

به نام خدا

گزارش پروژه اول درس شبیه سازی



استاد درس

دکتر بابک حجی

دستیار آموزشی

آقای حمیدرضا خواجه ارزانی

الهه رحمتی ۹۵۱۰۳۹۲۳

فهرست

۳ توضیح کد پروژه
۶ مفروضات
۶ Event notices و پیشامدها
۷ Trace اکسل
۸ فعالیت‌ها
۸ موجودیت‌ها
۸ Delay
۸ ویژگی‌های موجودیت‌ها
۸ متغیر حالت سیستم
۹ دیاگرام پیشامدها
۱۰ سوالات
۱۰ سوال ۱
۱۱ سوال ۲
۱۲ سوال ۳
۱۵ سوال ۴
۱۶ سوال ۵



توضیح کد پروژه

کد پروژه با زبان پایتون نوشته شده است.

خطوط اولیه برای import کردن کتابخانه‌های مورد استفاده در کد است. کتابخانه‌هایی برای ریاضیات، آمار و نوشتن در اکسل و ...

سطر بعدی از کاربر سیاست درخواستی را می‌گیرد که بعداً در بدنه‌ی اصلی کد تابع مربوط به آن سیاست را صدا کند. دستورات بعدی مربوط به ایجاد اکسل Trace و پر کردن شیت‌ها و سلول‌های آن است. اکسل Trace شامل دو شیت است. اولین شیت به نام "trace" است که توضیحات آن در بخش مربوطه آورده شده است. شیت دوم به نام "statistic" است که در آن آماره‌های جمع شده در انتهای هر دور ۱۰۰۰ ثانیه‌ای نوشته شده است. ستون اول آن ردیف است. ۵ ستون بعدی بهره‌وری سرورها به ترتیب از سرور ۱ تا ۵ و ۵ ستون بعدی مجموع زمان بیکاری سرورها به ترتیب از سرور ۱ تا ۵ است. در انتهای ۷۰۰ سطر دو سطر آخر میانگین و انحراف معیار هر ستون را نشان می‌دهد.

بخش بعدی مربوط به تعریف متغیرهاست که در این کد دو دسته بندی کلی دارند. متغیرهایی که پسوند statistic دارند متغیرهایی هستند که بعد از گذشت زمان سرد مورد استفاده قرار می‌گیرند و سایر متغیرها از همان ابتدا شروع برنامه مورد استفاده قرار می‌گیرند.

متغیرها:

Dismiss: مجموع پیام‌های از دست رفته در کل ۱۰۰۰ ثانیه

Served: مجموع پیام‌های خدمت رسانی شده توسط سرورها در ۱۰۰۰ ثانیه

Received: مجموع پیام‌های وارد شده به سیستم در ۱۰۰۰ ثانیه

FEL: لیستی از رویدادهای پیش رو در هر لحظه که مقادیر اولیه اش ['E', ۱۰۰۰], ['S', ۰] هستند.

Clock: ساعت شبیه سازی است که با توجه به لیست مرتب شده FEL تغییر می‌کند.

IdleServer: متغیری از نوع لیست که در هر لحظه اطلاعاتی از سرورهای خالی می‌دهد. این اطلاعات به ترتیب، آیدی سرور، آخرین زمانی بیکاری سرور تا آن لحظه و مجموع زمان‌های بیکاری سرور تا آن لحظه هستند.

Idle_ID: متغیری کمکی که در آن آیدی سرورهای خالی در هر لحظه ذخیره می‌شود. این متغیر به جای متغیر بالا و در مواقعی که اطلاعات سرور خالی را لازم نداریم مورد استفاده قرار می‌گیرد.

Total_Idle: متغیری از نوع لیست که مجموع زمان بیکاری هر سرور را از ابتدای شبیه‌سازی تا لحظه‌ای (Clock) که در آن قرار داریم ذخیره می‌کند. خانه‌ها به ترتیب به سرور ۱ تا ۵ تعلق دارند. یعنی Total_Idle[۰] اطلاعات مربوط به سرور ۱ را ذخیره می‌کند.

ServerState: یک لیست دو بعدی است که سطر اول آن حالت سرورها (۱ در هنگام مشغول بودن و ۰ در حالت آزاد بودن) و سطر دوم آن زمان پایان سرویس‌دهی را نشان می‌دهد. به این صورت که اگر سرور خالی باشد زمان پایان خدمت‌رسانی قبلی و اگر سرور مشغول باشد زمان پایان خدمتی که در حال انجام آن است را نشان می‌دهد.

Total_Served: لیستی است که به ترتیب تعداد پیام‌هایی که هر سرور تا آن لحظه سرویس داده است را از ابتدای شبیه‌سازی نشان می‌دهد.

X: لیستی از مقادیر محور X نمودار تشخیص زمان سرد(زمان)
 Y: لیستی از مقادیر محور Y نمودار تشخیص زمان سرد(تعداد پیام‌های خدمت رسانی شده توسط سرورها/ تعداد پیام‌های ورودی به سیستم)

Message_ID: متغیری که به هر پیام یک آیدی می‌دهد تا بتوان ورود و خروج آن را دنبال کرد.
 Row: متغیری که شماره ردیف‌های ستون اول اکسل Trace را تولید می‌کند.
 Loop: شمارنده حلقه اصلی کد که ۷۰۰ بار برنامه را اجرا می‌کند.

