی دارند به درخواست‌های تصادفی سرویس دهند. از این‌رو مسائل اولیه‌ای که مطالعه شده‌اند مربوط به تراکم درخواست مکالمات تلفنی بوده است. محقق پیشگام این نظریه، ریاضیدان دانمارکی *ارلانگ*[1] بوده است، که در سال ۱۹۰۹، نظریه احتمال‌ها و مکالمات تلفنی را منتشر کرد. نخستین کارها در نظریه صف‌بندی نسبتا به کندی انجام می‌گرفت، اما از سال ۱۹۵۰ به بعد روند آن تغییر نموده و کارهای بسیاری در این زمینه انجام شده است.

[1] Erlang

یکی از موضوعات مهم در نظریه صف‌بندی، شبکه‌های صف‌بندی می‌باشد. این‌گونه مسائل از نقطه‌نظر کاربردپذیری زیاد آن‌ها در مدل‌بندی کامپیوتر و شبکه‌های مخابرات از اهمیت ویژه‌ای برخوردارند. از شبکه-های مهم صف‌بندی می‌توان به شبکه صف‌های سری اشاره کرد که در این تحقیق به بررسی آن پرداخته‌ایم.

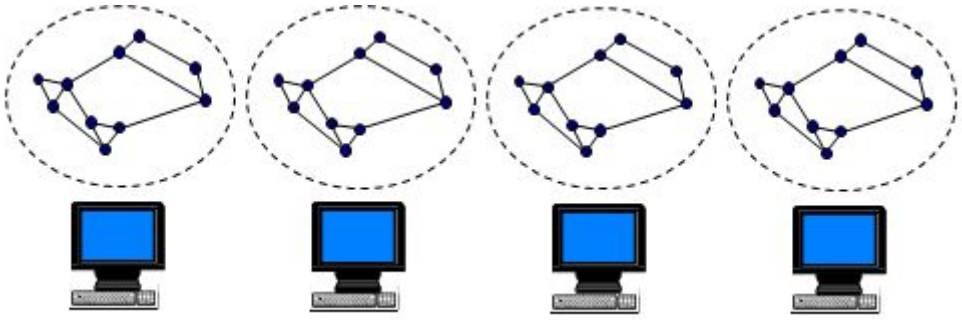
برای صف‌های سری مادام که محدویت‌های مربوط به ظرفیت بین ایستگاه‌ها وجود نداشته باشد و ورودی پواسون باشد نتایج را نسبتا به آسانی می‌توان به‌دست آورد. اما تحلیل برای صف‌های سری وقتی که محدودیت در ظرفیت ایستگاهی وجود داشته باشد بسیار پیچیده‌تر است و شبیه‌سازی می‌تواند به‌عنوان یک ابزار مهم برای محاسبه آماره‌های مختلف این‌گونه سیستم‌ها به‌کار رود[۱].

برای شبیه‌سازی سیستم‌های صف‌بندی می‌توان از روش *زمان‌بندی پیشامدها*[1] استفاده کرد، اما همان-طور که قبلا اشاره شد شبیه‌سازی به این روش زمان‌بر می‌باشد و به همین علت برای کاهش زمان اجرا باید تدبیری اندیشید. برای کاهش زمان شبیه‌سازی چن روشی مبتنی بر روابط بازگشتی ارائه داد[۹] که در فصل ۲ بررسی شده است.

