

به نام خدا

# پاسخ تمرین چهارم سیستم‌های توزیع شده

استاد: دکتر کمندی

حسام مومیوند فرد

۸۱۰۸۰۳۰۶۳

پاییز ۱۴۰۳

## فهرست

۳.....	پاسخ سؤال اول
۳.....	عمل کرد الگوریتم
۴.....	پاسخ سؤال دوم
۴.....	عمل کرد الگوریتم
۴.....	اثبات
۴.....	Agreement
۴.....	Validity
۵.....	Termination
۶.....	پاسخ سؤال سوم
۶.....	اثبات درستی الگوریتم
۶.....	Agreement
۶.....	Validity
۶.....	Termination

## پاسخ سؤال اول

راه حل‌های زیادی برای این مسأله وجود دارد

### عمل کرد الگوریتم

برای این کار میتوان تعداد نودهای coordinator را افزایش داد و برای هر نود cohort یک بردار از تصمیماتی که coordinator ها اعلام میکنند نیز اضافه کرد. مراحل این الگوریتم به صورت زیر خواهند بود:

فاز اول:

در فاز اول هر نود cohort تصمیمش را به تمام coordinator ها اعلام میکند و coordinator ها نیز تصمیمات خودشان را به یکدیگر اعلام میکنند و هر coordinator با توجه به بردار نظرات و تصمیم خودشان، تصمیم نهایی را اعلام مشخص میکنند ( کاملاً مطابق با الگوریتم دوفازی ساده)

فاز دوم:

در فاز دوم هر نود cohort منتظر پاسخ coordinator ها میماند و زمانی که تنها یک مقدار در بردار تصمیماتش برابر ۱ شود تصمیم بر کامیت میگیرد.

برای اینکه پیچیدگی الگوریتم مناسب باشد می‌توان  $\log n$  نود را به عنوان coordinator در نظر گرفت و در این حالت پیچیدگی پیامی برابر با  $O(n \log n + (\log n)^2)$  خواهد بود و پیچیدگی زمانی کماکان  $2t$  باقی میماند.

## پاسخ سؤال دوم

### عمل کرد الگوریتم

در الگوریتم سه فازی تنها حالتی که باعث کند شدن می‌شود (در صورتی که فالتی نداشته باشیم) این است که نود coordinator دیر تصمیم بگیرد، و وجود سیگنال ready در حالتی که خود coordinator تصمیم گرفته باشد تنها یک فاز اضافه است، پس برای حالتی که coordinator از تمام cohort ها پیام تصمیم را گرفته بود و خودش نیز تصمیم گرفته بود میتوانیم فاز ready را حذف کرده و مستقیماً پیام decision را ارسال کنیم. اما برای حالتی که تصمیم گیری coordinator طولانی‌تر از همه شود چند حالت وجود دارد:

حالت اول:

در صورتی که coordinator دست کم یک تصمیم • دریافت کند در همان زمان بدون اینکه نیازی باشد به اتمام کار خودش میتواند تصمیم • را broadcast کند؛ زیرا باید شرط اول ولیدیتی رعایت شود.

حالت دوم:

در صورتی که تمام نود ها تصمیم یک را گرفته باشند؛ در این وضعیت نود coordinator باید کار خودش را تمام کند و بعد از اتمام کار خودش با توجه به تصمیم خودش به بقیه اعلام کند که کامیت انجام بشود و یا خیر. در این وضعیت هم میتوان الگوریتم را در دوفاز تمام کرد و فاز ready قابل حذف است.

با توجه به این الگوریتم پیچیدگی پیامی و زمانی کاملاً مشابه با الگوریتم دوفازی است و از مرتبه های خواسته شده در صورت سؤال خواهد بود.

## اثبات

### Agreement

به علت اینکه هیچ فالتی ای نداریم قطعاً تمام نود ها تصمیم میگیرند و در پایان فاز دوم هم پیامی از طرف coordinator که شامل تصمیم نهایی است دریافت خواهند کرد پس این شرط ارضاء میشود.

### Validity

در صورتی که حتی یک نود تصمیم • داشته باشد هم تصمیم نهایی ما با توجه به الگوریتم • خواهد بود.

در صورتی که تمام نود ها با ۱ شروع کنند به علت اینکه فالتی ای نداریم قطعاً تمام تصمیمات به دست coordinator خواهند رسید و خود او هم تصمیم بر ۱ دارد و حتماً تصمیم نهایی را ۱ اعلام میکند.

## Termination

در صورتی که نودها کار پردازش تسک خودشان را انجام دهند و به علت نداشتن فالتی؛ همه در نهایت تصمیم coordinator را دریافت کرده و تصمیم خواهند گرفت.

## پاسخ سؤال سوم

به تعداد  $f+1$  نود را به عنوان coordinator انتخاب میکنیم و هر کدام از آن‌ها یک بردار وزن مانند الگوریتم floodset دارند.

فاز اول:

در این فاز تمام نودهای cohort تصمیم خود را به مجموعه نودهای coordinator اعلام میکنند.

فاز دوم:

نودهای coordinator بین خود با بردار وزن‌هایی که ساخته شده الگوریتم floodset را  $f+1$  راند اجرا میکنند و بردار وزن‌ها را آپدیت میکنند.

فاز سوم:

بعد از اجرای فاز دوم دست کم یک نود coordinator باقی مانده است. در صورتی که بردار وزن‌هایی تماماً ۱ باشد تصمیم ۱ و در غیر این صورت تصمیم ۰ را برای تمام نودها ارسال میکند.

## اثبات درستی الگوریتم

### Agreement

در این وضعیت چون تصمیم‌هایی توسط coordinator اعلام می‌شود پس هیچ دو نودی تصمیم متفاوت نمیگیرند.

### Validity

در هیچ حالتی این شرط نقض نمیشود زیرا:

اگر تمام نودها با مقدار یکسان ۱ شروع کنند در بدترین شرایط یک نود cohort تصمیمش را به یک coordinator میگوید سپس از کار می‌افتد و در فاز دوم آن coordinator بدون اینکه وارد الگوریتم فلدست شود از کار خواهد افتاد در این وضعیت بعد از اجرای فلدست مقادیر تمام خانه‌ها به غیر از cohort از کار افتاده یک خواهد بود و ما میتوانیم مقدار آن را هم یک در نظر بگیریم زیرا اگر واقعاً ۱ بوده باشد که خبر درست حدس زده ایم و در غیر این صورت اصلاً شرط ورود به چک کردن ولیدیتی برقرار نبوده (تمام نودها با تصمیم یکسان شروع به کار نکرده اند). برای حالتی که تمام تصمیمات صفر بوده باشد و مقدار یک خانه را ندانیم هم که اصلاً مقدارش مهم نیست زیرا تصمیم‌هایی با یک خانه ۱ صفر هم صفر میشود.

## Termination

در نهایت حداقل یک نود coordinator باقی خواهد ماند که تصمیم نهایی را اعلام کند و الگوریتم پایان یابد.

### یک راه حل دیگر

در هر راند تمام نود ها تصمیم خود به همراه برداری که از تمام تصمیماتی که از بقیه دریافت کرده اند را به یکدیگر پاس میدهند و هر نودی که بردار تصمیماتش به ازای تمام نودها تکمیل شده بود پیام decision را در شبکه broadcast میکند. این الگوریتم الهام گرفته از phase two floodset است با این تفاوت که مثل floodset لزوماً به  $f+1$  راند نیاز ندارد و مانند phase two مشکل این را ندارد که در صورت از کار افتادن یک coordinator تصمیم گیری اتفاق نیوفتد.