متغیرهایی که بعد از زمان سرد مورد استفاده قرار می‌گیرند:
 Dismiss_Statistic: مجموع پیام‌های از دست رفته بعد از گذشت زمان سرد
 Served_Statistic: مجموع پیام‌های خدمت رسانی شده توسط سرورها بعد از گذشت زمان سرد
 Recieved_Statistic: مجموع پیام‌های وارد شده به سیستم بعد از گذشت زمان سرد
 Statistics: لیستی که در هر دور آماره‌های لازم را جمع‌آوری می‌کند؛ ۱. بهره‌وری هر سرور و ۲. مجموع زمان بیکاری هر سرور

Total_Idle_Statistic: متغیری از نوع لیست که مجموع زمان بیکاری هر سرور را بعد از گذشت زمان سرد تا لحظه‌ای(Clock) که در آن قرار داریم ذخیره می‌کند. خانه‌ها به ترتیب به سرور ۱ تا ۵ تعلق دارند. یعنی
 [۰] Total_Idle_Statistic اطلاعات مربوط به سرور ۱ را ذخیره می‌کند.
 Total_Served_Statistic: لیستی است که به ترتیب تعداد پیام‌هایی که هر سرور تا آن لحظه سرویس داده است را از بعد از گذشت زمان سرد نشان می‌دهد.
 Last_Update_Clock: آخرین زمان به روز رسانی متغیرهای مربوط به بیکاری
 Count: متغیری کمکی برای تشخیص تعداد رویدادهایی که بعد از گذشت زمان سرد رخ داده است.
 در ادامه توابعی که در بدنه‌ی اصلی کد مورد استفاده قرار گرفته تعریف شده‌است.

تابع Sort:

این تابع FEL را بر اساس زمان و از کوچک به بزرگ مرتب می‌کند.

تابع Interarrival_Time_Generator:

این تابع زمان‌های بین ورود را تولید می‌کند. سپس رویداد ورود را تولید کرده و به FEL اضافه می‌کند.

تابع IdleServer_Update:

در این تابع متغیر IdleServer آپدیت می‌شود و در هر لحظه اطلاعاتی از سرورهای خالی را به دست می‌دهد.

تابع Total_Idle_Update:

متغیر Total_Idle و Total_Idle_Statistic را آپدیت می‌کند.

تابع `Service_Time_Generator`:

ورودی این تابع شماره سروری است که توسط سیاست انتخابی، انتخاب شده است. در این تابع زمان سرویس‌دهی با توجه به سرور انتخابی تولید می‌شود و سپس رویداد خروج تولید شده و به `FEL` اضافه می‌شود.

علاوه بر آن برخی متغیرها از جمله `ServerState`، `Total_Served_Statistic` و `Total_Served` آپدیت می‌شوند.

توابع مربوط به سیاست‌ها:

تابع `RR_Policy`:

این تابع ابتدا یک عدد رندم بین ۱ و ۱۰ تولید کرده و سپس آن را تبدیل به یک عدد رندم بین ۰ و تعداد سرور های خالی، می‌کند. سپس با توجه به عدد تولید شده یکی از سرورهای خالی را انتخاب کرده و آن را به عنوان ورودی به تابع `Service_Time_Generator` می‌دهد.

تابع `SIR_Policy`:

در این تابع ابتدا لیست `IdleServer` بر اساس آخرین زمان بیکاری از کوچک به بزرگ مرتب شده و سپس شماره سرور با کمترین زمان آخرین بیکاری به تابع `Service_Time_Generator` داده می‌شود.

تابع `LIR_Policy`:

در این تابع ابتدا لیست `IdleServer` بر اساس آخرین زمان بیکاری از بزرگ به کوچک مرتب شده و سپس شماره سرور با بیشترین زمان آخرین بیکاری به تابع `Service_Time_Generator` داده می‌شود.

تابع `LTIR_Policy`:

در این تابع ابتدا لیست `IdleServer` بر اساس مجموع زمان بیکاری از بزرگ به کوچک مرتب شده و سپس شماره سرور با بیشترین مجموع زمان بیکاری به تابع `Service_Time_Generator` داده می‌شود.

بدنه‌ی اصلی کد:

این بخش از دو حلقه تشکیل شده که حلقه بیرونی عملیات را ۷۰۰ بار تکرار می‌کند و حلقه‌ی داخلی عملیات را تا رسیدن به رویداد `['E', ۱۰۰۰]` ادامه می‌دهد.

در خطوط اولیه متغیرها در هر دور از حلقه‌ی بزرگتر صفر می‌شوند.

حلقه‌ی دوم:

در ابتدای هر دور از این حلقه متغیرهای مربوط به سرورهای خالی و مجموع زمان بیکاری آپدیت می‌شوند.

در ادامه با توجه به اینکه چه رویدادی در حال وقوع است اقداماتی انجام می‌شود:



رویداد شروع (S): زمان ورود بعدی تولید می‌شود.

رویداد خروج (D): حالت سرورها آپدیت می‌شود.

رویداد پایان (E): برای آخرین بار متغیر مجموع زمان بیکاری آپدیت می‌شود و حلقه به پایان می‌رسد.

رویداد ورود (A): اگر سروری خالی وجود داشته باشد تابع سیاست مربوطه صدا زده می‌شود و متغیرهای لازم آپدیت می‌شوند. در غیر این صورت به تعداد از دست رفته‌ها اضافه می‌شود. در انتها تابع تولید زمان بین ورود (در هر دو حالت) صدا زده می‌شود.

مراحل بعدی آپدیت کردن FEL، پر کردن شیت Trace از اکسل Trace است و پر کردن لیست‌های x و y برای کشیدن نمودار است.

بعد از خروج از هر حلقه‌ی ۱۰۰۰ ثانیه‌ای لیست مربوط به آماره‌ها آپدیت شده و در شیت Statistic مربوط به اکسل Trace نوشته می‌شوند.

خارج از حلقه‌ها و در خطوط انتهایی کد دستورات مربوط به کشیدن نمودار زمان سرد، محاسبه میانگین و انحراف معیار آماره‌های شیت Statistic و ذخیره کردن اکسل Trace نوشته شده است.