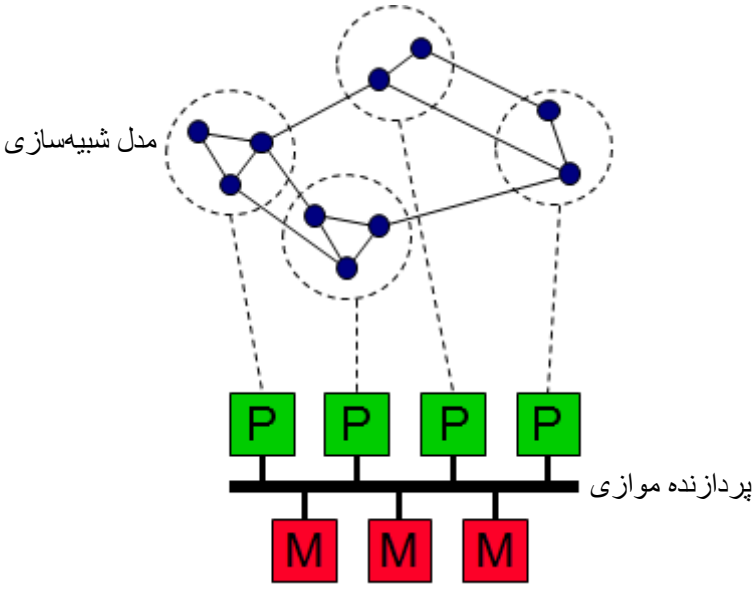
با وجودی که روش فوق تاحدی باعث کاهش در زمان اجرا می‌شود به علت تکرارهای بسیار که برای شبیه‌سازی داریم، در مواردی که کارایی الگوریتم سری به دلایل اقتصادی یا محدودیت‌های زمان اجرا کافی نیست از کامپیوترهایی با چند پردازنده کمک می‌گیریم.

به‌طورکلی، برای بهره‌برداری از توان پردازش موازی برای یک مساله شبیه‌سازی، دو روش مختلف وجود دارد: روش *پردازش تکراری موازی*[2] (شکل ۲-۱)، که در آن چند تکرار از مدل شبیه‌سازی یکسان را به صورت موازی اجرا می‌کند و *توازی مدل*[3] (شکل ۲-۲و ۲-۳)، که یک وظیفه را به چند زیروظیفه برای شبیه-سازی موازی و توزیع شده تجزیه می‌کند. برای مطالعه بیشتر می‌توان به منابع [۱۸-۲۲,۲۱-۳۷,۲۴] مراجعه کرد.

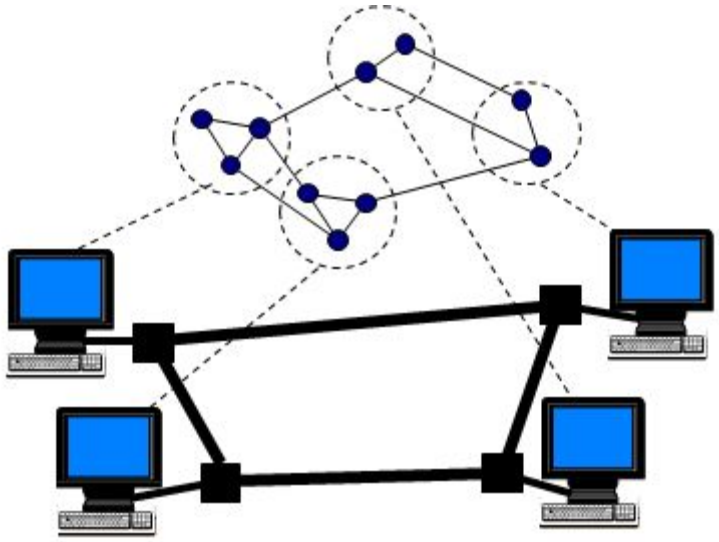
[1] Event schedule
[2] Parallel replicated processing
[3] Model parallelization



