



به نام خدا



تکلیف سوم درس سیستم‌های چندعاملی

۸۱۰۱۰۱۲۳۶

مرضیه علیدادی

۱ سوال اول

۱.۱ benevolent

اگر در یک سیستم عامل‌ها به طور ضمنی یک هدف مشترک داشته باشند، هیچ پتانسیلی برای ایجاد تضاد و اختلاف بین آن‌ها وجود نخواهد داشت. بدین ترتیب، عامل‌ها می‌توانند به گونه‌ای طراحی شوند که هر زمان که لازم باشد به یکدیگر کمک کنند؛ حتی اگر به این معنی باشد که یک یا چند عامل برای انجام آن کار باید متضرر شوند. به طور شهودی، تنها چیزی که اهمیت دارد، اهداف کلی سیستم است، نه اهداف هر یک از عامل‌ها. معمولاً این فرض زمانی امکان پذیر است که همه‌ی عامل‌های آن سیستم متعلق به یک سازمان یا یک فرد باشند. در نظر گرفتن این فرض برای یک سیستم، طراحی آن سیستم را ساده می‌کند.

۲.۱ self-interested

در این نوع سیستم‌های چندعاملی، نمی‌توان تصور کرد که عامل‌ها دارای یک هدف مشترک باشند، زیرا آن‌ها اغلب توسط افراد یا سازمان‌های مختلف طراحی شده‌اند تا به منافع آن‌ها دست یابند. بنابراین ممکن است منافع یک عامل با منافع دیگر عامل‌ها در تضاد باشد. با وجود پتانسیل تضاد منافع، عامل‌ها در این سیستم در نهایت برای دستیابی به اهداف خود نیاز به همکاری دارند.

۲ سوال دوم

با توجه به اینکه این تسک حساس است و تحمل خطای پایینی دارد، ترجیح داده می‌شود که سیستم از نوع task sharing طراحی شود.

۱.۲ result sharing

۱.۱.۲ مزایا

- ① هر عامل وظیفه‌ی تولید بخشی از خروجی کلی را بر عهده دارد و نتیجه‌ی نهایی از ترکیب همه‌ی خروجی‌های عامل‌ها به دست می‌آید. این رویکرد می‌تواند به دقت و کارایی بالاتر منجر شود زیرا هر عامل در یک کار خاص متخصص است.
- ② اشتراک‌گذاری نتایج می‌تواند مقیاس‌پذیرتر باشد زیرا می‌توان عامل‌های جدیدی را بدون نیاز به طراحی مجدد به سیستم اضافه کرد.

۲.۱.۲ معایب

- ① به اشتراک‌گذاری نتایج می‌تواند برای طراحی و پیاده‌سازی، پیچیده‌تر از به اشتراک‌گذاری وظایف باشد زیرا به هماهنگی بین عامل‌ها برای ترکیب نتایج آن‌ها نیاز دارد.
- ② به اشتراک‌گذاری نتایج می‌تواند از نظر محاسباتی گران‌تر باشد، زیرا هر عامل باید نتایج خود را با دیگران در میان بگذارد و آن‌ها را ترکیب کند.
- ③ به اشتراک‌گذاری نتایج می‌تواند تحمل خطای کمتری فراهم کند، زیرا اگر یک عامل از کار بیفتد، کل سیستم ممکن است از کار بیفتد.
- ④ احتمال به وجود آمدن گلوگاه در این سیستم‌ها وجود دارد.

۲.۲ task sharing

۱.۲.۲ مزایا

- ① اطمینان افزایش می‌یابد. راه‌حل‌های مشتق‌شده‌ی مستقل را می‌توان بررسی کرد، خطاهای احتمالی را برجسته کرد و اطمینان به راه‌حل کلی را افزایش داد.
- ② تحمل خطای بالاتری را فراهم می‌کند زیرا اگر یک عامل با شکست مواجه شود، عامل‌های دیگر می‌توانند به تولید نتایج ادامه دهند.
- ③ طراحی و پیاده‌سازی به اشتراک‌گذاری وظایف ساده‌تر است زیرا هر عامل وظیفه خاصی را بر عهده دارد.
- ④ به اشتراک‌گذاری وظایف می‌تواند از نظر محاسباتی هزینه‌ی کمتری داشته باشد زیرا هر عامل فقط باید وظیفه خاص خود را انجام دهد.

۲.۲.۲ معایب

- ① به اشتراک‌گذاری وظایف می‌تواند منجر به دقت و کارایی کمتری شود زیرا هر عامل ممکن است به تمام اطلاعات لازم برای تصمیم‌گیری آگاهانه دسترسی نداشته باشد.
- ② به اشتراک‌گذاری وظایف می‌تواند مقیاس‌پذیری کمتری فراهم کند زیرا افزودن عوامل جدید ممکن است به طراحی مجدد قابل توجهی نیاز داشته باشد.