مفروضات

- در انتخاب سرور توسط سیاست‌ها (به جز RR) در صورتی که دو یا چند سرور شرایط یکسانی برای انتخاب داشته باشند سروری که شماره کوچکتری دارد (آیدی سرور) انتخاب می‌شود.
- شبیه‌سازی با رویداد (S, ۰) شروع می‌شود.
- در تمامی بخش‌ها قسمت امتیازی انجام شده است.
- زمانی که پیامی را به سروری تخصیص می‌دهیم به تعداد سرویس‌های آن سرور و سیستم اضافه می‌شود به عبارتی در هر لحظه، تعداد پیام‌هایی که سرویس گرفته اند و یا در حال سرویس‌گیری هستند محاسبه می‌شوند.
- پوشه‌ی حاوی جداول Trace برای هر سیاست تحت عنوان "Trace"، اکسلی شامل آماره‌های ۷۰۰ دور که روی آن تحلیل انجام شده تحت عنوان "Statistic" و کد پروژه به پیوست آمده است.
- داده‌های تحلیل شده و آماره‌های موجود در Trace و Statistic همگی مربوط به بعد از زمان سرد هستند.

پیشامدها و Event notices

- پیشامد ورود پیام
(پیام ID، زمان ورود، A)
- پیشامد پایان سرویس دهی یکی از ۵ سرور

(پیام ID, سرور تخصیص یافته به پیام, زمان خروج, A)

• پیشامد آغاز شبیه‌سازی

(S, 0)

• پیشامد پایان شبیه‌سازی

(E, 1000)

اکسل Trace

این اکسل که در پیوست در پوشه‌ای تحت عنوان Trace آمده است در هر بار اجرای کد در لپ‌تاپ شما ایجاد می‌شود. این اکسل شامل دو شیت است. شیت Trace و شیت Statistic. در اینجا عناصر شیت Trace را توضیح می‌دهیم و شیت دوم در قسمت مربوطه توضیح داده خواهد شد.

Clock: زمان وقوع رویداد را نشان می‌دهد.

Event: رویدادی که رخ داده است را مشخص می‌کند.

Message ID: آیدی پیامی که وارد یا خارج شده است را نشان می‌دهد.

FEL: لیستی از رویدادهای پیش رو را به صورت مرتب شده نشان می‌دهد.

در ۵ ستون بعدی (F تا G) به ترتیب حالت سرورهای ۱ تا ۵ نوشته شده است که در صورت مشغول بودن با ۱ و در صورت آزاد بودن با ۰ نمایش داده می‌شود. این مقدار آپدیت شده است و حالت سرورها را بعد از وقوع رویداد این زمان نشان می‌دهد. حالت سرورها یک لحظه بعد از وقوع رویداد

در ۵ ستون بعدی (K تا O) به ترتیب آخرین بیکاری سرورهای ۱ تا ۵ نوشته شده است. توجه کنید که در این حالت تنها سرورهایی که تا یک لحظه قبل از وقوع رویداد خالی بودند مقدار می‌گیرند. برای اینکه ببینید سرور به درستی انتخاب شده است یا خیر.

در ۵ ستون بعدی (P تا T) به ترتیب مجموع بیکاری سرورهای ۱ تا ۵ نوشته شده است. توجه کنید که در این حالت نیز تنها سرورهایی که تا یک لحظه قبل از وقوع رویداد خالی بودند مقدار می‌گیرند.

در ادامه مقادیر آماره‌های مربوط به هر سرور نشان داده می‌شود:

ستون‌های (U تا Z) مجموع زمانی بیکاری هر سرور را به ترتیب از ۱ تا ۵، بعد از گذشت زمان سرد نشان می‌دهد.

۵ ستون بعدی (Z تا AD) بهره‌وری هر سرور را به ترتیب از ۱ تا ۵، بعد از گذشت زمان سرد نشان می‌دهد.

و سه ستون آخر به ترتیب تعداد پیام‌های از دست‌رفته، تعداد پیام‌های سرویس داده شده و تعداد پیام‌های دریافت شده در سیستم را از لحظه‌ی شروع نشان می‌دهد.

فعالیت‌ها

- زمان‌های بین ورود
- زمان‌های سرویس‌دهی

موجودیت‌ها

- سرورها
- پیام‌ها

Delay

با توجه به اینکه سیستم شرح داده شده صف نداشت و در حالت مشغول بودن همه‌ی سرورها، پیام از دست می‌رفت این سیستم زمانی تحت عنوان Delay ندارد.