شکل ۲- ۱.پردازش تکراری موازی



شکل ۲- ۲.پردازش موازی



شکل ۲- ۳.پردازش توزیع‌شده

اجرای شبیه‌سازی تکراری موازی یا مدل موازی‌شده، از منظر سخت افزار پردازش مورد استفاده، دسته-بندی شده‌اند (به شکل‌های ۲-۱، ۲-۲، ۲-۳ مراجعه شود). شبیه‌سازی موازی قابلیت اجرا روی کامپیوتری را دارد که دارای چندین پردازشگر است که از حافظه مشترک بهره می‌گیرند یا روی تعدادی از پردازشگرهای مستقل که با مخابره اطلاعات هماهنگی لازم را ایجاد می‌کنند[۲۷]. به‌دست آوردن صحت اجرای موازی به همزمانی بین پردازنده‌ها نیاز دارد. شبیه‌سازی موازی در واقع به شبیه‌سازی‌های همزمان دقیق مربوط است که روی چند پردازنده به هم متصل اجرا می‌شود. همه پردازنده‌ها باهم سرویس می‌دهند تا به طور جمعی یک مجموعه کامل از مدل‌های کاربردی شبیه‌سازی شود[۳۷].

در زمینه شبیه‌سازی گسسته پیشامد موازی، روش کلاسیک این است که فضای حالت مدل به تعدادی زیر حالت تقسیم می‌شود تا به‌طور همزمان توسط پردازش‌های موازی شبیه‌سازی شود. این روش، تجزیه-پذیری فضای حالت مدل را نیاز دارد. متاسفانه در بسیاری موارد، یک مدل فقط تعداد محدودی از قابلیت تجزیه را در بر می‌گیرد. یک سیستم صف تک سرویس‌دهنده، فضای حالتی بسیار ساده دارد و اکثرا فقط تعدادی از متقاضیان به‌طور همزمان حضور دارند. به‌علاوه، حتی در شبکه‌های صف با فضای حالتی بسیار

پیچیده، میزان توازی، با تجزیه مدل فاصله‌ای، محدود می‌باشد[۲۷]. برای جزئیات بیش‌تر در زمینه شبیه-سازی موازی و توزیع شده خواننده علاقمند را به منابع [۴۰,۳۷,۳۵,۱۷-۱۴,۱۰] ارجاع می‌دهیم.

دسته دیگری از تکنیک‌های شبیه‌سازی موازی *قطعه‌بندی زمان*[1] است. در این روش هرپردازنده، برای شبیه‌سازی سیستم تحت بررسی، به یک بخش از کل زمان مورد توجه اختصاص داده شده است. برای مطالعه جزئیات بیش‌تر خواننده علاقمند را به [۲۴-۲۲,۷,۶] ارجاع می‌دهیم.

روش قطعه‌بندی زمان با روش *فضا-زمان*[2] ارائه شده توسط *کندی و شرمن*[3] مرتبط می‌باشد[۸]. ایده آن‌ها، اختصاص هر پردازنده به شبیه‌سازی بخشی از سیستم تحت مطالعه برای یک بخش از زمان مورد توجه می‌باشد. بنابراین روش فضا-زمان روشی از ترکیب شبیه‌سازی توزیع شده و قطعه‌بندی زمان می‌باشد[۶].

روش‌های شبیه‌سازی *زمان-موازی*[4] ایده‌ای مشابه قطعه‌بندی زمان را به‌کار می‌برد[۲۲]. زمان شبیه-سازی شده به تعدادی بخش تقسیم می‌شود و مسئولیت محاسبه تغییرات حالت درون هر بخش به یک پردازش موازی جداگانه اختصاص دارد. این تکنیک همچنان‌که درجه توازی توسط تجزیه‌پذیری فضای حالت مدل محدود نمی‌شود می‌تواند برای شبیه‌سازی سیستم‌های صف به‌کار برده شود[۲۷]. برای مطالعه جزئیات بیش‌تر خواننده علاقمند را به [۴۱,۳۳] ارجاع می‌دهیم.

درجه توازی الگوریتم‌ها یا پروتکل‌های تقسیم فضا توسط اندازه بخش‌های ممکن و طبیعت سیستم شبیه‌سازی شده محدود می‌شود. تعداد پردازنده‌هایی که می‌تواند استفاده شود غالبا خیلی کمتر از تعداد زیر-سیستم‌ها می‌باشد. الگوریتم‌های تقسیم زمان و تقسیم فضا-زمان، برای شبیه‌سازی امیدبخش‌تر به نظر می-رسند[۱۰].

