«به نام خدا»

عنوان تحقیق:

«بررسی قانون Amdhal
در پردازنده های تک هسته ای و چند هسته ای»

محقق: زهرا بخشنده

شماره دانشجویی: 98522157

استاد: دکتر میثم عبداللهی

نظریه محاسبه چند پردازشی نشان می دهد که یک محدودیت ریاضی در افزایش سرعت اجرای یک برنامه به صورت موازی وجود دارد.

وقتی برنامه ای روی بیش از یک CPU اجرا می شود، طبیعتا زمان اجرای آن باید کاهش یابد. اما چقدر؟ محدودیت سرعت چیست؟

افزودن CPU برای کوتاه کردن زمان اجرا

ما همواره می توانیم برنامه خود را با توزیع بخش های آن، بر روی چندین CPU، سریعتر اجرا کنیم. اما همیشه بخش هایی از منطق برنامه وجود دارد که باید به صورت سری، توسط یک CPU واحد اجرا شود.

بنابراین، دو محدودیت اساسی برای افزایش سرعت اجرای برنامه به صورت موازی وجود دارد:

- کسری از برنامه که می تواند به طور موازی اجرا شود، p، هرگز 100 نیست.
- به دلیل محدودیت های سخت افزاری، پس از یک مرحله خاص، اضافه کردن CPU جدید مزایای کمتری دارد.

تلاش برای اجرای موازی برنامه، به مقابله با این دو محدودیت برمی گردد. ما در تلاش برای افزایش کسر موازی شده از برنامه، یعنی p هستیم، زیرا در برخی موارد حتی یک تغییر کوچک در p (برای مثال، از 0.8 به 0.85) باعث تغییری چشمگیر در کارایی توسط CPU های اضافه شده می شود.

افزایش سرعت اجرا

افزایش سرعت حاصل از n-CPU، یعنی Speedup (n) ، در واقع نسبت زمان اجراي يك CPU به زمان اجراي موازي n-CPU است. یعنی: Speedup (n) = T (1)/T (n).

برای مثال، اگر زمان اجرای یک برنامه با یک CPU را 100 ثانیه اندازه بگیریم و برنامه در 60 ثانیه با 2 CPU اجرا شود، آنگاه: Speedup(2) = 100/60 = 1.67.

این عدد، افزایش کارایی سیستم، حاصل از افزودن سخت افزار به به آن را نشان می دهد. طبیعتا ما انتظار داریم T(n) کمتر از T(1) باشد. چون اگر اینطور نباشد، افزودن CPU باعث کندتر شدن برنامه شده است و حتما مشکلی وجود دارد!

بنابراین Speedup (n) باید یک عدد بزرگتر از 1 باشد، و هرچه بیشتر باشد ، ما بیشتر خوشحال می شویم😊

به طور شهودی، ممکن است امیدوار باشید که افزایش سرعت، برابر با تعداد CPU ها باشد، اما این ایده آل را هرگز نمی توان به دست آورد. به طور معمول، عدد Speedup (n) باید کمتر از n باشد، که نشان دهنده آن است که تمام قسمت های یک برنامه نمی توانند از اجرای موازی بهره مند شوند.

با این حال، ممکن است در موارد نادر Speedup (n) بزرگتر از n باشد، که به آن superlinear speedup می گویند.

تعریف قانون Amdhal

همانطور که گفته شد، همیشه بخش هایی از یک برنامه وجود دارد که نمی توانید آن ها را به صورت موازی یا همروند اجرا کنید. این بخش ها باید به صورت سری و به ترتیب اجرا شوند.

گزاره ریاضی این ایده را قانون آمدال می نامند. فرض کنیم p کسری از کد برنامه باشد که می توان به صورت موازی اجرا کرد (p همیشه کسری کمتر از 1.0 است). در نتیجه بقیه آن یعنی (1-p) کسر از کد باید به صورت سری اجرا شود. درعمل، p از 0.2 تا 0.99 متغیر است.

حال اگر این کسر p از محاسبه را با تسریع S افزایش دهیم، افزایش سرعت کلی برابر است با:

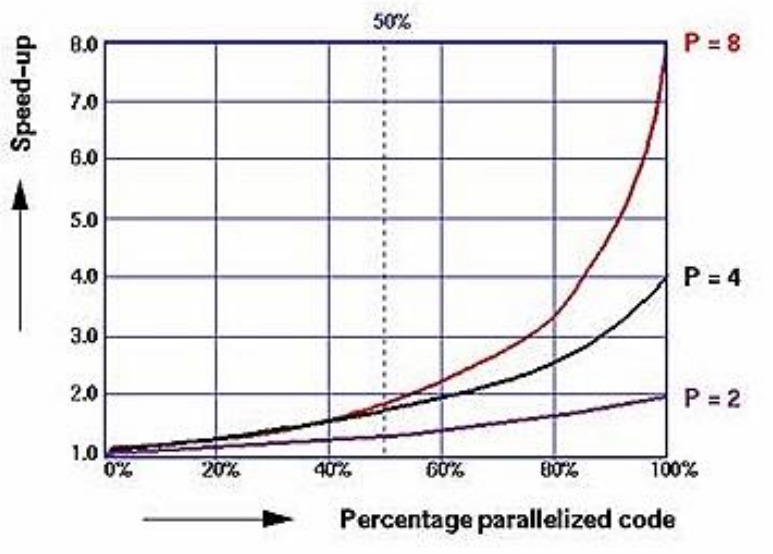
$$\text{Speedup (f, S)} = \frac{1}{(\frac{p}{S})+(1-p)}$$

طبق این قانون، در استفاده از n پردازنده به صورت موازی، مقدار حداکثر تسریع برای یک برنامه، متناسب است با: p تقسیم بر تعداد CPU هایی است که می توانید اعمال کنیم، به علاوه قسمت سریال باقی مانده، یعنی (1-p):

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

همانطور که n به سمت بینهایت میل می کند ، مقدار این عبارت به سمت $\frac{1}{(1-p)}$ میل می کند، و به معنی این است که قسمت ترتیبی کد برنامه ، در هر برنامه موازی می تواند محدودیت های شدیدی روی ماکزیمم سرعت اجرای قابل حصول قرار دهد . قانون AMDAHL به ما می گوید که به منظور استفاده ی مؤثر از از این ماشین های موازی ، قسمت ترتیبی کد باید در مقایسه با کل برنامه موازی ، بسیار کوچک باشد .

همانطور که در نمودار زیر نشان داده شده است ، کسر p تاثیر زیادی بر حداکثر افزایش سرعت دارد:



در واقع هرچه CPU های بیشتری داشته باشیم، از افزایش p سود بیشتری نسیبمان خواهد شد. با استفاده از تنها CPU 4، فقط به p = 0.6 نیاز داریم تا نصف تسریع ایده آل را بدست آوریم. اما با CPU 8 ، به p = 0.85 نیاز داریم تا به نصف افزایش سرعت ایده آل برسیم.

قانون Amdhal در پردازنده های چند هسته ای

برای اعمال قانون Amdahl بر روی تراشه چند هسته ای ، ما نیاز به یک مدل هزینه برای تعداد و عملکرد هسته هایی داریم که تراشه می تواند پشتیبانی کند. در اینجا ما یک مدل سخت افزاری ساده را بر اساس مدل نرم افزاری ساده Amdahl توسعه می دهیم.

ما ابتدا فرض می کنیم که یک تراشه چند هسته ای با اندازه مشخص می تواند حداکثر n هسته اصلی معادل داشته باشد. این محدودیت تعداد هسته ها، از منابعی ناشی می شود که طراح تراشه مایل است به هسته های پردازنده اختصاص دهد (با حافظه پنهان L1).

آنچه تراشه را به n هسته محدود می کند ممکن است قدرت باشد؛ ممکن است مساحت باشد؛ ممکن است ترکیبی از قدرت ، مساحت و عوامل دیگر باشد.

ما همچنین فرض می کنیم که طراحان، تکنیک هایی برای ایجاد هسته ای قدرتمند تر با عملکرد متوالی پیشرفته تر دارند. فرض می کنیم یک هسته قدیمی دارای عملکرد 1 باشد.

ما به طور خاص فرض می کنیم که معماران می توانند از منابع r هسته قدیمی برای ایجاد یک هسته جدید و قدرتمند با performance متوالی perf(r) استفاده کنند.

معماران همیشه باید منابع اصلی را هنگام perf (r) > r افزایش دهند ، زیرا انجام این کار هم اجرای متوالی و هم موازی را سرعت می بخشد. اما اگر perf (r) <r شود، افزایش عملکرد اصلی به اجرای متوالی کمک می کند ، اما به اجرای موازی آسیب می رساند.