ویژگی‌های موجودیت‌ها

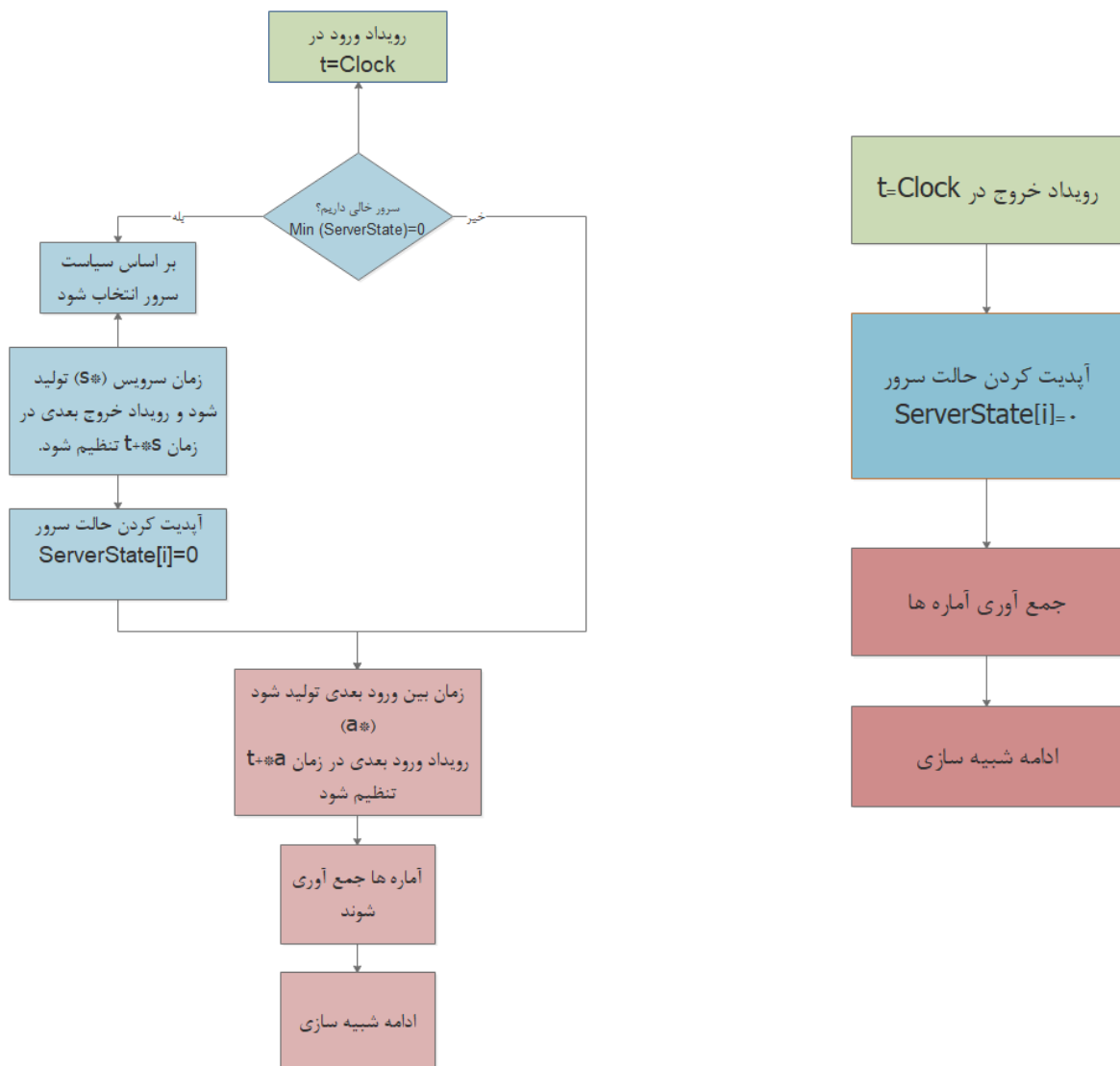
۱. پیام‌ها
 - ID
۲. سرورها
 - آخرین زمان بیکاری
 - تعداد پیام‌هایی که سرویس دادند
 - مجموع زمان‌های بیکاری

متغیر حالت سیستم

SystemState: این آرایه‌ی ۵ تایی حالت سرورها (مشغول = ۱ و آزاد = ۰) را در هر لحظه t به دست می‌دهد.



دیاگرام پیشامدها



سوالیات

سوال ۱

برای بررسی این سوال معیار عادلانه بودن مهم است. اگر عدالت را به این معنی که هر کس در جایگاه خودش قرار بگیرد تعریف کنیم فاکتورهای زیادی دارد که باید در نظر گرفت. برای مثال مهارت فرد یکی از این فاکتورهاست. سرور یک با توجه به نرخ خدمت‌دهی بالاتر (با این فرض که ماهیت کار سرورها یکسان است) احتمالاً مهارت بیشتری دارد. پس جواب به این سوال سخت است ولی اگر قرار باشد هیچ کدام این فاکتورها را تاثیر ندهیم می‌توان این مسئله را بررسی نمود.

در نتیجه به طور کلی و بدون در نظر گرفتن همه‌ی این موارد به نظر من سیاستی که بر اساس بیشترین مجموع زمان بیکاری انتخاب می‌کند عادلانه تر است چرا که سعی دارد از همه‌ی افراد به یک اندازه استفاده کند و جلوی این موضوع را که سروری بیشتر از سایرین استراحت کند را می‌گیرد.

در مورد سیاست رندم حس می‌کنم که روش عادلانه‌ای نیست چون کارکرد افراد را اصلاً در نظر نمی‌گیرد و کاملاً شانسی و با شانس برابر تخصیص را انجام می‌دهد.

