

Figure 2: Some screenshots from MAPLE (a) Adding a new model to the data store (b) Cloud deployment cost display (c) Cost and execution time for processing documents in Deep Reader with different serverless configurations

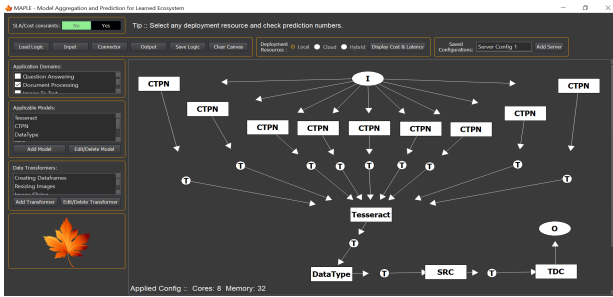


Figure 3: MAPLE Tool Screen

Performance and cost model: Maple has in-built performance and cost extrapolation models. These models are used to estimate the performance and cost of inference when performance data is not available in the data store.

Deployment orchestrator: The deployment orchestrator uses model data store as well as performance and cost models to evaluate the cost of deployment on heterogeneous architectures (CPU, GPU) and services (VMs instances, serverless etc.) over multiple clouds such as AWS, Azure, and Google Cloud Platform. The cost is evaluated by fetching the cost specifications available on the cloud vendor’s website. In case user specifications are provided, appropriate hardware configuration and architecture are suggested for the deployment. The MAPLE GUI has the following features:

SLA specification: Using a toggle button, a user can specify whether there are specific SLAs to meet or not.

Deployment option selection: MAPLE supports local (data center), cloud (includes AWS, GCP, Azure), and Hybrid (includes a mix of supporting clouds and local servers) deployment options. Users can select any one option and instance to get the end-to-end cost and latency of designed workflows on the provided configurations.

Server Configuration: This palette gets enabled once user selects "local" or "hybrid" as a deployment resource. Users can add any number of server configurations and use them to design a logic.

Application Domain selection: MAPLE has a few predefined domains listed in this palette. User is allowed to choose or add one or more domains that suit the application.

Applicable Model Selection: As per the selection of appropriate applicable domains in the palette mentioned above, the ML/DL models which fall under respective domains are populated in this palette.

Users can select any number of models and pull on the canvas for design application logic. Users can add to this list (figure 2(a)) *Transformer Function Selection:* The end-to-end latency of an application consisting of ML/DL models is affected by the pre-processing of input data and post-processing of output data. MAPLE provides the creation and application of such data transformers.

3 USECASE

We demonstrate the use of MAPLE for estimating the performance of an information extraction application called Deep Reader [2, 3] on the AWS serverless platform Lambda. The application consists of multiple models for text detection (e.g. CTPN [5] or CRAFT [1]), text recognition (e.g. Tesseract cietes), and text labeling along with transformation functions for data processing. Users can choose the models of their choice using the information extraction application in the tool. A complete workflow is designed using the required models as shown in figure 3. Using the characterization data available in the data store for the models, the end-to-end cost and performance are evaluated for various configurations of the serverless instances chosen as shown in figure 2(b) and 2(c).

4 CONCLUSION

We proposed a tool for visualizing AI workflows and estimating the performance and cost of deploying these workflows on multiple clouds. Additionally, tools features are demonstrated using an information extraction application called Deep Reader.

REFERENCES

- [1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoon Yun, and Hwalsuk Lee. 2019. Character region awareness for text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9365–9374.
- [2] Dheeraj Chahal, Ravi Ojha, Manju Ramesh, and Rekha Singhal. 2020. Migrating Large Deep Learning Models to Serverless Architecture. In *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. 111–116. <https://doi.org/10.1109/ISSREW51248.2020.00047>
- [3] Dheeraj Chahal, Manju Ramesh, Ravi Ojha, and Rekha Singhal. 2021. High Performance Serverless Architecture for Deep Learning Workflows. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 790–796.
- [4] Daniel Crankshaw, Gur-Eyal Sela, Xiangxi Mo, Corey Zumar, Ion Stoica, Joseph Gonzalez, and Alexey Tumanov. 2020. InferLine: Latency-Aware Provisioning and Scaling for Prediction Serving Pipelines (SoCC '20). Association for Computing Machinery, New York, NY, USA, 477–491. <https://doi.org/10.1145/3419111.3421285>
- [5] Zhi Tian, Weilin Huang, He Tong, Pan He, and Yu Qiao. 2016. Detecting Text in Natural Image with Connectionist Text Proposal Network, Vol. 9912. 56–72. https://doi.org/10.1007/978-3-319-46484-8_4