تراشه های چند هسته ای Symmetric

یک تراشه چند هسته ای symmetric مستلزم آن است که تمام هسته های آن هزینه یکسانی داشته باشند. در این حالت، افزایش سرعت طبق قانون آمدال از رابطه زیر محاسبه می شود:

$$\text{Speedup}_{symmetric}\ (p, n, r) = \frac{1}{\frac{1-p}{perf(r)} + \frac{p.r}{perf(r).n}}$$

تراشه های چند هسته ای Asymmetric

جایگزین تراشه چند هسته ای symmetric، تراشه چند هسته ای asymmetric است که در آن یک یا چند هسته قوی تر از بقیه هستند. با فرض ساده انگاری قانون آمدال ، منطقی است که منابع اضافی را برای افزایش توانایی تنها یک هسته اختصاص دهیم. در این حالت، افزایش سرعت طبق قانون آمدال از رابطه زیر محاسبه می شود:

$$\text{Speedup}_{asymmetric}\ (p, n, r) = \frac{1}{\frac{1-p}{perf(r)} + \frac{p}{perf(r)+n-r}}$$

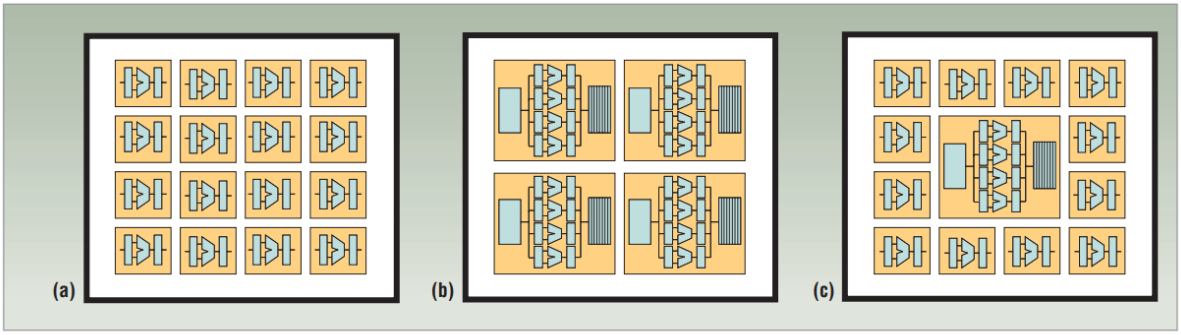


Figure 1. Varieties of multicore chips. (a) Symmetric multicore with 16 one-base core equivalent cores, (b) symmetric multicore with four four-BCE cores, and (c) asymmetric multicore with one four-BCE core and 12 one-BCE cores. These figures omit important structures such as memory interfaces, shared caches, and interconnects, and assume that area, not power, is a chip's limiting resource.

تراشه های چند هسته ای Dynamic

در این حالت، هسته های r را با برای افزایش عملکرد تنها جزء متوالی، به صورت dynamic با یکدیگر ترکیب می کنیم. در حالت متوالی، هنگامی که تکنیک های dynamic از منابع r هسته استفاده کنند، تراشه چند هسته ای dynamic می تواند با عملکرد perf (r) اجرا شود.

در حالت موازی ، تراشه چند هسته ای dynamic با استفاده از همه هسته های پایه به صورت موازی دارای performance ای برابر n می شود. به طور کلی، به رابطه زیر می رسیم:

$$\text{Speedup}_{dynamic}\ (p, n, r) = \frac{1}{\frac{1-p}{perf(r)} + \frac{p}{n}}$$

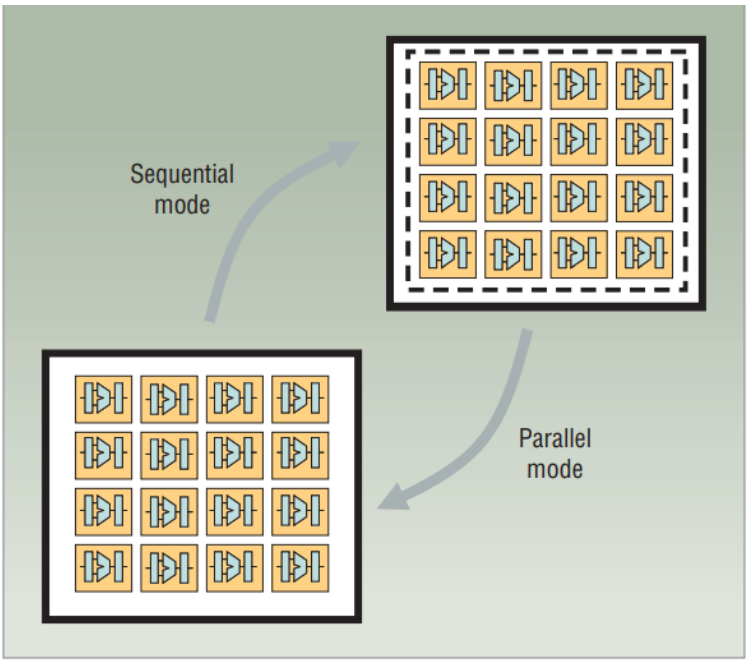


Figure 3. Dynamic multicore chip with 16 one-BCE cores.