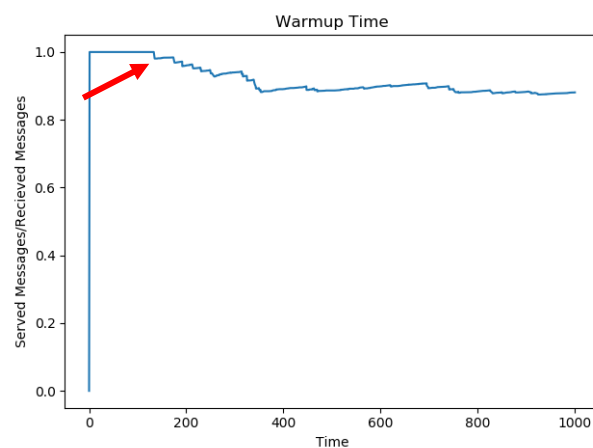
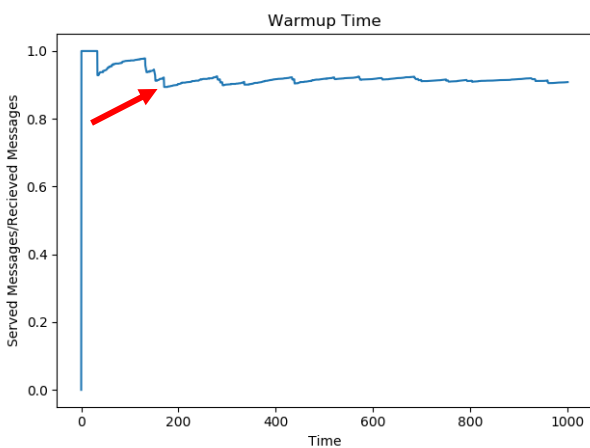
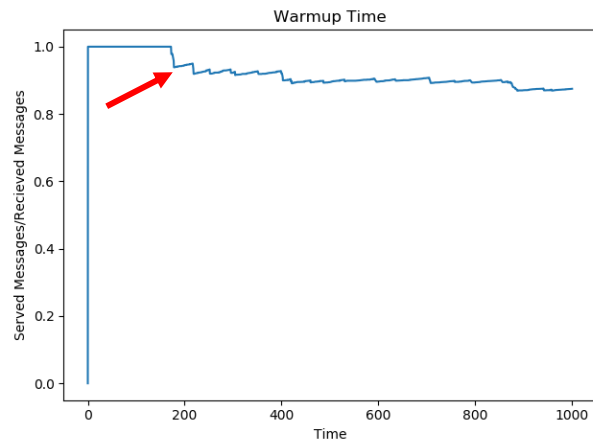
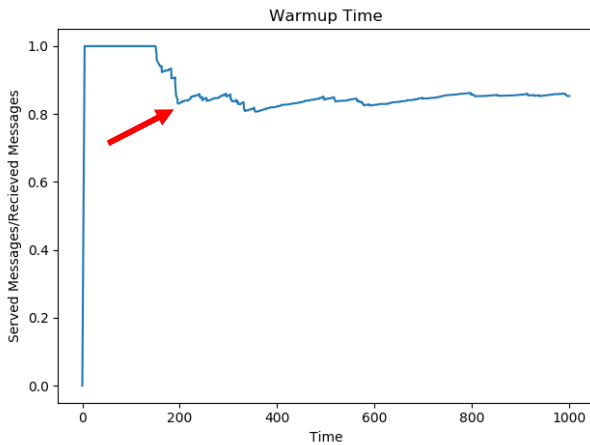
اما در مورد دو سیاست باقی مانده که بر اساس آخرین زمان بیکاری تخصیص را انجام می‌دهند، حالت اول که بر اساس ماکزیمم این مقدار (آخرین زمان بیکاری) است می‌تواند منطقی باشد چون فرد نسبت به دیگران در آخرین بیکاری خود بیشتر استراحت کرده است. اما از نظر من حالت عکس که بر اساس مینیمم این مقدار (آخرین زمان بیکاری) است منطقی نیست.

مسئله‌ای که وجود دارد در نظر گرفتن سایر فاکتورها اعم از نرخ خدمت‌دهی سرورهاست. اما اگر کاملاً بدون بررسی و استدلال و کاملاً حسی بررسی شود به نظر من سیاست آخر که بر اساس بیشترین مجموع زمان بیکاری است منطقی تر و عادلانه تر است.

سوال ۲

زمان سرد سیستم:

برای محاسبه‌ی زمان سرد سیستم ابتدا نمودار (بهره‌وری سیستم- زمان) برای ۱۰۰۰ ثانیه اول رسم شد. منظور از بهره‌وری سیستم نسبت پیام‌های سرویس داده‌شده، به کل پیام‌های دریافتی است. با توجه به نمودارهای زیر مشاهده می‌شود که زمان سرد سیستم بین ۱۰۰ تا ۲۰۰ ثانیه متغیر است. در کد پروژه این زمان به صورت تقریبی برابر با $t=150s$ در نظر گرفته شد.