در ادامه چند نمونه از کارهای انجام شده با استفاده از مطالب اخیر را به اختصار توضیح می‌دهیم :

[1] Time Segmentation
[2] Space-Time
[3] Chandy and Sherman
[4] Time-Parallel

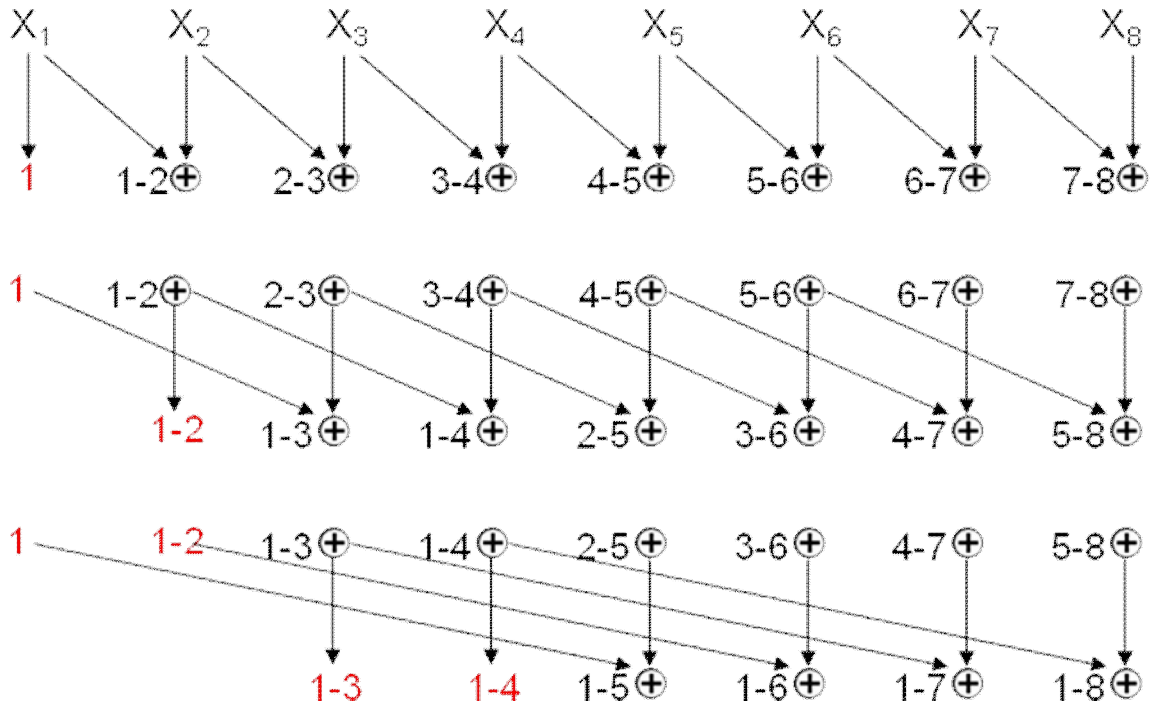
گرینبرگ، لوباچفسکی و میترانی[1]، محاسبات پیشوندی[2] موازی را برای محاسبه دینامیک‌های سیستم صف $G/G/1$ با استفاده از روابط بازگشتی استفاده کردند[۲۰,۱۹]. به‌علاوه، آن‌ها تکنیک‌هایی برای به‌کار بردن این روش با چندین نوع مختلف از شبکه‌های صف شامل صف‌های سری با بافر[3]های کراندار، شبکه‌های دوری بسته، و چندین نوع از شبکه‌های بدون دور از شبکه‌های صف‌های $G/G/1$ به‌کار بردند.

