

پروژه نهایی درس آزمایشگاه سیستم عامل

ترم اول سال تحصیلی ۱۴۰۳-۱۴۰۲

تعریف مساله:

Readahead یک فراخوانی سیستمی است که محتویات یک فایل را در کش صفحه یا Page Cache بارگذاری می کند. این فراخوان سیستمی، فایل را از قبل واکنشی می کند تا زمانی که بعداً به آن دسترسی پیدا کرد، محتوای آن به جای آن که از دیسک خوانده شود از حافظه اصلی خوانده شود و در نتیجه تأخیر دسترسی به فایل بسیار کمتر می شود. تعیین میزان خواندن پیش رو یا Readahead همیشه چالش برانگیز است:

- قابلیت Readhead خیلی کم، نیاز به خواندن های زیادی از دیسک خواهد داشت.
- تعداد زیاد Readahead باعث آلودگی Cache با داده های بی فایده می شود.
- هردو مورد بالا به عملکرد سیستم عامل آسیب می رساند.

مقدار Readahead نمونه بارز یک پارامتر مهم در سیستم ذخیره سازی است. در حالی که این پارامتر قابل تنظیم است، اغلب ثابت در نظر گرفته می شود و مقدار پیش فرض آن مورد استفاده است. برخی از سیستم عامل ها به کاربران اجازه می دهند نکات کمکی را از طریق دستورات *Fadvise* و *Madvise* به سیستم عامل اطلاع دهند تا به سیستم عامل کمک کنند تا فایل هایی را که صرفاً متوالی یا تصادفی استفاده خواهند شد را تشخیص دهند. اما این نکات کمکی اغلب در یافتن مقادیر بهینه برای بارهای کاری یا Workload های در حال تغییر، ناکام هستند. در این پروژه هدف ما پیش بینی اندازه های بهینه Readahead در حالی که تحت بار کاری پویا I/O کار می کند، است.

با اجرای ۴ بنچ مارک مختلف RockDB [۱]، هر کدام با ۲۰ اندازه Readahead در مطالعات پیشین سعی شده است تا اندازه های Readahead را طوری تعیین کنند که بهترین عملکرد را (در واحد عملیات در ثانیه) برای هر بار کاری داشته باشند. این بنچ مارک ها به عنوان داده های آموزشی استفاده شده است که می تواند برای پیش بینی مقادیر Readahead در بارهای کاری و محیط های مختلف کمک کننده باشد. بررسی ها نشان می دهد که هر بار کاری رفتار منحصر به فردی دارد و برای رسیدن به عملکرد بهینه نیاز به اندازه Readahead متفاوتی دارد. ارتباط بین الگوهای دسترسی به فایل، برجسب های بار کاری RocksDB و عملکرد سیستم عامل بررسی شده است تا اطلاعات و ویژگی های مورد نیاز برای ساخت یک مدل خوب تعیین شود.

جمع آوری داده:

از LTTng [۲] برای جمع آوری داده های ردیابی استفاده کنید، سپس از آن ها برای یافتن ویژگی های مفید (Feature) برای مساله Readahead استفاده کنید. بیشتر نقاط ردیابی کش صفحه [۳] را ثبت کنید (به عنوان مثال، `add_to_page_cache` یا `writeback_dirty_page`). به عنوان مثال می توانید از ردیابی با اجرای چندین بنچمارک ۱۰ دقیقه ای RocksDB در دستگاه NVMe-SSD داده جمع آوری و پردازش کنید. ده دقیقه برای RocksDB کافی است تا به حالت ثابت برسد. پس از بررسی این ردیابی ها، مجموعه ای از ویژگی های کاندید را بر اساس دامنه ی مساله انتخاب نمایید. سپس ویژگی های مورد نظر را انتخاب کرده و توابع hook ای طراحی کنید که مسئول جمع آوری اطلاعات لازم (مثلاً `struct page`) برای استنتاج



پروژه نهایی درس آزمایشگاه سیستم عامل

ترم اول سال تحصیلی ۱۴۰۳-۱۴۰۲

باشند و آن ها را فراخوانی کنید. توابع hook مثلا می توانند داده های خام مهمی مانند: اختلاف زمانی از ابتدای اجرا، شماره inode و آفست صفحه فایل هایی که در مکان هایی که هوک ها به آنها دسترسی پیدا کرده است باشند.

پنج ویژگی زیر را می توانید برای مدل یادگیری خود انتخاب کنید: تعداد تراکنش های انجام شده در هر ثانیه، میانگین متحرک تجمعی محاسبه شده و انحراف استاندارد متحرک تجمعی آفست های صفحه، میانگین تفاوت های آفست مطلق صفحه برای تراکنش های متوالی و شماره inode (برای اطمینان از اینکه ما فقط دسترسی های فایل RocksDB را پردازش می کنیم).

سوالات پروژه:

- ۱- مجموعه داده ی مناسبی براساس توضیحات فوق، مطالعه ی منابع ارجاع داده شده و جستجو در اینترنت ایجاد کنید.
- ۲- ویژگی های مناسب را براساس توضیحات فوق و بررسی های خود استخراج نمایید و سایر ویژگی ها را حذف کنید.
- ۳- مجموعه داده را براساس رویکرد Z-Score نرمال سازی کنید.
- ۴- با استفاده از رویکرد t-NSE اقدام به نمایش داده ها نمایید.
- ۵- یک مدل شبکه عصبی برای دسته بندی ۴ کلاس شامل کلاس های readseq، readrandom، readreverse و readrandomwriterandom با استفاده از تابع خطای Cross-Entropy و تنظیم دقیق هایپرپارامترها طراحی کنید. مدل را به روش ۱۰-fold cross validation آموزش دهید و نتایج آن را با معیار Accuracy گزارش کنید.
- ۶- یک مدل درخت تصمیم با توضیحات سوال قبل آموزش و سپس نتایج آن را براساس Accuracy همراه با شمای درخت تصمیم نمایش دهید.
- ۷- نتایج بدست آمده در سوال ۵ و ۶ را مقایسه کنید و تشریح کنید که دست آورد شما از انجام این پروژه چه بوده است. در حقیقت نشان دهید که چقدر مساله را درک کردید و در نهایت چه نتیجه ای گرفته اید.
- ۸- پس از اتمام فرآیند فوق، مدل بهتر را طی پیاده سازی یک فراخوان سیستمی با عنوان Readahead که قابلیت های فوق را داشته باشد به Pintos اضافه کنید و این فراخوان سیستمی را در گزارش خود تشریح کنید. لازم به ذکر است که افزودن این فراخوان سیستمی به Pintos ممکن است نیازمند تغییرات دیگری در سایر فایل های هسته ی Pintos نیز باشد. بنابراین در صورت تغییر سایر فایل ها در گزارش خود به آن ها اشاره کنید.



پروژه نهایی درس آزمایشگاه سیستم عامل

ترم اول سال تحصیلی ۱۴۰۲-۱۴۰۳

منابع:

- [1] Facebook. RocksDB. <https://rocksdb.org/>, September 2019.
- [2] LTTng. LTTng: an open source tracing framework for Linux. <https://lttng.org>, April 2019.
- [3] Mathieu Desnoyers. Using the Linux kernel tracepoints, 2016. <https://www.kernel.org/doc/Documentation/trace/tracepoints.txt>.

*** هر گروه یک فایل PDF حاوی گزارش کتبی پروژه به همراه مجموعه ی داده ها و کدهای توسعه داده شده را حداکثر تا پایان وقت روز ۱۰ بهمن ماه به آدرس ایمیل s_nabavi@sbu.ac.ir ارسال نماید. سعی شود به تمامی سوالات پروژه به تفصیل و با توضیحات مناسب پاسخ داده شود. اشکال مناسب در متن گزارش موجب درک بهتر خواهند بود.

موفق باشید

نبوی