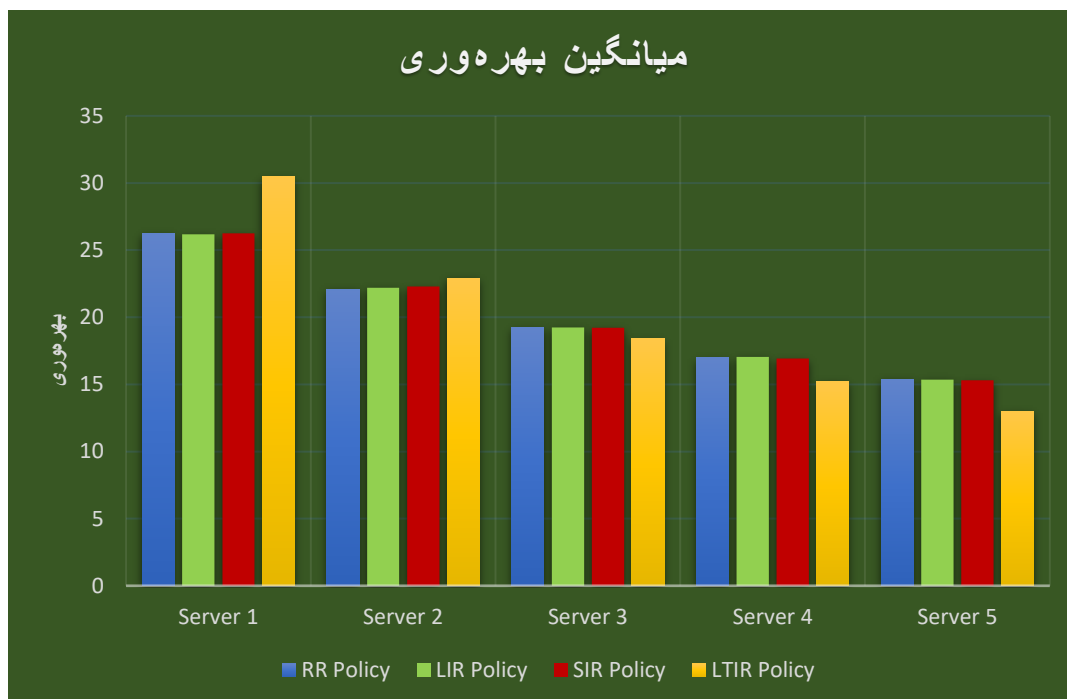
سوال ۳

آماره های گفته شده در صورت سوال در انتهای هر ۱۰۰۰ ثانیه جمع آوری شده و سپس بعد از ۷۰۰ بار برای سیاست و هر سرور به صورت جداگانه ، میانگین و انحراف معیار آنها محاسبه شد. (در هر بار اجرای کد پروژه در اکسل Trace شیتی به نام Statistic ایجاد می شود که این مقادیر در آن ذخیره می شود.) داده ها به شرح زیر می باشد:

(فایل مربوط به داده های جداول زیر تحت عنوان "Statistics" پیوست شده است.)

بهره وری (نسبت تعداد پیام هایی که هر سرور پاسخ داده به کل پیام های سرویس داده شده):

Policy	Ave/SD	بهره وری				
		Server 1	Server 2	Server 3	Server 4	Server 5
RR Policy	Average	26.25039	22.09917	19.27365	16.98611	15.39068
	SD	1.968425	1.875752	1.84228	1.763608	1.703363
LIR Policy	Average	26.17651	22.18849	19.23787	17.0434	15.35372
	SD	1.726205	1.706905	1.665788	1.52767	1.566923
SIR Policy	Average	26.25518	22.2906	19.20782	16.93037	15.31603
	SD	2.762429	2.650917	2.393879	2.218125	2.114749
LTIR Policy	Average	30.47507	22.90673	18.40719	15.21731	12.9937
	SD	2.829314	2.636025	2.396905	2.06418	1.971315



با توجه به نمودار بالا مشاهده می‌شود که به طور کلی میانگین بهره‌وری سرور ۱ در تمامی سیاست‌ها از سایر سرورها بیشتر است و بعد از آن به ترتیب سرورهای ۴،۳،۲ و ۵ قرار می‌گیرند.

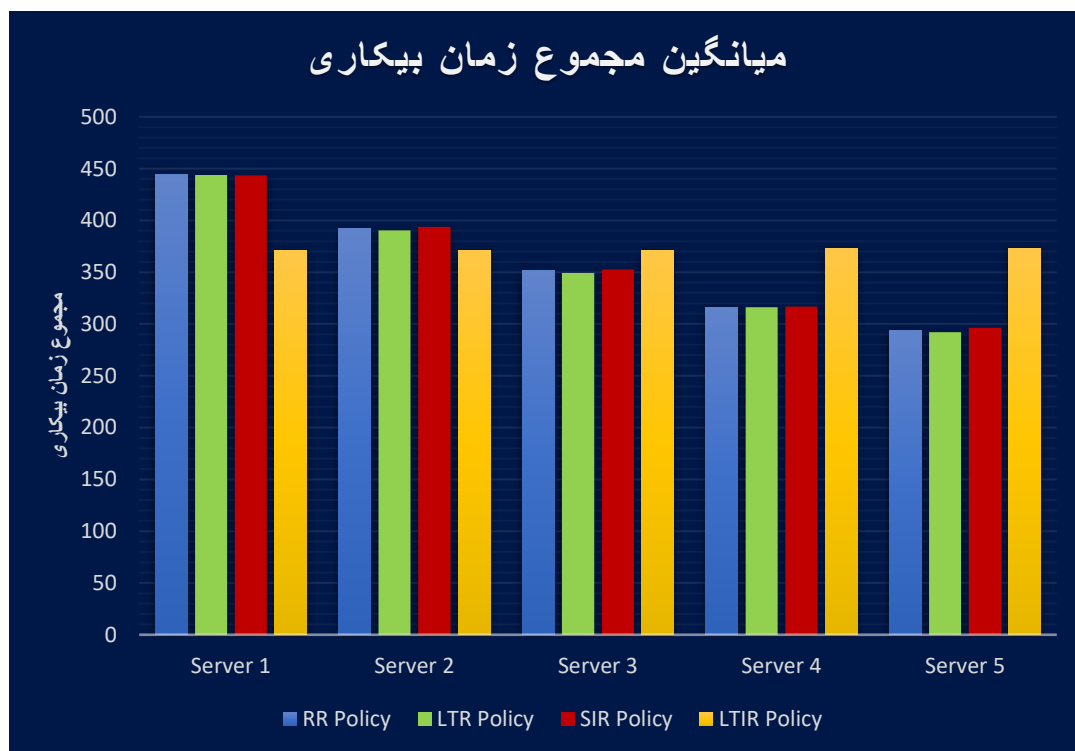
اگر میانگین توزیع زمان خدمت‌دهی هر سرور را در نظر بگیریم این امر قابل توجیه است چرا که توزیع زمان‌های بین ورود ثابت بوده و میانگینش برابر ۲،۸ ثانیه است. بنابراین تعداد ورودی‌ها تقریباً ثابت خواهد بود. حال با توجه به اینکه میانگین توزیع زمان خدمت‌دهی سرور ۱ از همه کمتر است می‌توان درک کرد که تعداد پیام‌هایی که به این سرور تخصیص داده می‌شود به طور کلی بیشتر است و در نتیجه میانگین بهره‌وری این سرور نیز بیشتر خواهد شد. با همین استدلال برای دیگر سرورها نیز این اتفاق توجیه پذیر است چرا که میانگین توزیع خدمت‌دهی سرورها (۶،۸،۱۰،۱۲،۱۴) به صورت صعودی افزایش داشت.

