

به نام خدا

برنامه نویسی چند هسته ای

تمرین اول

امیرمحمد پیرحسین لو

۹۵۳۱۰۱۴

(۱) huma(heterogeneous uniform memory access): نوعی از معماری است که برای ارتباط CPU و GPU هایی است که حافظه یکسان دارند. از مزایای آن می توان به برنامه نویسی ساده تر و کپی کم تر داده بین memory pool ها اشاره کرد.

Cache only memory architecture (COMA): نوعی از معماری در پردازنده های چند هسته ای است که از main memory به عنوان حافظه cache استفاده می شود. در NUMA اگر درخواست داده شود، داده از main memory کپی می شود و در cache قرار می گیرد اما در CUMA، حافظه اصلی ای وجود ندارد. درخواست داده از یک هسته به حافظه نهانی که که دسترسی مستقیم به آن ندارد موجب انتقال کامل آن داده به حافظه نهان هسته درخواست دهنده می شود. این امر موجب کاهش کپی های اضافه در حافظه می شود و کارایی را بالا می برد. از چالش های آن می توان به چگونگی پیدا کردن داده ها و سیاستی که هنگام پر بودن حافظه به کار برده می شود اشاره کرد.

(۲) اگر برنامه، a تا دستورالعمل داشته باشد و هر دستورالعمل b واحد زمانی طول بکشد، زمان کل اجرا برابر $a.b$ خواهد شد.

اگر همان برنامه، f درصدش هر دستورالعمل b/c طول بکشد (پردازنده c برابر سریع تر باشد)، زمان کل اجرا برابر $a.b(c + f - c.f)/c$ خواهد شد، یعنی $c/(c + f - c.f)$ برابر تند تر اجرا می شود.

Gustafson's law: میزان تسریع یک برنامه (S) که روی N هسته اجرا می شود در صورتی که s درصد برنامه شامل قسمت سریال است (قسمت سریال از موازی سازی سودی نمی برد) نسبت به زمانی که برنامه روی یک هسته اجرا شود برابر است با:

$$S = N + (1 - N) * s / 100$$

بیانی دیگر:

$$S_{latency}(s) = 1 - p + p.s$$

$S_{latency}$: میزان تسریع سرعت زمان رفت و برگشت کل برنامه

s : میزان تسریع سرعت زمان رفت و برگشت بخشی از کار که با بهبود منابع سیستم سرعتش بیشتر می شود.

p: میزان بخشی از کار که بهبود منابع سیستم سرعت آن را افزایش می دهد. (قبل اعمال بهبود)

شرایط: سایز مساله باید تغییر کند(افزایش یابد) اما با یک مقدار ثابت محدود شود. به عبارتی قانون بیان می کند که با بهبود منابع سیستم، برنامه بزرگتر را می شود در زمان یکسان اجرا کرد.

Sun-Ni's law: هر چه قدر که قدرت پردازشی سیستم بیشتر می شود و اندازه برنامه را بیشتر می کنیم تا در همان مدت زمان قبلی برنامه را اجرا کنیم، افزایش کارایی محدود به اندازه حافظه اصلی می شود.

شرایط: سایز مساله باید افزایش یابد اما باید با توجه به اندازه حافظه سیستم این افزایش صورت پذیرد(به اندازه حافظه سیستم محدود است).

(3) به دلیل زیاد بودن تعداد نخ ها، عملیات های context switch زیاد می شود که سرشار زیادی دارد و باعث کاهش کارایی می شود. از طرفی چون اندازه حافظه پنهان زیاد است و همچنین تعداد نخ ها زیاد است، تقریباً همه نخ ها در حافظه پنهان برای خود فضایی را در نظر میگیرند و حافظه پر می شود. وقتی حافظه پنهان پر شد و درخواستی miss شد، داده miss شده از حافظه اصلی به cache آورده می شود و جایگزین داده ی مربوط به یک نخ دیگر می شود. حال وقتی آن نخ کنترل پردازنده را در دست می گیرد، همه درخواست هایش miss می شود و باید داده خود را از main memory آورده و جایگزین داده مربوط به نخ دیگر در cache کند که همین روند تکراری در ادامه باعث می شود تعداد miss ها خیلی زیاد شود و کارایی خیلی پایین می آید.

(4) پردازنده مورد بررسی: Intel(R) Core™ i7-7500U

- a. تعداد هسته ها: 6 (4 تا مجازی)
- b. تعداد نخ ها: 4
- c. بیشینه فرکانس هسته ها: 2.90 GHz
- d. تعداد سطوح حافظه پنهان: 4
- e. اندازه ی حافظه های cache و سیاست های نوشتن در آن ها و اندازه ی خطوط حافظه پنهان:

i. L1 D-cache: 32 KB * 2 – 8-way set associative, 64-byte line size

ii. L1 I-cache: 32 KB * 2 – 8-way set associative, 64-byte line size

iii. L2 cache: 256 KB * 2 – 4-way set associative, 64-byte line size

iv. L3 cache: 4 MB – 16-way set associative, 64-byte line size

f. حجم حافظه اصلی سیستم و نوع آن: DDR4 – 16 GB

g. توپولوژی اتصال هسته های پردازنده: ring bus topology