۳ سوال سوم

۱.۳ الف

۱.۱.۳ مزایا

- ① این روش یک رویکرد رسمی و ساختار یافته برای طراحی سیستم به اشتراک گذاری وظایف ارائه می‌دهد که می‌تواند قابلیت اطمینان و کارایی سیستم را بهبود بخشد.
- ② استفاده از پیام‌ها در این روش، امکان ارتباط واضح بین عامل‌ها را فراهم می‌کند که می‌تواند به جلوگیری از سوء تفاهم و خطا کمک کند.
- ③ این روش امکان انعطاف پذیری در انتخاب عامل‌ها برای انجام وظایف را فراهم می‌کند، زیرا آن‌ها می‌توانند بر اساس در دسترس بودن و تخصص خود، پیشنهاد خود را برای انجام هر یک از وظایف ارائه کنند.
- ④ این روش منعطف است و می‌تواند با انواع مختلف وظایف و عامل‌ها سازگار شود.

۲.۱.۳ معایب

- ① پیاده سازی این روش می‌تواند پیچیده باشد، به خصوص برای سیستم‌هایی که تعداد زیادی عامل یا وظایف دارند.
- ② استفاده از پیام‌ها در این روش می‌تواند منجر به افزایش ترافیک شبکه و سربار ارتباط شود.
- ③ این روش مستلزم این است که عامل‌ها بتوانند با یکدیگر ارتباط و تعامل داشته باشند که ممکن است در همه‌ی سیستم‌ها قابل اجرا نباشد.
- ④ این روش نیاز به یک مشخصه‌ی وظیفه و معیارهای واجد شرایط بودن دارد که ممکن است در برخی موارد ایجاد آن دشوار باشد.
- ⑤ فرآیند مناقصه می‌تواند زمان‌بر باشد و ممکن است به میزان قابل توجهی ارتباط بین عامل‌ها نیاز داشته باشد.
- ⑥ این روش ممکن است در شرایطی که تعداد کمی از عامل‌های واجد شرایط وجود دارد یا در مواردی که عامل‌ها دارای منافع متضاد هستند، به خوبی کار نکند.

۲.۳ ب

۱.۲.۳ مفروضات

☐ عامل A به دنبال بهترین عامل برای انجام وظیفه T است.

☐ سه عامل B، C، D به درخواست پاسخ داده‌اند.

۲.۲.۳ مراحل

① Task announcement: عامل A یک پیام Task announcement را به همه‌ی عامل‌های واجد شرایط (B, C, D) ارسال می‌کند که حاوی شرح وظایف و معیارهای واجد شرایط بودن است.

② Bid processing: عامل‌های B, C, D، Task announcement را بررسی می‌کنند و تصمیم می‌گیرند که آیا برای کار پیشنهاد بدهند یا خیر. هر عامل یک پیام پیشنهادی به عامل A ارسال می‌کند که حاوی قیمت پیشنهادی خود و سایر اطلاعات مرتبط است.

③ Award processing: عامل A پیشنهادها را بررسی می‌کند و عاملی را که بهترین پیشنهاد را داشته باشد انتخاب می‌کند. عامل A یک پیام Award به عامل برنده (مثلاً عامل B) ارسال می‌کند که حاوی جزئیات کار و قیمت توافق شده است.

④ Request and inform processing: در طول اجرای وظیفه، عامل B ممکن است نیاز به درخواست اطلاعات از عامل A یا سایر عامل‌ها داشته باشد. اگر اطلاعات بلافاصله در دسترس باشد، Request با یک پیام inform انجام می‌شود. پس از تکمیل کار، عامل B یک پیام اطلاعاتی حاوی نتایج کار را برای عامل A ارسال می‌کند.

۴ سوال چهارم

۱.۴ الف

برای طراحی هر قسمت از سیستم، احتمال ایجاد گلوگاه در سیستم result sharing اجزا به طور جداگانه طراحی شده و سپس با هم مونتاژ می‌شوند، بیشتر است. این به این دلیل است که هر گونه تاخیر یا مشکل در طراحی یا آزمایش یک جزء می‌تواند کل فرآیند را به تاخیر بیندازد و منجر به گلوگاه شود. و همینطور اگر از ساختمان داده‌ی اشتراکی مانند blackboard استفاده شود، به دلیل مسائلی مانند انحصار متقابل و ... امکان ایجاد گلوگاه وجود خواهد داشت. در مقابل، در یک سیستم task sharing، وظایف بین عامل‌های مختلف توزیع می‌شود و هر عامل می‌تواند به طور مستقل روی وظیفه‌ی خود کار کند و احتمال ایجاد گلوگاه کاهش می‌یابد.

برای طراحی کل سیستم، تعیین اینکه کدام سیستم بیشتر باعث ایجاد گلوگاه می‌شود دشوار است. هر دو سیستم task sharing و result sharing مزایا و معایب خود را از نظر گلوگاه‌های بالقوه دارند. این بستگی به طراحی و اجرای خاص سیستم دارد.