نکته ی قابل توجه دیگر این است که برای سرور ۱ در LTIR Policy اختلاف زیادی با سایر سیاست‌ها دیده می‌شود. به عبارتی در این سیاست به سرور ۱ بیشترین پیام ممکن تخصیص داده شده است. این امر نشان می‌دهد که در این سیاست در هر لحظه مجموع بیکاری سرور ۱ به طور کلی از سایر سرورها بیشتر بوده است.

نکته ی دیگری که قابل توجه است شباهت نمودارهای سه سیاست اول با هم در هر سرور است که تقریباً در یک مقدار هستند و این نتیجه نیز تا حد خوبی تحت تاثیر میانگین توزیع خدمت‌دهی است. چرا که به طور کلی مدت زمانی که طول می‌کشد تا سرور ۱ خالی شود کمتر از سایر سرورهاست. پس با در نظر داشتن اینکه نرخ ورود با نرخ خدمت‌دهی سرور‌ها اختلاف زیادی دارد، همین امر باعث می‌شود که در لحظه ی ورود پیام جدید در اغلب اوقات سروری که میانگین بالاتر دارد پر باشد و این فارغ از این است که در چه سیاستی هستیم. در نتیجه تعداد پیام‌هایی که به این سرور تخصیص داده می‌شود کمتر است و نتیجه، نمودار بالا خواهد بود.

مجموع زمانی بیکاری:

Policy	Ave/SD	مجموع بیکاری				
		Server 1	Server 2	Server 3	Server 4	Server 5
RR Policy	Average	444.0233	392.1873	351.7399	315.9555	293.95
	SD	39.96757	41.89157	44.23445	44.55949	45.44353
LIR Policy	Average	443.8901	390.3038	349.4173	316.3291	292.0473
	SD	34.65272	36.39311	38.81514	39.71876	39.23231
SIR Policy	Average	443.5226	393.8818	352.7496	316.8464	296.3714
	SD	51.75615	56.80749	61.49582	63.56942	66.98788
LTIR Policy	Average	370.8644	371.1743	371.1933	372.4677	372.8618
	SD	29.64332	30.22035	31.51809	32.23329	34.46977



در نمودار بالا مشاهده می‌شود که در سه سیاست اول میانگین مجموع زمان بیکاری در سرور ۱ بیشتر است و سپس به ترتیب در سرور های ۲ و ۳ و ۴ و ۵ کم می‌شود. به عبارتی در سه سیاست اول این مقدار در سرورها روند نزولی دارد. این امر نیز با توجه به نرخ ورود که تقریباً ثابت است توجیه پذیر است. با اینکه نرخ بهره‌وری سرور یک و میانگین زمان خدمت دهی‌اش بیشتر است ولی به خاطر محدود بودن تعداد پیام‌های ورودی، سرور ۱ پس از پایان خدمتش زمان‌های بیشتری را در انتظار ورود پیام جدید می‌ماند و همین باعث افزایش میانگین زمان خدمت‌دهی این سرور در مقایسه با سایر سرورها می‌شود. اما در سیاست آخر که بر اساس بیشترین مجموع زمان بیکاری است کاهش محسوسی را در این میانگین می‌بینیم که نشان می‌دهد در این سیاست

بیشترین تعداد پیام ممکن به این سرور تخصیص داده شده است و قاعدتا باید بهره‌وری بالایی داشته باشد که با نمودار قبلی مطابقت دارد.

نکته‌ی دیگر این است که در این نمودار نیز سه سیاست اول در هر سرور رفتاری مشابه نشان دادند که همین امر بر استدلالی که در بحث قبلی مطرح شد، صحت می‌گذارد.

با دیدی دقیق‌تر نسبت به مقادیر انحراف معیار بدست آمده از هر یک از سیاست‌های مورد بحث، به نظر می‌آید که سیاست SIR با اختلاف، بیشترین مقادیر انحراف معیار را در هر ۴ سرور به خود اختصاص داده است. این امر نشان می‌دهد که اتخاذ این سیاست به عنوان سیاست مطلوب سیستم، ممکن است سبب شود تا نتایجی که در این شبیه‌سازی بدست آوردیم خیلی قابل اتکا نباشد و نتایج واقعی بدست آمده پس از اتخاذ این سیاست، با نتایج بدست آمده‌ی ما در این پروژه متفاوت باشد. لذا این عدم اطمینان در خروجی بدست آمده از این سیاست نمی‌تواند اهداف ما را دنبال کند.

در انتها با توجه به تحلیل‌های بالا اگر بخواهیم دوباره پاسخی برای سوال ۱ داشته باشیم، می‌توانیم به طور کلی بگوییم که میانگین زمان بیکاری در سیاست آخر در اکثر سرورها کمتر از سایر سیاست‌هاست و اگر از بعد مدیری نگاه کنیم که به دنبال کم کردن زمان بیکاری و افزایش بهره‌وری سیستم است، احتمالا این سیاست مطلوبیت بیشتری دارد. البته که باز تاکید می‌شود که این انتخاب کاملا به شرایط و هدف شبیه‌سازی بستگی دارد.

