

زمان‌بندی وظایف در سیستم‌های چندهسته‌ای با استفاده از الگوریتم ژنتیک

مقدمه

در این پروژه، به بررسی مسئله زمان‌بندی وظایف در سیستم‌های چندهسته‌ای می‌پردازیم. برای این منظور از یک الگوریتم ژنتیک برای تخصیص وظایف به هسته‌ها به گونه‌ای استفاده می‌شود که سربار محاسباتی و عدم توازن کاهش یافته و کیفیت خدمات (QoS) حفظ شود.

۱. تولید وظایف با استفاده از الگوریتم UUNIFAST

برای تولید وظایف با میزان مصرف خاص از الگوریتم UUNIFAST استفاده می‌کنیم که در آن مجموع مصرف پردازنده‌ها برابر با یک مقدار مشخص باقی می‌ماند.

- ابتدا میزان استفاده کلی از پردازنده را بین وظایف مختلف به طور تصادفی تقسیم می‌کنیم.
- سپس برای هر وظیفه، یک دوره (پریود) رایج مانند ۱۰، ۲۰، ۴۰، ۵۰ یا ۱۰۰ واحد زمانی انتخاب شده و زمان اجرا به صورت $\text{period} \times \text{utilization} = \text{execution}$ محاسبه می‌شود.
- ضرب‌الاجل هر وظیفه نیز برابر با پریود آن در نظر گرفته شده است (ضرب‌الاجل ضمنی).

۲. پیاده‌سازی الگوریتم ژنتیک

الگوریتم ژنتیک برای یافتن تخصیص بهینه وظایف به هسته‌ها طراحی شده است. مراحل کلیدی آن به صورت زیر است:

مرحله تولید جمعیت اولیه

تخصیص‌های اولیه به صورت تصادفی تولید می‌شوند؛ هر کروموزوم آرایه‌ای است که در آن مقدار هر خانه نشان‌دهنده‌ی شماره‌ی هسته‌ای است که وظیفه به آن تخصیص یافته است.

تابع برازندگی (Function Fitness)

برای ارزیابی هر کروموزوم:

- مصرف هر هسته جمع می شود.
- اگر مصرف هسته ای از ۱ بیشتر شود، جریمه اعمال می شود.
- همچنین برای توازن مصرف بین هسته ها، انحراف معیار در برازندگی لحاظ شده است.

انتخاب والدین

والدین براساس احتمال تناسب شان (برازندگی) انتخاب می شوند (روش چرخ رولت).

تقاطع (Crossover) و جهش (Mutation)

فرزندان از والدین تولید می شوند. تقاطع نقطه ای و جهش تصادفی روی ژن ها انجام می شود تا تنوع جمعیت حفظ شود.

الیتسم (Elitism)

بهترین کروموزوم ها بدون تغییر به نسل بعد منتقل می شوند.

۳. تحلیل و محاسبه معیارها

پس از تخصیص وظایف به هسته ها، معیارهای زیر محاسبه می شوند:

- میزان مصرف هر هسته: جمع مصرف وظایف تخصیص یافته به هر هسته.
- **Makespan**: زمان کلی اجرای سیستم، به صورت $\max(\text{utilization}) \times \text{hyperperiod}$.
- **QoS** وظایف: ۱۰۰ درصد در صورت مصرف $1 \leq$.
- **QoS** سیستم: در صورتی که مصرف هیچ هسته ای بیشتر از ۱ نباشد برابر با ۱۰۰ است.

۴. شبیه سازی و پیکربندی ها

شبیه سازی برای پیکربندی های مختلف اجرا می شود. در این پیکربندی ها تعداد هسته ها ۸، ۱۶ و ۳۲ بوده و بارگذاری به صورت ۲۵۰۰ تا ۰۰۱ در نظر گرفته شده است.

- برای هر پیکربندی، حدود ۳ برابر تعداد هسته ها وظیفه تولید می شود.
- سپس الگوریتم ژنتیک اجرا شده و تخصیص وظایف انجام می گردد.
- نتایج به صورت عددی و تصویری تحلیل می شوند.

۵. تجسم نتایج

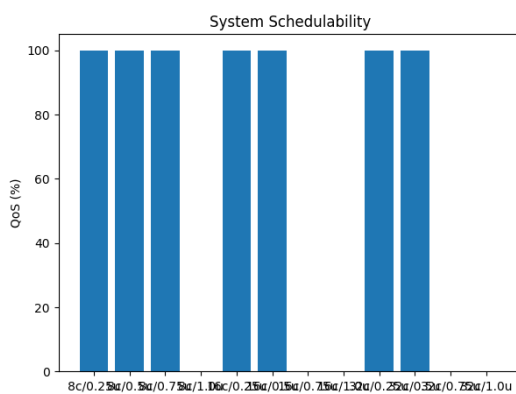
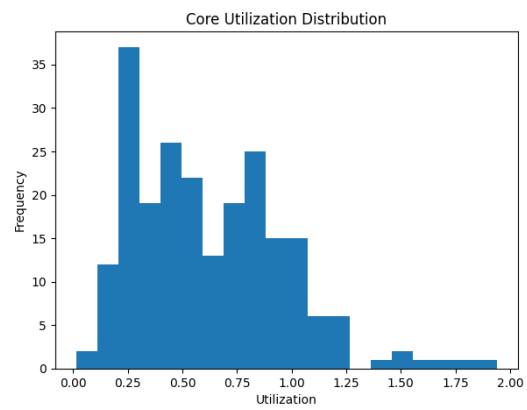
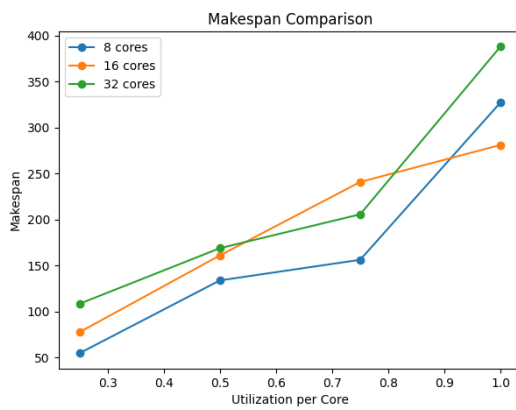
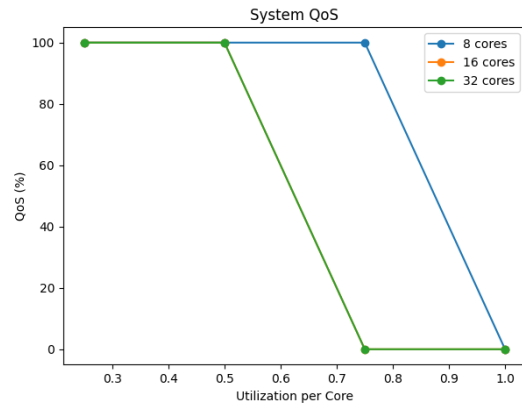
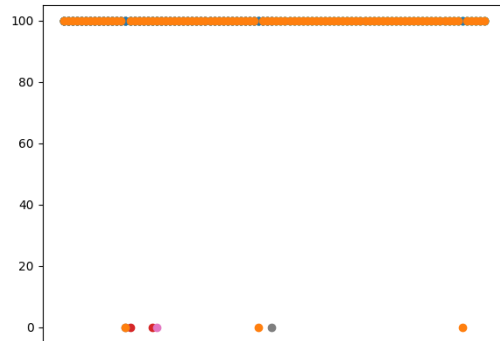
نمودارهای زیر رسم می شوند:

- نمودار QoS وظایف: میزان QoS برای هر وظیفه به صورت نقطه ای رسم می شود.
- نمودار QoS سیستم: به ازای بارهای مختلف در سیستم های با تعداد هسته متفاوت.
- نمودار Makespan: مقایسه زمان اجرای سیستم در شرایط مختلف.
- توزیع مصرف هسته ها: نمایش فراوانی مصرف در هسته های مختلف.
- نمودار قابلیت زمان بندی (Schedulability): درصد QoS سیستم در پیکربندی های مختلف.
- جدول نمونه وظایف: نمایش پارامترهای ۵ وظیفه ای ابتدایی در اولین پیکربندی.

نتیجه گیری

این مطالعه نشان می دهد که استفاده از الگوریتم های تکاملی مانند الگوریتم ژنتیک می تواند در زمان بندی وظایف در سیستم های چند هسته ای موثر باشد. معیارهایی مانند توازن مصرف، QoS و Makespan در این رویکرد بهینه می شوند.

نکته: برای اجرای این پروژه به کتابخانه های NumPy، Matplotlib در پایتون نیاز دارید. نتایج به صورت تصویری ذخیره شده و در فایل `one_results.png.phase`



Sample Task Parameters

ID	Rate	Period	Deadline	Util
0	1.0387797254017028	30	30	0.10387797254017028
1	0.2745615471608976	50	50	0.005491236874321795
2	2.3557101521578927	20	20	0.1177850760789404
3	4.327778401469218	300	300	0.04227778401469218
4	3.5212585814833997	300	300	0.039212585814833997

چند هسته‌ای سیستم‌های در وظایف زمان‌بندی به مربوط بصری نتایج نمایش ۱: شکل