Pratyush Agnihotri, Boris Koldehofe, Carsten Binnig, Manisha Luthra. PANDA: performance prediction for parallel and dynamic stream processing. In Proceedings of the 16th ACM International Conference on Distributed and Event-Based Systems (DEBS'22), pp. 180–181, ACM press

PANDA: Performance Prediction for Parallel ANd Dynamic StreAm Processing

Pratyush Agnihotri
Technical University of Darmstadt

Boris Koldehofe
University of Groningen

Carsten Binnig
Technical University of Darmstadt

Manisha Luthra
Technical University of Darmstadt

ABSTRACT

Distributed Stream Processing (DSP) systems highly rely on *parallelism* mechanisms to deliver high performance in terms of latency and throughput. Yet the development of such parallel systems altogether comes with numerous challenges. In this paper, we focus on *how* to select appropriate resources for parallel stream processing under the presence of highly *dynamic* and *unseen* workloads. We present PANDA that provides a novel learned approach for highly efficient and parallel DSP systems. The main idea is to provide accurate resource estimates and hence optimal *parallelism degree* using zero-shot cost models to ensure the performance demands.

CCS CONCEPTS

• **Computer systems organization → Real-time systems**.

KEYWORDS

Parallel stream processing, Zero-shot cost models

1 INTRODUCTION

Why Parallel Stream Processing is Important? Distributed Stream Processing (DSP) is a paradigm that allows the processing of a high volume of infinite data streams and delivers continuous results to the end-users or applications of these systems. Today, various real-world applications build on DSP for their core operations. For instance, this year's DEBS Grand challenge requires triggering pattern detection queries timely for a financial trading company, Infront Financial Technology. Infront processes around *24 billion* events on average everyday, which translates to roughly 300,000 events per second [1]. One of the core mechanisms of DSP for meeting such high performance requirements is to split the data and use resource *parallelism* to deal with high loads. As such, most prominent from real-world systems at Alibaba that uses Blink [2] implement parallelism in their DSP applications to efficiently manage the high workload requirements of its application.

Yet designing such massively parallel systems comes with major challenges. One of the core challenges is *how* to select appropriate resources for parallelism? Here, the decision highly depends on the characteristics of the resource type, but also the characteristics of the workload (data stream and query) being executed on such resources. Thus, accurately modelling the performance of streaming queries with different degrees of parallelism on a wide range of resources is a major challenge especially for streaming applications where workload changes are very common.

Existing Solutions and Their Pitfalls. Several works aim at addressing this issue by proposing prediction methods for parallelism degree [3]. Recently, machine learning have been applied in these resource management decisions, that has achieved promising results [4, 5]. However, these learned approaches are highly *workload-driven*, i.e., they fail to perform well on *new* workload or even when a workload changes at runtime. Supporting accurate resource management for dynamic or unseen workloads at runtime is highly crucial for proper functioning of DSP systems as, consequently, over- or under-provisioning can lead to either wasted resources or inferior performance.

A prominent direction that has shown promising initial results, which works for unseen workloads are *zero-shot cost models* [6] that aim to predict costs of a continuous query on the placed resources. The main promise of zero-shot models is to (once) train on a variety of workloads with so-called *transferable features* that allows the model to generalize across streaming workloads. However, currently the zero-shot models are limited to the task of cost-estimation, in particular, latency and throughput prediction. Thus, it is unclear how these zero-shot cost models can solve other optimization tasks of DSP systems like predicting parallelism degree of resources.

Accurate Resource Estimates by PANDA. In this paper, we present PANDA that addresses the aforementioned challenge, by proposing a *learned* approach for parallelism prediction by leveraging zero-shot cost models [6, 7] that can deal with unseen workloads. The end goal of PANDA is to guarantee the performance and elasticity requirements of DSP applications using accurate estimates based on our learned approach. While we initially focus on CPU-based hardware only, in future we believe that we can support accurate performance predictions also for stream processing on heterogeneous hardware such as CPUs and GPUs. In the remaining sections, we first detail on our approach, challenges and present preliminary experiments addressing the first challenge.

2 OUR APPROACH: PANDA

The main goal of PANDA is to provide a new learned approach for accurate performance prediction for parallel execution in DSP. Figure 1 (left) presents the system model that PANDA is based on. Overall, PANDA aims to support performance predictions for parallel stream processing with different mechanisms: *task*- and *data*-parallel. The main idea of leveraging the zero-shot cost models is that the model is able to accurately predict cost metrics such as throughput or latency under varying degrees of parallelism for all these mechanisms. However, extending zero-shot models for predictions in parallel stream processing is non-trivial.

As a first challenge, zero-shot model architectures need to be extended to predict cost for parallel stream processing. In its core, the zero-shot cost model uses a graph structure with nodes representing operators, producers and consumers with their *transferable features* such as predicate types, selectivities of the operators and tuple-width of the data stream [6]. In addition, we now feed the zero-shot cost model with parallelism degree of an operator as an additional transferable feature such that the model is able to learn from correlations of the given degree and the costs. Another key addition is that PANDA encodes hardware related features such as

Pratyush Agnihotri, Boris Koldehofe, Carsten Binnig, Manisha Luthra. PANDA: performance prediction for parallel and dynamic stream processing. In Proceedings of the 16th ACM International Conference on Distributed and Event-Based Systems (DEBS'22), pp. 180–181, ACM press

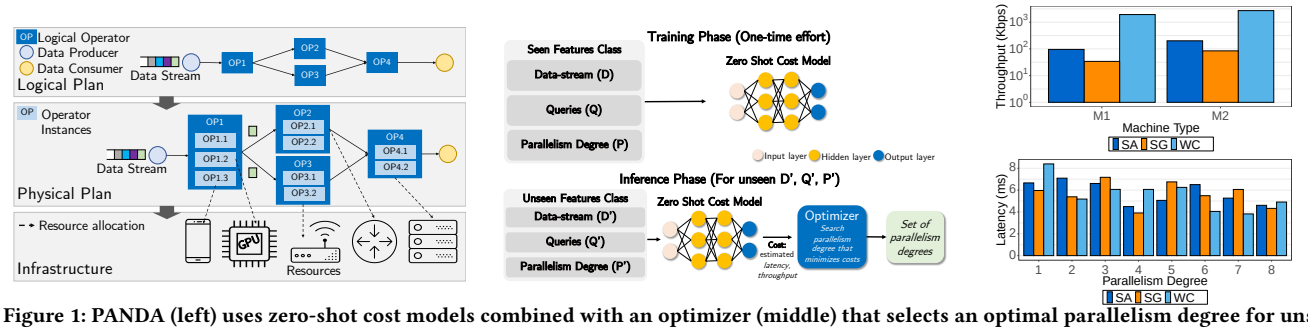

Figure 1: PANDA (left) uses zero-shot cost models combined with an optimizer (middle) that selects an optimal parallelism degree for unseen workloads. Our initial results show a high influence of parallelism on the performance (right) for different workloads.

number of cores of the placed resource, as this is directly related to the parallelism effect. This common graph representation allows the zero-shot cost models to generalize across workloads, queries and parallelism.

As a second challenge, for training a zero-shot model new training data needs to be collected that involves different query plans, different input data streams as well as using a varying degree of the operators within the plan. For this training data, the corresponding cost metrics need to be collected and can then be used for training the zero-shot cost model (cf. Figure 1 (middle) training phase).

The third challenge is how to integrate such a model into DSP. Once a new unseen logical plan is submitted, the task manager uses the cost model of PANDA to map operators to resources based on the predicted costs given by the zero-shot cost models. Thus during the inference phase (cf. Figure 1 (middle) inference), PANDA uses the predicted costs and an optimizer such as an ILP solver to find a set of optimal parallelism degrees for the given query plan such that the costs are minimized (or maximized). In future, we aim to use specialized parallel hardware architectures such as GPUs as another resource type that will be used for deployment. Another, key improvement we want to aim for is an end-to-end learned zero-shot model that is able to predict the parallelism degree directly without the need of a separate optimizer component.

3 PRELIMINARY EXPERIMENTS

In our preliminary experiments, we benchmark PANDA that is built on top of Flink to evaluate the influence of parallelism on the performance of DSP system with different hardware types.

Setup and Benchmark. We evaluate using three streaming workloads: *SA* - Twitter Sentiment Analysis, *SG* - DEBS Smart Grid, *WC* - Word Count, with a total of 12 GiB data. For benchmarking, we use two different machine architectures, processing speed, memory and parallel cores *(i)* M1: Intel Core i5-4200M CPU @ 2.50GHz machine with 4GB RAM and 2 cores and *(ii)* M2: Intel Core i7-3520M CPU @ 2.90GHz with 16GB RAM 4 CPU cores and NVIDIA GF108M GPU (96 cores). We used a key-based parallelism strategy of Apache Flink for creating multiple instances of query operators. As metrics, we use end-to-end latency and throughput at the consumer.

Initial Results. In Figure 1 (right), we report initial results on the impact of the parallelism degree and different hardware resources on the performance. In the results, we can clearly see a decreasing trend of end-to-end latency when running the different workloads on machine M2 with increasing parallelism degree. A similar trend is observed for throughput on different machines M1 and M2 with different amount of cores and processing speed. For the throughput evaluation (note the log scale), we take the respective parallelism degree as the amount of cores for better usage of parallelism. We observe that for different workloads and hardware characteristics a clear impact of parallelism degree on the performance, i.e., with better parallelism degree we observe better results and vice versa. Another important observation is that for different workloads, the performance changes significantly differently, which the zero-shot model has to learn and capture for the prediction approach. Moreover, a completely *new* workload such as a streaming join query may result in completely different performance in comparison to these benchmarks, and how to predict parallelism under these changes is our envisioned *key* contribution. With the learned approach, we aim to study this impact and use our zero-shot cost modelling approach to provide accurate estimates on parallelism degrees for an overall better performance of the DSP system.

Acknowledgements. This work is co-funded by the German Research Foundation (DFG) in the Collaborative Research Center (CRC) 1053 - MAKI.

REFERENCES

[1] S. Frischbier, M. Paic, A. Echler, and C. Roth, "Managing the complexity of processing financial data at scale - an experience report," in *CSDM*, 2019, pp. 14–26.
[2] X. Jiang, "Blink: How Alibaba Uses Apache Flink," https://www.ververica.com/blog/blink-flink-alibaba-search, 2021, [Online; accessed 27-05-2022].
[3] X. Liu and R. Buyya, "Resource management and scheduling in distributed stream processing systems: a taxonomy, review, and future directions," *in CSUR*, vol. 53, pp. 1–41, 2020.
[4] R. Mayer, B. Koldehofe, and K. Rothermel, "Predictable low-latency event detection with parallel complex event processing," *in IEEE IoTJ*, vol. 2, pp. 274–286, 2015.
[5] V. Cardellini, F. L. Presti, M. Nardelli, and G. R. Russo, "Decentralized self-adaptation for elastic data stream processing," *in FGCS*, vol. 87, pp. 171–185, 2018.
[6] R. Heinrich, M. Luthra, H. Kornmayer, and C. Binnig, "Zero-shot cost models for distributed stream processing," in *DEBS*, 2022, accepted for publication.
[7] B. Hilprecht and C. Binnig, "Zero-shot cost models for out-of-the-box learned cost prediction," *arXiv preprint*, 2022.