محاسبات پیشوندی، N حاصل‌ضرب اولیه از N متغیر $X_N,...,X_2,X_1$ را محاسبه می‌کند :

$$P_1 = X_1$$
$$P_2 = X_1 * X_2$$
$$P_3 = X_1 * X_2 * X_3$$
$$.$$
$$.$$
$$.$$
$$P_N = X_1 * X_2 * ... * X_N$$

که (*) یک عمل شرکت‌پذیر[4] است. این مجموعه از روابط می‌تواند به‌طور فشرده به‌صورت $P_i = P_{i-1} * X_i$، $i = 1,2,...,N$ بازنویسی شود که P_0 عنصر همانی می‌باشد. محاسبات پیشوندی به علت وجود الگوریتم‌های کارآمد برای اجرای این محاسبات روی یک کامپیوتر موازی مورد توجه هستند(به شکل ۲-۴ توجه کنید).

[1] Greenberg, Lubachevsky, and Mitrani
[2] Prefix
[3] Buffer
[4] Associative



شکل ۲- ۴. درخت عددی برای اجرای یک محاسبه پیشوندی موازی

برای به‌کار بردن این روش، مساله شبیه‌سازی به صورت زیر تعریف شده است: یک صف $G/G/1$ با نظم $FIFO$[1] با زمان‌های ورود $t_i, i \geq 1$ و زمان‌های سرویس $s_i, i \geq 1$ در نظر بگیرید. معیار مورد توجه، N زمان خروج اول D_i، $1 \leq i \leq N$ می‌باشد. فرض کنید که A_i زمان ورود کار i ام را مشخص می‌کند. آنگاه،

$$A_i = A_{i-1} + t_i = A_{i-n} + \sum_{m=i-n+1}^{i} t_m, \quad i > n \qquad (۱-۲)$$

و

[1] First in, First out

$$D_i = max(d_{i-1}, A_i) + S_i$$
$$= max[d_{i-n} + \sum_{k=i-n+1}^{i} S_k, \max_{i-n<j\leq i}(A_j + \sum_{l=j}^{i} S_l)], \quad i > n \tag{۲-۲}$$

با مرتب‌سازی دوباره اندیس‌ها، (۲-۱) و (۲-۲) به صورت زیر نوشته می‌شود :

$$A_{n+i} = A_n + A'_{n,i} \tag{۳-۲}$$

که در آن

$$A'_{n,i} = \sum_{m=n+1}^{n+i} t_m, i \geq 1$$

و

$$D_{n+i} = max(D_n + S'_{n+1,n+i}, D'_{n,n+i}) + S_i \tag{۴-۲}$$

که در آن

$$S'_{k,l} = \sum_{m=k}^{l} S_m$$

و

$$d'_{n,n+i} = \max_{n<j\leq n+i}(A_j + S'_{j,n+i})$$

فرض کنید P پردازنده داریم و پردازنده p_i زمان‌های خروج D_{n_i+1} و $D_{n_{i+1}}$ که $n_0 = 0$ و $n_p = N$ را تولید می‌کند. مشاهدات زیر چگونگی کار الگوریتم گرینبرگ، میترانی و لوباچفسکی را شرح می‌دهد : همه nها، که $n = n_0,...,n_{p-1}$ را در نظر بگیرید. (آ) $a'_{n,i}$ می‌تواند به طور همزمان محاسبه شود؛ (ب) اگر A_n و $A'_{n,i}$ معلوم باشد می‌توان A_{n+i} را به طور همزمان محاسبه کرد . (پ) اگر $A_j, j \geq 0$

معلوم باشد $d'_{n,n+1}$ را می‌توان به طور همزمان محاسبه کرد. و (ت) اگر D_n و $D_{n,n+i}$ معلوم باشد D_{n+i} را می‌توان به طور همزمان محاسبه کرد .

و به‌این ترتیب می‌توان روش محاسبه پیشوندی را برای محاسبه زمان‌های خروج استفاده کرد. برای مطالعه بیشتر در این زمینه می‌توان به منابع [۳۱,۳۰,۲۰,۱۹,۱۱] رجوع کرد.

ارماکوف و کریولین[1] برای شبیه‌سازی سیستم صف‌های سری با بافرهای نامتناهی، روش دیگری با به‌کار بردن روابط بازگشتی ارائه داد[۱۲] و همین روش را برای صف‌های سری با بافرهای متناهی تعمیم داد[۱۳]. این روش در فصل ۴ بررسی شده است.

چن روشی برای شبیه‌سازی موازی صف‌های $G/G/1$ و شبکه‌های خاص این صف‌ها شامل سیستم-های صف سری با بافرهای نامتناهی و متناهی با انسداد[2] یا ترک کردن[3] را ارائه داد[۱۰]. در این روش، فواصل طولانی‌ترین مسیر در گراف‌های جهت‌دار را برای محاسبه آماره‌های سیستم صف مورد استفاده قرار گرفته است.

برای نمایش طولانی‌ترین مسیر، یک سیستم صف $G/G/1$ را در نظر بگیرید که به صورت زیر عمل می‌کند. متقاضیان وارد سیستم می‌شوند و بر طبق نظم $FIFO$ سرویس می‌گیرند. یک یا چند متقاضی در هر لحظه می‌توانند وارد شوند و سرویس‌ها می‌توانند به صورت گروهی یا تکی، به شرطی که نظم $FIFO$ را حفظ کنند انجام شوند. هیچ محدودیت دیگری روی وابستگی مابین ورودها و سرویس‌ها وجود ندارد. داده سیستم (یا مجموعه ورودی) خانواده‌ای از متغیرهای تصادفی $\{A_k, S_k : k \geq 1\}$ است که A_k و S_k زمان‌های ورود و سرویس متقاضی k هستند. برای شبیه‌سازی، مناسب‌تر است تا زمان‌های ورود A_k را به‌جای زمان-های بین ورود $A_k - A_{k-1}$ پایه استفاده کنیم. $A_k = \sum_{i=1}^{k}(A_i - A_{i-1})$ می‌تواند از زمان‌های بین ورودها به

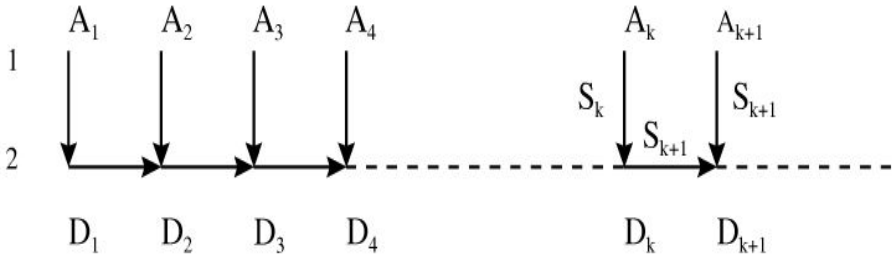
[1] Ermakov and Krivulin
[2] block
[3] Loss

وسیله روش پردازش موازی استاندارد تولید شده باشد. زمان خروج متقاضی $D_1 \le D_2 \le \ldots$ به وسیله داده سیستم از رابطه بازگشتی

$$D_k = \max\{A_k, D_{k-1}\} + S_k \ , \quad if \quad k \ge 1$$

که $A_0 = D_0 = 0$ مشخص می‌شود. همه متغیرهای کارایی سیستم (برای مثال طول صف‌ها، زمان‌های انتظار، دوره‌های شلوغ) توابعی از داده سیستم و زمان‌های خروج هستند. در نتیجه طراحی یک شبیه‌سازی سیستم معادل با توسعه یک الگوریتم برای تولید دنباله D_k از دنباله $\{A_k, S_k\}$ می‌باشد که هدف نیز همین است.

