باسمه تعالی

مستند تحقیق کاربردی

پروژه پیاده سازی پایلوت سرویس ابری هوش مصنوعی

کارفرما: مرکز نوآوری همراه اول

مدیر پروژه: دکتر محمدحسین رهبان

مقدمه

این مستند در راستای معرفی راهکارها و تکنولوژی های مختلفی که در طول پروژه پیاده سازی پایلوت سرویس ابری هوش مصنوعی بررسی و بعضا استفاده شده اند آماده شده است. در این مستند تلاش شده است تا خواننده دید کاملی نسبت به مزایا و معایب هرکدام از این تکنولوژی ها و دلایلی که منجر به تصمیم گیری در مورد استفاده یا عدم استفاده از آنها در محصول نهایی شده است، پیدا کند.

ابزارهای پردازش توزیع شده و مدیریت منابع

یکی از اولین و مهم ترین تصمیماتی که تیم توسعه درگیر آن بود انتخاب بستر مدیریت منابع برای اجرای توزیع شده کدهای یادگیری ماشین توسعه داده شده توسط فریمورک های apache spark و PyTorch بود. با بررسی تکنولوژی های موجود در این حوزه نتیجتا به دو کاندیدای اصلی رسیدیم که در ادامه هرکدام را بررسی خواهیم کرد.

Apache Hadoop Yarn

در منابع موجود اولین تکنولوژی ای که به منظور مدیریت منابع در یک سیستم توزیع شده که توسط apache spark توسعه داده شده است، معرفی می شود، Apache Hadoop yarn می باشد که عضوی دیگر از خانواده Apache Foundation می باشد.

Apache Yarn: این تکنولوژی در ابتدا به منظور فراهم کردن بستری برای اجرای ایزوله پردازه های نوشته شده به زبان Java برای پردازش داده های حجیم در یک محیط sandbox توسعه داده شد. سپس در نسخه های بعدی این تکنولوژی قابلیت پشتیبانی از Docker Container ها نیز به این تکنولوژی اضافه شد. یکی از مهم ترین جنبه هایی

که در توسعه این زیرساخت مدنظر قرار گرفته است موضوعات مرتبط با امنیت می باشد. Yarn از روش های مختلفی نظیر Kerberos، دسته بندی Docker image ها به privileged/non-privileged برای دسترسی به منابع به منظور تامین امنیت پردازش ها استفاده می کند. همچنین اکثر تنظیمات مرتبط با امنیت Docker image ها به صورت پیش فرض در حالت بسته قرار دارد و مدیر و توسعه دهنده به صورت دستی باید اجازه دسترسی های بیشتر به Docker image ها را بدهد که این موضوع باعث می شود توسعه دهندگانی که تجربه کمتری در کار با این ابزار دارند احتمال خطای کمتری در ابتدا داشته باشند. به طور کلی می‌توان گفت Apache Yarn همخوانی مناسبی با سایر محصولات خانواده Apache نظیر Spark, Hive, Impala, Hadoop HDFS دارد و هماهنگ سازی این محصولات با یکدیگر به سادگی قابل انجام است.

یکی از مهم ترین پارامتر های تاثیر گذار در انتخاب Apache Yarn به عنوان بستر اجرای توزیع شده پردازش های یادگیری ماشین، امکان و سهولت اجرای پردازش های یادگیری عمیق توسعه داده شده توسط PyTorch بود. اجرای این پردازش ها به صورت پیش فرض در بستر Apache Yarn امکان پذیر نمی باشد. اما ابزارهای واسطی بدین منظور توسعه داده شده اند که از آنها می‌توان به TonY و Apache Submarine اشاره کرد.

TonY یا Tensorflow on Yarn در ابتدا به منظور یک فریمورک برای اجرای پردازش های Tensorflow بر روی Apache Yarn توسعه داده شد. اما در نسخه های بعدی آن از MxNet و PyTorch نیز پشتیبانی کرد.

Apache Submarine نیز یکی از زیرپروژه های Apache Hadoop می باشد که به منظور اجرای پردازش های PyTorch و Tensorflow بر روی Apache Yarn توسعه داده شده است. اما هنوز در نسخه بتا قرار دارد و یک نسخه stable از آن توسط Apache Foundation ارائه نشده است.

در مجموع می‌توان گفت هردوی تکنولوژی های بالا به دلیل اینکه در نسخه های اولیه قرار دارند و جامعه استفاده کننده زیادی ندارند نتیجتا منابع کافی برای رفع مشکلاتی که در طول کار با آنها رخ می دهد در سطح اینترنت وجود ندارد.

Kubernetes: کوبرنیتیز یک سیستم open-source که در ابتدا توسط google به منظور پیاده سازی یک پلتفرم برای اتوماتیک کردن پروسه های deployment, scaling و اجرای container ها در سطح یک کلاستر از سیستم های فیزیکی که نقش host را دارند توسعه داده شده است. Kubernetes برای مدیریت منابع و پردازش ها یک رویکرد پایین به بالا را اتخاذ کرده است و اجازه می دهد به صورت همزمان چندین job schedulers در داخل کلاستر pod های خود را مدیریت کنند. این رویکرد باعث شده است که توسعه دهندگان امکان بهینه سازی مناسبی در اختصاص دادن منابع به هرکدام از pod ها داشته باشند اما معایبی نیز دارد. از جمله معایب آن می‌توان به این مورد اشاره کرد که این رویکرد باعث می شود در زمانی که نیاز pod ها بیشتر از منابع فیزیکی در اختیار کلاستر است، کلاستر دچار نوعی ناپایداری بشود. به طور کلی می‌توان گفت Kubernetes در زمانی که منابع فیزیکی کافی در اختیار کلاستر است

به خوبی عمل می کند. از مهم ترین مزایای استفاده از Kubernetes می توان به این مورد اشاره کرد که به دلیل پشتیبانی بسیار خوب google از این محصول در حال حاضر یک اکوسیستم غنی از ابزارهای مختلف که به راحتی با Kubernetes هماهنگ می شوند وجود دارد و همچنین مستندات بسیار خوبی حول این تکنولوژی و ابزارهای جانبی آن در سطح اینترنت وجود دارد. به طور خلاصه تر می‌توان گفت community استفاده کنندگان از این تکنولوژی به خوبی پشتیبانی های لازم برای یادگیری و رفع مشکلات پیش آمده در طول استفاده از این محصول را ارائه می دهند.

نهایتا تیم توسعه پایلوت سرویس ابری هوش مصنوعی به دلیل جامعه کاربری بزرگ تر و همچنین اکوسیستم غنی تر Kubernetes از یک سو و از سوی دیگر عدم هماهنگی بالای apache yarn با فریمورک PyTorch تصمیم به استفاده از Kubernetes به عنوان زیرساخت مدیریت منابع سخت افزاری و اجرای توزیع شده پردازش ها گرفت.

مدیریت و اجرای پردازش های یادگیری ماشین توسعه داده شده در بستر Apache Spark

به طور کلی برای اجرای پردازه های spark بر روی زیرساخت Kubernetes دو روش وجود دارد. روش اول استفاده از Spark-submit می باشد. در واقع این روش یک روش پایه ای برای اجرای spark بر روی Kubernetes است که توسط خود Apache Spark ارائه شده است. در ابتدا تیم توسعه نیز تلاش کرد به کمک این روش پردازه های spark را بر روی کلاستر اجرا کند. از مهم ترین نقطه ضعف های این روش می‌توان به این مورد اشاره کرد که این روش باعث می شود که توسعه دهنده کنترل یکپارچه ای روی منابع تولید شده در Kubernetes نداشته باشد و همچنین این روش از اجرای پردازه ها از kubeflow که یک مکانیزم اجرای پردازه های یادگیری ماشین بر روی Kubernetes است نیز پشتیبانی نمی کند.

روش دوم استفاده از Spark-Operator است. Spark-Operator در واقع یک ابزار جانبی است که توسط Google Cloud Platform ارائه شده است. Spark-Operator با استفاده از قابلیت Kubernetes Custom resources یک منبع سفارشی شده در Kubernetes برای پردازش های اسپارک می سازد. در این صورت ارسال یک پردازش اسپارک بر روی کلاستر مشابه ارسال یک پردازش عادی Kubernetes می شود که منجر به یکپارچگی سیستم و سهولت مدیریت منابع توسط توسعه دهنده خواهد شد. همچنین این روش تنها روشی است که علاوه بر اجرای مستقیم پردازش بر روی Kubernetes می‌توان از آن به عنوان روشی برای اجرای پردازش های اسپارک بر روی Kubeflow استفاده کرد.

در نهایت تیم توسعه پس از استفاده اولیه از spark-submit و مشاهده مشکلات پیش آمده تصمیم گرفت که از روش spark-operator به عنوان روش اصلی اجرای پردازش های spark بر روی Kubernetes استفاده کند