DYNAMIC ADAPTATION IN STREAM PROCESSING SYSTEMS

Paris - France

8 January 2023

Daniel WLADDIMIRO

Directeur de thèse : Pierre SENS

Co-encadrants : Luciana ARANTES et Nicolas HIDALGO

LIP6 - Sorbonne Université, CNRS

Table of contents

Introduction

Context

Web 2.0

- High data volumes
- Dynamic behaviour of the data flow

Context

- Real-time processing of large data streams
  - Low-latency processing

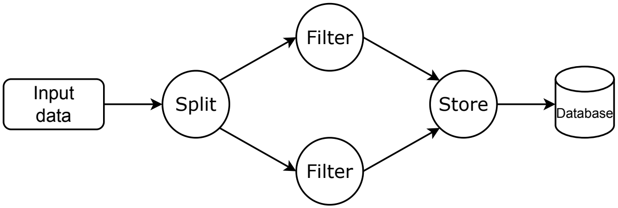
Applications

- Stock exchange prediction
- Network security monitoring
- Collecting information in natural disasters

Stream processing systems (SPS)

Logical architecture

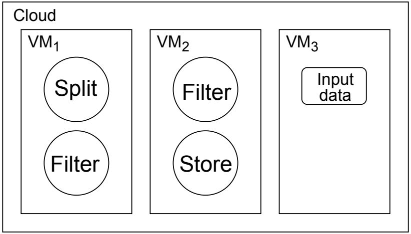
- The DAG defines the processing logic of the SPS
- A vertex represents a processing operator
- Unidirectional edges represent the data flow



Stream processing systems (SPS)

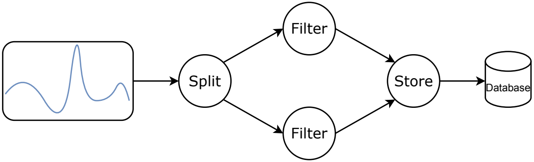
Physical architecture

- The DAG must be mapped to a physical environment
- Distributed platform : Cluster or Cloud



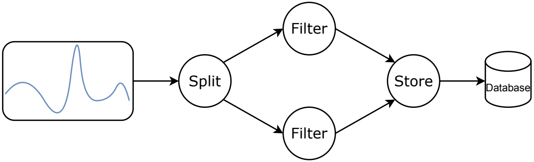
Existing SPS

Input rate can present traffic spikes or peaks



Existing SPS

Input rate can present traffic spikes or peaks



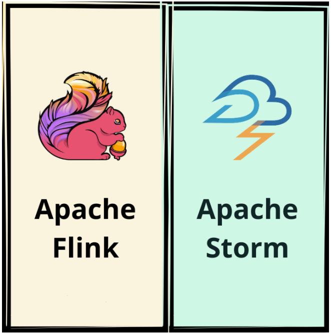
Situation

Overloaded operators and increased end-to-end latency

Existing SPS

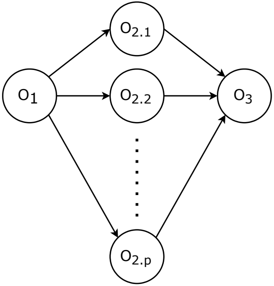
Existing SPS Frameworks:

- Apache Storm [Toshniwal et al. 2014]
- Apache Flink [Carbone et al. 2015]



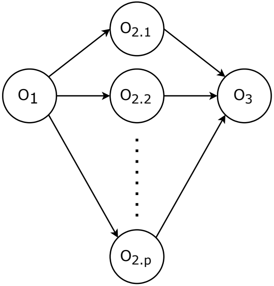
Existing SPS

Replication : Operators can be parallelised



Existing SPS

Replication : Operators can be parallelised



Situation

Overprovisioning or underprovisioning of replicas

Existing SPS

Problem

- Majority SPSs do not dynamically adapt the number of replicas according to input rate

Existing SPS

Problem

- Majority SPSs do not dynamically adapt the number of replicas according to input rate

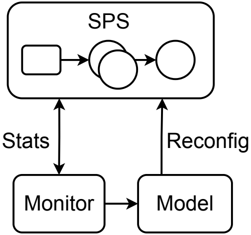
Solution

Automatically increase/decrease the number of replicas of critical operators

Existing Adaptive SPS

Existing Adaptive SPS

Adaptive SPS can modify the number of replicas

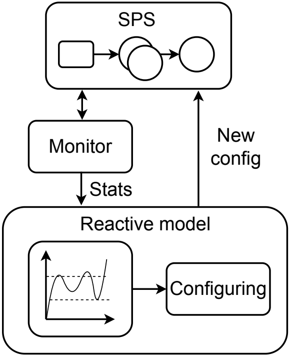


There are two approaches:

- Reactive approach
- Predictive approach

Reactive approach

- Statistics via a monitor
- Operator state analysis using thresholds



Reactive approach

Metrics:

- CPU [Gulisano et al. 2012]
- Latency [Madsen, Zhou, and Su 2016; Satzger et al. 2011; Heinze et al. 2014]
- Throughput [Kahveci and Gedik 2020; Russo et al. 2021; Gedik et al. 2014]

Reactive approach

Metrics:

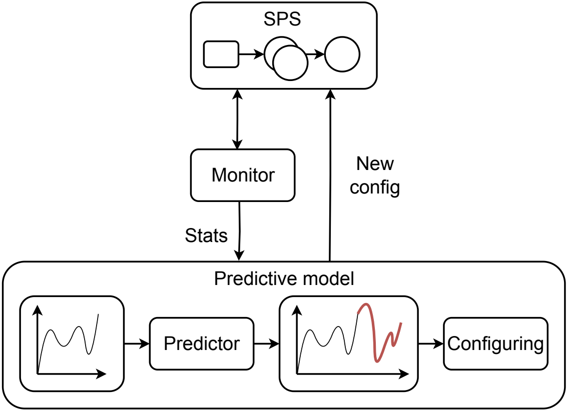
- CPU [Gulisano et al. 2012]
- Latency [Madsen, Zhou, and Su 2016; Satzger et al. 2011; Heinze et al. 2014]
- Throughput [Kahveci and Gedik 2020; Russo et al. 2021; Gedik et al. 2014]

Limitation

Most solutions consider a single metric

Predictive approach

- Based on prediction to determine the SPS reconfiguration
- Apply a predictor model



Related Work

Predictor models:

- Reinforcement learning [Cardellini et al. 2018]
- Time series [Kombi et al. 2019]
- Fuzzy logic [Mencagli, Torquati, and Danelutto 2018]
- ANN [Lombardi et al. 2018]

Related Work

Predictor models:

- Reinforcement learning [Cardellini et al. 2018]
- Time series [Kombi et al. 2019]
- Fuzzy logic [Mencagli, Torquati, and Danelutto 2018]
- ANN [Lombardi et al. 2018]

Limitation

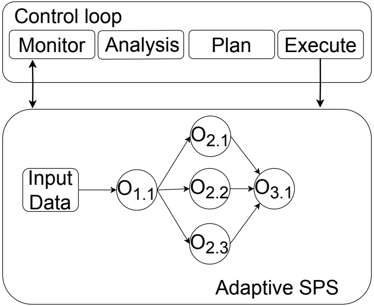
Predictive models are specific to an input rate or scenario

Our Adaptive SPS

Our Adaptive SPS

Our Proposal

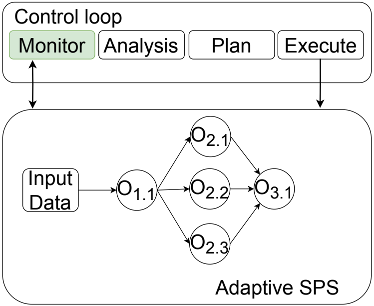
An adaptive SPS based on **MAPE model** whose aim is to **adapt the number of replicas** of the operators according to the peaks in the data stream.



Our Adaptive SPS

Our Proposal

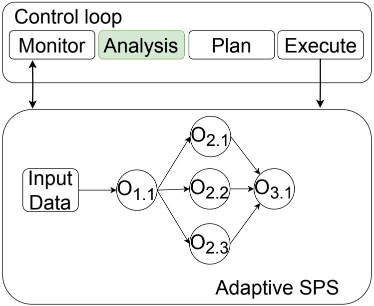
An adaptive SPS based on **MAPE model** whose aim is to **adapt the number of replicas** of the operators according to the peaks in the data stream.



Our Adaptive SPS

Our Proposal

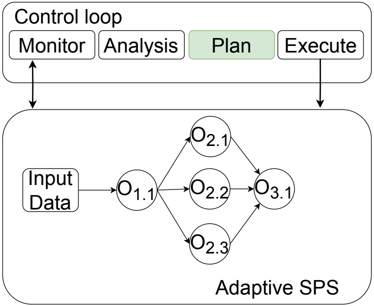
An adaptive SPS based on **MAPE model** whose aim is to **adapt the number of replicas** of the operators according to the peaks in the data stream.