۲.۴ ب

هم برای طراحی هر قسمت از سیستم و هم برای کل سیستم، یک سیستم task sharing تحمل خطای بیشتری دارد زیرا خرابی یک عامل بر کل سیستم تأثیر نمی‌گذارد. اگر یک عامل شکست بخورد، سایر عامل‌ها می‌توانند به کار بر روی وظایف خود ادامه دهند و عملکرد کلی سیستم ممکن است به طور قابل توجهی تحت تأثیر قرار نگیرد. در یک سیستم result

sharing، خرابی در یک جزء می‌تواند کل سیستم را تحت تاثیر قرار دهد و منجر به ریسک بالاتر خطا شود.

۳.۴ ج

برای طراحی هر قسمت از سیستم، یک سیستم result sharing عموماً هزینه بیشتری برای طراحی دارد. این به این دلیل است که اجزا باید به طور جداگانه طراحی شده و سپس با هم مونتاژ شوند که نیاز به هماهنگی و آزمایش بیشتری دارد. در مقابل، یک سیستم task sharing را می‌توان با عامل‌های مستقلی که بر روی وظایف خاص کار می‌کنند طراحی کرد که می‌تواند پیچیدگی و هزینه‌ی کلی طراحی را کاهش دهد. برای طراحی کل سیستم، تعیین اینکه کدام سیستم هزینه‌ی بیشتری برای طراحی دارد دشوار است. هر دو سیستم result sharing و task sharing مزایا و معایب خود را از نظر پیچیدگی و هزینه طراحی دارند. این بستگی به طراحی و اجرای خاص سیستم دارد.

۵ سوال پنجم

۱.۵ الف

اولین شما برای حل مشارکتی مسائل blackboard بوده است. نتایج با استفاده از این ساختمان داده به اشتراک گذاشته می‌شوند. چندین عامل می‌توانند آن را بخوانند و بر روی آن بنویسند. عامل‌ها راه‌حل‌های جزئی را بر روی آن می‌نویسند. این ساختمان داده ممکن است به صورت سلسله مراتبی ساختار یافته باشد. مسائلی مانند انحصار متقابل برای جلوگیری از فعالیت همزمان عامل‌ها، باید در نظر گرفته شود؛ که این، امکان به وجود آمدن گلوگاه را فراهم می‌کند.

سیستم blackboard یک معماری برای حل مسئله است که از یک ساختمان داده‌ی مشترک به نام blackboard و چندین entity مستقل به نام منابع دانش تشکیل شده است. هر منبع دانش دارای یک دانش تخصصی است که معمولاً در قالب قوانین کدگذاری شده است و قادر به خواندن و نوشتن روی blackboard است. فرآیند حل مسئله شامل منابع دانشی است که blackboard را زیر نظر می‌گیرند و زمانی که بتوانند راه‌حل‌های جزئی برای مسئله ارائه دهند، روی آن می‌نویسند. در واقع این فرآیند مانند این است که مجموعه‌ای از کارگران وجود دارند که به blackboard نگاه می‌کنند، هر کدام می‌توانند همه چیز را از روی blackboard بخوانند و تصمیم بگیرند که چه زمانی چیز ارزشمندی برای نوشتن و اضافه کردن به آن دارند.

به عنوان مثال، یک سیستم تشخیص پزشکی با استفاده از این معماری را در نظر بگیرید. منابع دانش می‌توانند در حوزه‌های مختلف پزشکی مانند قلب، مغز و اعصاب و ... متخصص باشند. هنگامی که یک بیمار با علائم وارد می‌شود، منابع دانش می‌توانند مشاهدات و فرضیات خود را روی blackboard بنویسند. سپس سیستم می‌تواند از این دانش ترکیبی برای ایجاد یک طرح تشخیص و درمان استفاده کند.

۲.۵ ب

blackboard حاوی اطلاعاتی در مورد بیماران و تسک‌های مورد نیاز آن‌ها و همچنین وضعیت هر تسک است. ساختار آن به شرح زیر توصیف می‌شود:

اطلاعات بیمار	تسک‌های مورد نیاز	وضعیت تسک‌ها	اطلاعات دیگر ...

منابع دانش را می‌توان متخصصانی در زمینه‌های مختلف مانند معاینه اولیه، آزمایش، بررسی آزمایش و ... در نظر گرفت. هر منبع دانش، blackboard را برای تسک‌هایی که می‌تواند در آن‌ها مشارکت داشته باشد، نظارت می‌کند و نتایج خود را در blackboard می‌نویسد. این سیستم می‌تواند تسک‌ها را بر اساس فوریت یا اهمیت اولویت بندی کند و در صورت امکان، پردازش موازی آن‌ها را فراهم کند. این معماری امکان حل کارآمد مسائل در یک محیط بیمارستانی پیچیده با وظایف و اولویت‌های متعدد را فراهم می‌کند. منابع دانش:

