

به نام خدا

تمرین اول درس ریزپردازنده

غزل زمانی نژاد

۹۷۵۲۲۱۶۶

قانون امدال بیان می کند که در یک سیستم دارای پردازش موازی مقدار تسریع دارای حد مشخصی است و از یک مقدار خاص بیشتر نمی شود. قانون امدال دارای یک فرمول است که به کمک آن می توان میزان speedup تاخیر در انجام یک حجم کار ثابت را به صورت عددی بیان کرد. Speedup یک نسبت است که با استفاده از آن عملکرد سیستم با بهبود آن سیستم نسبت به عملکرد سیستم بدون بهبود سنجیده می شود.

فرمول قانون امدال:

$$S_{\text{latency}}(s) = \frac{1}{(1 - p) + \frac{p}{s}}$$

p: برابر با نسبت زمان اجرایی تسک هایی که از بهبود سیستم استفاده می کنند به کل زمان اجراست. این نسبت همیشه از ۱ کوچکتر است. مثلاً در سیستمی زمان اجرای تسک هایی که با بهبود سیستم سود می برند ۱۰ ثانیه و زمان اجرای کل برنامه ۴۰ ثانیه است.

$$p = 10 / 40 = 0.25$$

s: برابر با میزان speedup سیستم بعد از بهبود منابع سیستم است. این نسبت همیشه از ۱ بزرگتر است. مثلاً در سیستمی زمان اجرای کل برنامه قبل از بهبود ۶ ثانیه و پس از بهبود ۳ ثانیه است. در این سیستم:

$$s = 6 / 3 = 2$$

• با مقداری محاسبات ریاضی می توانیم حد را در جایی که S به سمت بی نهایت میل می کند محاسبه کنیم.

$$\begin{cases} S_{\text{latency}}(s) \leq \frac{1}{1-p} \\ \lim_{s \rightarrow \infty} S_{\text{latency}}(s) = \frac{1}{1-p} \end{cases}$$

نتیجه می‌گیریم که هرچه سیستم را بهبود ببخشیم، باز هم latency کلی سیستم تحت تاثیر تسک‌هایی است که از بهبود سیستم سود نمی‌برند.

• همچنین اگر مقدار p کوچک باشد، optimization تاثیر کمی می‌گذارد.

به بررسی یک مثال می‌پردازیم: در یک سیستم تک‌هسته زمان اجرای برنامه‌ای با حجم ثابت ۲۰ ساعت است. اما این برنامه به گونه‌ای است که ۱ ساعت از تسک‌های آن نمی‌تواند به صورت موازی اجرا شود و باید به صورت سری اجرا شوند (یعنی ۱ ساعت از تسک‌ها از بهبود سیستم سودی نمی‌برند). پس در اینجا

$$p = 19 / 20 = 0.95$$

طبق حد محاسبه شده در بالا نتیجه می‌گیریم که میزان speedup این سیستم پس از بهبود نسبت به حالت قبل از بهبود نهایتاً ۲۰ است.

$$s = 1 / (1 - 0.95) = 20$$

قانون امدال تنها برای مسائلی مورد استفاده قرار می‌گیرد که در آن حجم تسک‌ها ثابت است. برای مسائلی که در آنها سایز ثابت نیست، از قانون Gustafson استفاده می‌شود.

این قانون برای پردازنده‌های چندهسته‌ای به صورت زیر درآمده است:

$$\text{Speedup}_{\text{parallel}}(f, n) = \frac{1}{(1-f) + \frac{f}{n}}$$

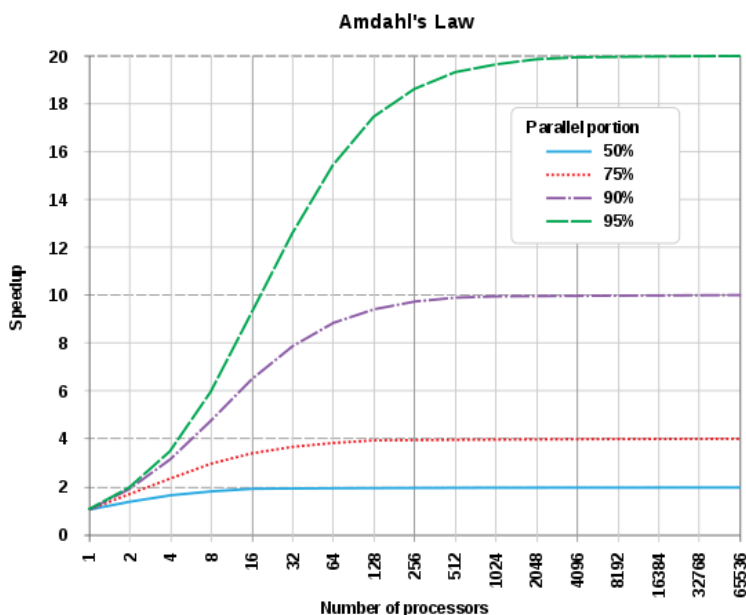
f : برابر با نسبت زمان اجرایی تسک‌هایی که از موازی‌سازی سیستم استفاده می‌کنند به کل زمان اجراست.

n : تعداد هسته‌های پردازنده

در پردازنده‌های چند هسته‌ای با اضافه کردن تعداد هسته‌ها به‌طور مداوم تنها در صورتی مفید است که برنامه‌ها از ۱۰۰٪ هسته‌ها استفاده کنند که این غیرممکن است. در یک نقطه مشخص (که با دانستن parallelization efficiency می‌توانیم آن نقطه را محاسبه کنیم) اگر از هسته‌های کمتری استفاده کنیم که فرکانس بیشتری دارند (نسبت به استفاده از هسته‌های زیادی که فرکانس کمتری دارند) به عملکرد بهتری دست می‌یابیم.

در شکل زیر به خوبی مشاهده می‌شود که اگر میزان تسک‌هایی که می‌توانند از موازی‌سازی استفاده کنند کم باشد، بالا بردن تعداد هسته‌ها تاثیر چندانی در بالا رفتن میزان speedup نخواهد داشت.

اما با رساندن میزان تسک‌های موازی به ۹۵٪، مشاهده می‌کنیم که زیاد کردن تعداد هسته‌ها تاثیر زیادی در بالا رفتن speedup دارد.



با وجود عملکرد مناسبی که قانون امدال دارد، این قانون محدودیت‌هایی نیز دارد. این محدودیت‌ها عبارتند از:

- تمامی اکشن‌هایی که در یک برنامه اجرا می‌شود، به میزان یکسانی از موازی‌سازی استفاده نمی‌کنند.
- قانون امدال تنها در صورتی اعمال می‌شود که CPU، bottleneck باشد. اگر برنامه ما توسط CPU محدود نشود، بعد از زیاد کردن تعداد هسته‌ها دیگر عملکرد بهبود نمی‌یابد.
- کد بسیاری از برنامه‌ها به‌گونه‌ای پیاده‌سازی می‌شود که از تعداد مشخصی هسته استفاده کند.
- تخمین زدن عملکرد CPUها تنها در صورتی دقیق است که معماری پایه CPUها مشابه باشند.

برای پاسخ به این تمرین از منابع زیر استفاده شده است:

<https://www.geeksforgeeks.org/computer-organization-amdahls-law-and-its-proof/>

https://research.cs.wisc.edu/multifacet/papers/ieeecomputer08_amdahl_multicore.pdf

https://en.wikipedia.org/wiki/Amdahl%27s_law

<https://www.pugetsystems.com/labs/articles/Estimating-CPU-Performance-using-Amdahls-Law-619/>