Our Adaptive SPS

Our Proposal

An adaptive SPS based on **MAPE model** whose aim is to **adapt the number of replicas** of the operators according to the peaks in the data stream.

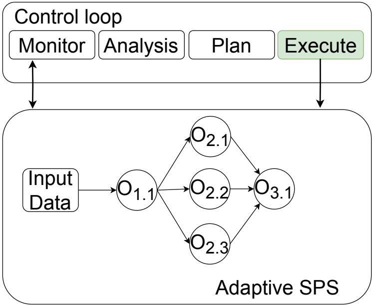


Our Adaptive SPS

SCIENCES SORBONNE UNIVERSITÉ

Our Proposal

An adaptive SPS based on **MAPE model** whose aim is to **adapt the number of replicas** of the operators according to the peaks in the data stream.



Our Adaptive SPS

Present two approaches:

- **Reactive approach** (RA-SPS): on-the-fly analysis of operator load
- **Predictive approach** (PA-SPS): predicting the number of replicas required

Our Adaptive SPS : Implementation

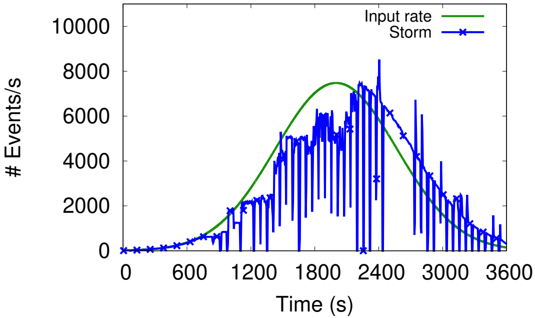
An extension of Apache Storm

APACHE STORM™

Storm limitation

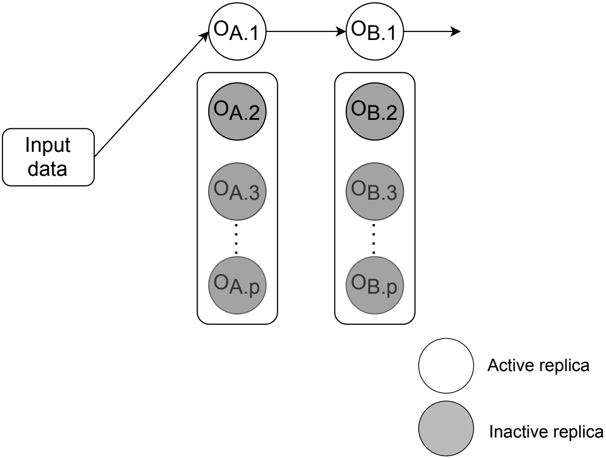
Limitation

Downtime in each reconfiguration



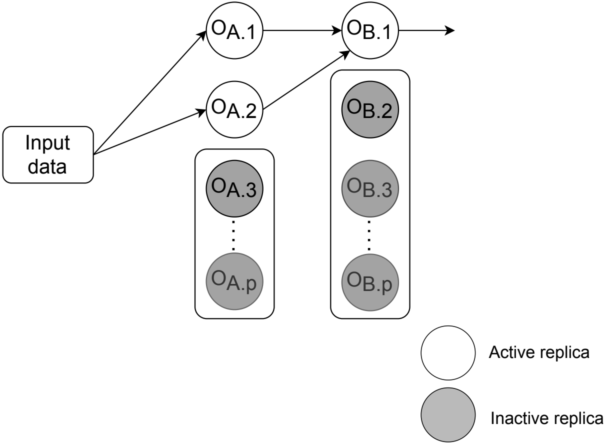
Our Adaptive SPS : Pool of replicas

Pool of pre-allocate replicas for each operator.



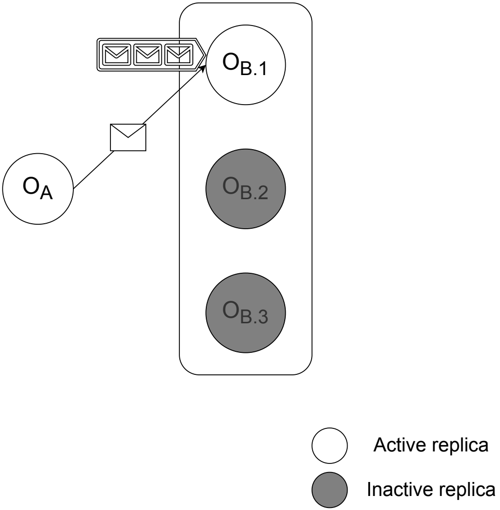
Our Adaptive SPS : Pool of replicas

Pool of pre-allocate replicas for each operator.



Storm limitation

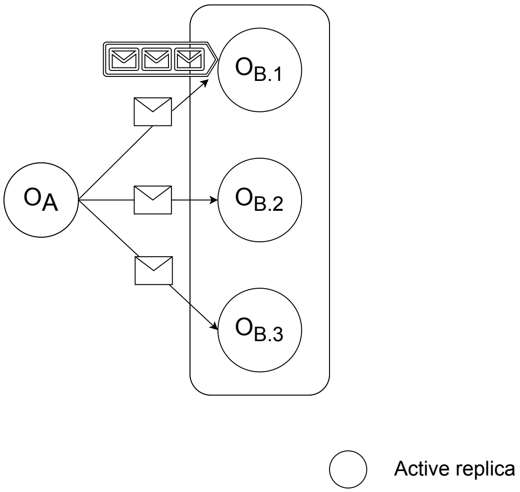
Shuffle grouping (Round-robin) among the active replicas of an operator



Storm limitation

Limitation

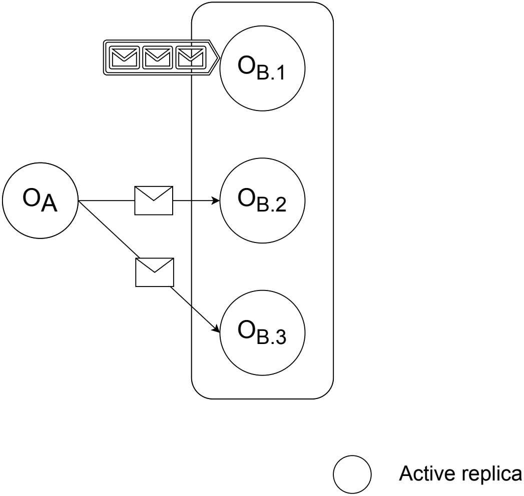
Load balancing among active replicas



Our Adaptive SPS : Load-Aware grouping

SCIENCES SORBONNE UNIVERSITÉ

Stream grouping for distributing the load among the active replicas of an operator



Our Adaptive SPS
Reactive approach (RA-SPS)

Reactive approach : RA-SPS

Proposal

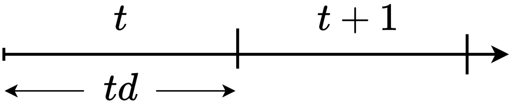
Analyse the **operator state** to modify the number of active replicas of the operators in the DAG

Reactive approach : RA-SPS

State metric δ is a multi-metric based in :

- Utilization (U) : Operator load
- Execution time (E) : Degradation execution time
- Queue (Q) : Impact of queue size

Each metric is discrete and calculated according to the time interval t



Reactive approach : RA-SPS

Utilisation metric

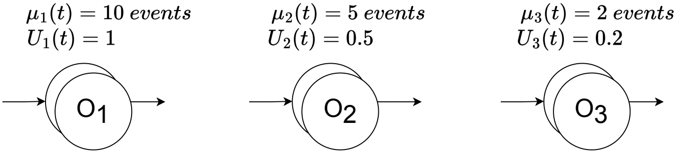
Percentage of time that the operator O_i is processing during t

$$U_i(t) = \frac{\mu_i(t) \times et_i(t)}{r_i(t) \times td}$$

Reactive approach : RA-SPS

Utilisation metric

$$U_i(t) = \frac{\mu_i(t) \times et_i(t)}{r_i(t) \times td}$$



$$et_i(t) = 200\ ms\ ;\ r_i(t) = 2\ ms\ ;\ td = 1000\ ms\ ;\ i \in [1, 2, 3]$$

Reactive approach : RA-SPS

Execution time metric

Execution time degradation of the operator O_i during t

$$E_i(t) = 1 - \frac{et_i}{et_i(t)}$$

et_i average execution time of one event by operator O_i according to the benchmark