سوال ۴

با توجه به اینکه در سیاست LIR سروری انتخاب می‌شود که از آخرین بیکاری اش زمان بیشتری گذشته باشد، به نظر می‌رسد که استفاده از این سیاست در زمانی که کار سرورها خسته کننده است منطقی تر باشد چرا که سرور زمان بیشتری را استراحت کرده است. برای مثال اگر کار سرورها مدام با کامپیوتر باشد برای جلوگیری از گردن درد یا چشم درد سرورها می‌توان از این سیاست استفاده کرد. اگرچه این موضوع کاملا بستگی به هدف سیستم دارد و اینکه مدیر چه هدفی را دنبال می‌کند، زیرا بهره‌وری سیستم بر اساس هدف تعریف می‌شود. برای مثال این سیاست، در نرخ خدمت‌دهی‌های متفاوت، بیشتر اوقات کار را به سرورهای ماهرتر واگذار می‌کند و می‌تواند به کم کردن تعداد از دست رفته‌ها کمک کند که در صورتی که هدف افزایش تعداد پیام‌های پاسخ‌داده شده باشد، می‌تواند موثر باشد. به عبارتی اگر اختلاف در نرخ خدمت‌دهی به علت مهارت باشد و کار همه‌ی سرورها یکسان باشد مدیر ترجیح می‌دهد از سرور با مهارت بالاتر، بیشتر استفاده کند تا بهره‌وری کلی سیستم را بالا ببرد. اگر چه که این موضوع به عوامل دیگر نیز بستگی دارد.

مورد استفاده دیگری که می‌توان برای این سیاست در نظر گرفت، حالتی است که بیکار بودن بیش از حد سرور ها به هر نحوی باعث ایجاد حس بد میان آنها یا کاربران شود.

نکته‌ی دیگری که در مورد این سیاست می‌توان گفت این است که در این سیاست بین زمان کار کردن و استراحت سرورها و مهارت آنها تناسب نسبتاً خوبی وجود دارد.

در سیاست RR با توجه به اینکه هیچ وزن و معیاری وجود ندارد خیلی نمی‌شود بررسی کرد اما به طور کلی انتخاب کردن به صورت رندم مثل قرعه‌کشی می‌ماند و افراد با این ذهنیت هستند که این انتخاب شانسی صورت گرفته است و برای همین کمتر

معارض می‌شوند. بنابراین شاید در جایی که تعداد سرور ها زیاد است یا معیار خاص و دقیقی از نحوه عملکرد سرورها در سیستم در دست ما نیست (مثلا از زمان‌های بیکاری افراد اطلاعی نداریم)، این بهترین سیاست باشد چرا که باعث می‌شود سرورها معارض نشوند و فکر کنند که عدالت رعایت شده است.

سیاست SIR که بر اساس کمترین آخرین بیکاری است به طور کلی مناسب به نظر نمی‌رسد اما در بعضی موارد می‌تواند بهتر باشد. برای مثال حالتی را در نظر بگیرید که فعالیت سرورها به گونه‌ای است که اگر وقفه‌ای (زمان بیکاری) در کار آنها ایجاد شود، در عملکرد سرور تاثیر منفی داشته باشد.

یا برای مثال در حالتی که نرخ خدمت‌دهی سرورها متفاوت است این سیاست می‌تواند تا حدودی کمک کند سروری که نرخ خدمت‌دهی کمتری دارد که احتمالا به دلیل مهارت کمترش است، بیشتر کار کند و این به بهبود خدمت‌رسانی و بالا بردن مهارت سرور کمک کند.

ولی اگر قرار باشد عادلانه بررسی کنیم با در نظر گرفتن اینکه عدالت یعنی هر کسی در جای خود قرار بگیرد، معیارهایی لازم است. لازم است تا هدف سیستم و مدیر، نیازهای افراد، ماهیت کار افراد، دلیل تفاوت نرخ ها و ... مشخص باشد تا تصمیم درستی گرفته شود.

سوال ۵

در این سیاست همواره سعی می‌شود تا افراد به طور یکسان کار کنند و هیچ سروری بیشتر یا کمتر از دیگران استراحت نکند. به عبارتی این سیاست سعی دارد از همه‌ی افراد به یک اندازه استفاده کند. این عامل هم می‌تواند مزیت باشد و هم عیب.

در صورتی که پرداختی به سرورها یکسان باشد از دید یک مدیر می‌تواند مزیت باشد چراکه از همه‌ی نیروها در حال استفاده است. البته در همین حالت که سعی می‌شود تا هیچ ارجحیتی برای سرور با نرخ خدمت‌دهی بالاتر وجود نداشته باشد، در صورتی که هدف بیشتر شدن تعداد پیام‌های سرویس داده شده باشد، این سیاست ممکن است نسبت به سایرین خیلی خوب عمل نکند. زیرا در راستای این هدف بهتر است تا از سرورهای ماهرتر بیشتر استفاده شود.

از طرفی در حالتی که سرورها بر اساس میزان بهره‌وری دریافتی داشته باشند این سیاست می‌تواند در صورت قابل تغییر بودن نرخ خدمت‌دهی، باعث ایجاد رقابت و بهبود نتایج افراد و در نتیجه بهبود سیستم شود.

با نگاهی دیگر می‌توان دریافت که در این حالت مهارت افراد و .. که باعث تفاوت در نرخ خدمت‌دهی می‌شود در نظر گرفته نمی‌شود و همین می‌تواند باعث ایجاد بی‌عدالتی شود. یا حتی در مواردی باعث از دست رفتن انگیزه سرورهایی که نرخ خدمت‌دهی بالاتری دارند، شود که نتیجه‌ی این امر کم کاری سرورها و در نتیجه آسیب رسانی به سیستم است.

پس در کل زمانی می‌توان بررسی و تحلیل کرد که هدف سیستم، عوامل متغیر و سایر فاکتورها و .. مشخص باشد. در این حالت که اطلاعات محدود داریم نمی‌توانیم قضاوت درستی بکنیم.