وسیله استفاده شده برای الگوریتم، گراف جهت‌دار نشان داده شده در شکل ۱ است که بسط (تکامل تدریجی) یا مسیر ساده سیستم $G/G/1$ را نمایش می‌دهد. زمان ورود $A_1, A_2, \ldots$ متقاضیان $1,2,\ldots$ با گره‌های در سطر ۱ و زمان‌های خروج متقاضیان $D_1, D_2, \ldots$ با گره‌های در سطر ۲ متناظر هستند. فرض کنید k_1 و k_2 به ترتیب گره‌های در سطر ۱ و ۲ از ستون kام را نشان دهند. K، پارامتر متقاضی، و ۱و۲ سطوح فاصله‌ای هستند. متناظر با هر کمان یک متغیر فاصله است: S_k فاصله روی k_1 و k_2، و روی کمان $(k-1)_2$ به k_2 است. فاصله روی یک مسیر از یک گره به دیگری مجموع فواصل روی کمان‌ها در طول مسیر است.



شکل ۲- ۵. گراف مسیر ساده $G/G/1$

فرض می‌کنیم $d(i_l, k_m)$ فاصله مسیر از گره i_l به گره k_m را مشخص کند. فواصل تک کمان موارد خاص هستند یعنی

$$d(k_1, k_2) = d((k-1)_1, k_2) = S_k$$

در زیر معادله بازگشتی برای زمان‌های خروج برحسب فواصل بلندترین مسیر در گراف ارائه شده است.

طبق قضیه ۱ در منبع [۱۰]، برای هر $i < i' < k$

$$D_k = \max\{D_i + d(i_2, k_2), C_{i,k}\}$$

که

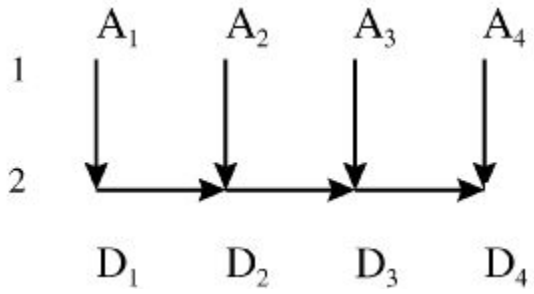
$$C_{i,k} = \max\{A_{i'} + d(i_1', k_2) : i' = i+1, \ldots, k\} \qquad (۲-۵)$$

همچنین

$$D_k = \max\{A_1 + d(1_1, k_2), C_{1,k}\} \qquad (۲-۶)$$

بنا به روابط قبل، مساله تولید D_k معادل با تولید فواصل بلندترین مسیر از گره‌های $i_1 (i \le k)$ به گره k_2 در گراف شکل ۲-۵ می‌باشد. مشخص کردن این فواصل آسان‌تر از آن است که ممکن است به نظر برسد زیرا معادله بازگشتی (۲-۵) همچنان‌که در ادامه نشان می‌دهیم به نحو مطلوبی، برای شبیه‌سازی موازی مناسب است.

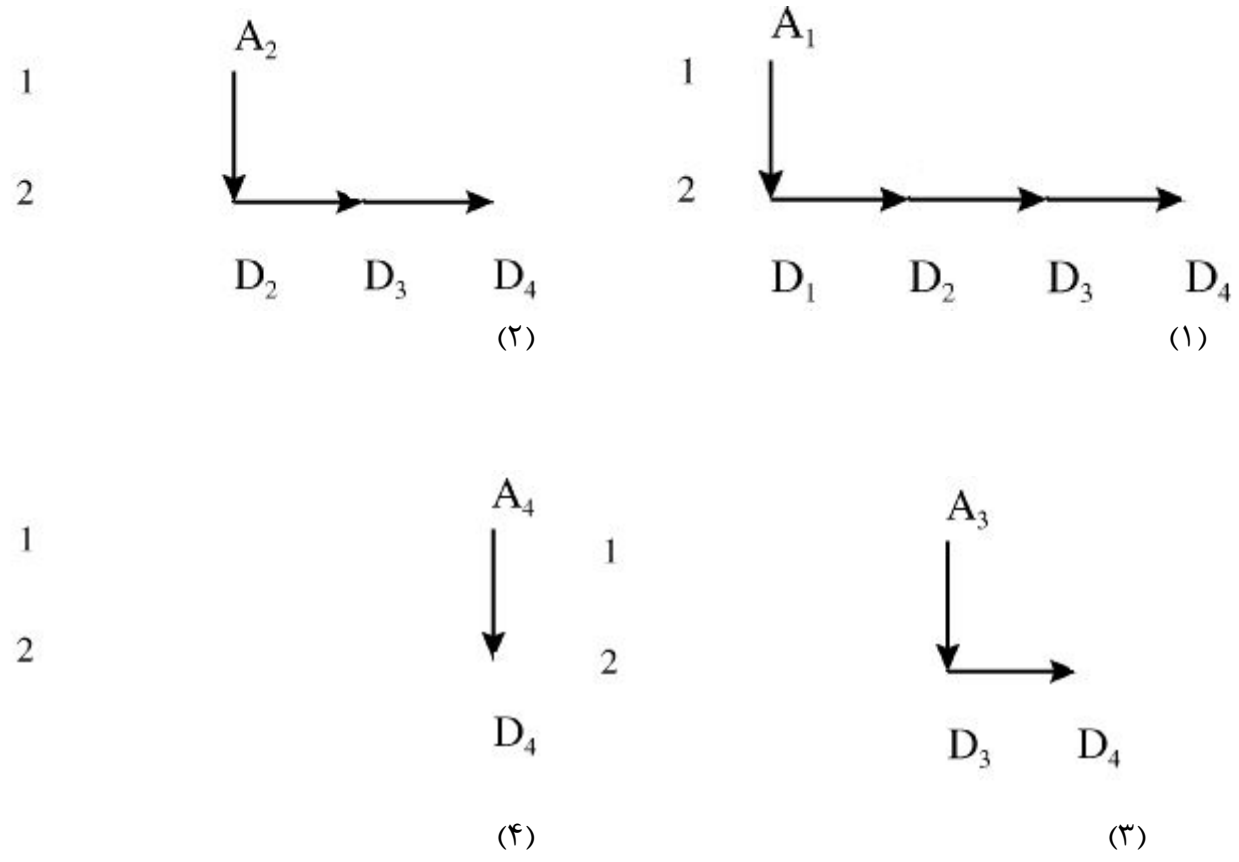
به عنوان مثال، شکل زیر را در نظر بگیرید



شکل ۲- ۶. مثالی با ۴ متقاضی برای شرح روابط (۲-۵) و (۲-۶)

$$D_4 = \max\{A_1 + d(1_1, 4_2), C_{1,4}\}$$

$$C_{1,4} = \max\{A_{i'} + d(i_1', 4_2) : i' = 2, \ldots, 4\}$$
$$= \max\{A_2 + d(2_1, 4_2), A_3 + d(3_1, 4_2), A_4 + d(4_1, 4_2)\}$$



شکل ۲- ۷. نمایش طولانی‌ترین مسیر برای یافتن D_k با ۴ متقاضی

از شکل‌های فوق و قضیه ذکر شده، می‌توان نتیجه گرفت که باید طولانی‌ترین مسیر را برای یافتن D_k در نظر گرفت.

فرض کنید هدف شبیه‌سازی سیستم صف $G/G/1$ با استفاده از P پردازنده روی یک کامپیوتر موازی MIMD(SIMD) است. برای راحتی، فرض کنید که $P = 2^r$ و تعداد متقاضیانی که شبیه‌سازی می-