Reactive approach : RA-SPS

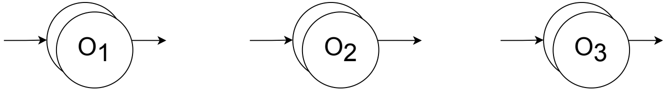
Execution time metric

$$E_i(t) = 1 - \frac{et_i}{et_i(t)}$$

$et_1(t) = 1000\ ms$
$E_1(t) = 0.9$

$et_2(t) = 200\ ms$
$E_2(t) = 0.5$

$et_3 = 100\ ms$
$E_3(t) = 0$



$$et_i = 100\ ms\ ;\ i \in [1, 2, 3]$$

Reactive approach : RA-SPS

Queue metric

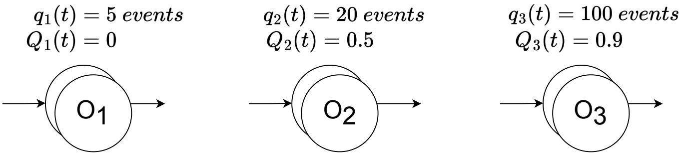
Impact of the queue size of the operator O_i according to the processing during t

$$Q_i(t) = 1 - \frac{\mu_i(t)}{q_i(t)}$$

Reactive approach : RA-SPS

Queue metric

$$Q_i(t) = 1 - \frac{\mu_i(t)}{q_i(t)}$$



$$\mu_i(t) = 10\ events\ ;\ i \in [1, 2, 3]$$

Reactive approach : RA-SPS

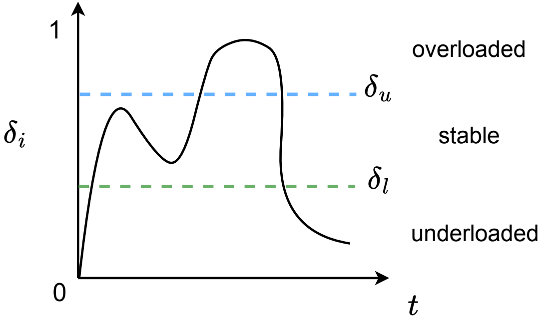
State metric $\delta_i(t)$ is determined by the three metrics and their respective weights

$$\delta_i(t) = U_i(t) \times \omega_U + Q_i(t) \times \omega_Q + E_i(t) \times \omega_E$$

Reactive approach : RA-SPS

Analysis of the operator state

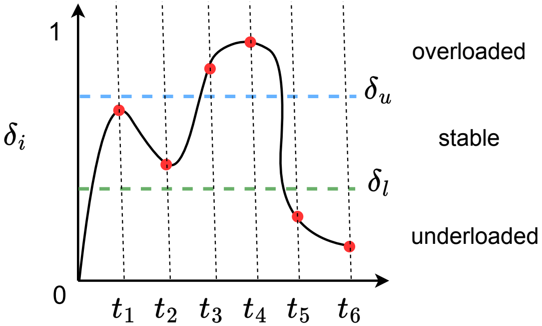
$\delta_i(t) < \delta_l$	underloaded
$\delta_l \leq \delta_i(t) \leq \delta_u$	stable
$\delta_i(t) > \delta_l$	overloaded



Reactive approach : RA-SPS

Planning algorithm

Analyse if the current state is equal to the precedent state

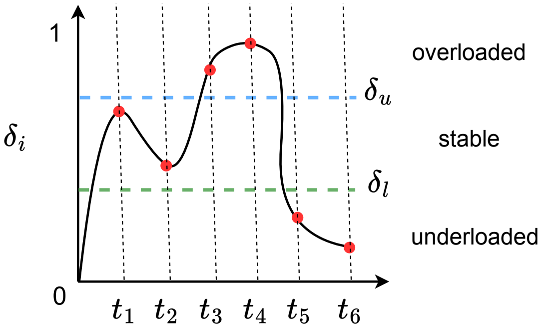


t_1

Reactive approach : RA-SPS

Planning algorithm

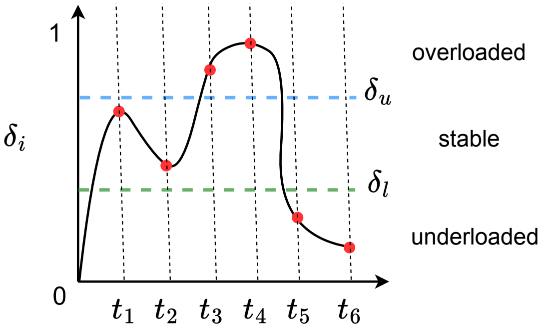
Analyse if the current state is equal to the precedent state



t_2 : State($\delta_i(t_2)$) = State($\delta_i(t_1)$) ?

Planning algorithm

Analyse if the current state is equal to the precedent state



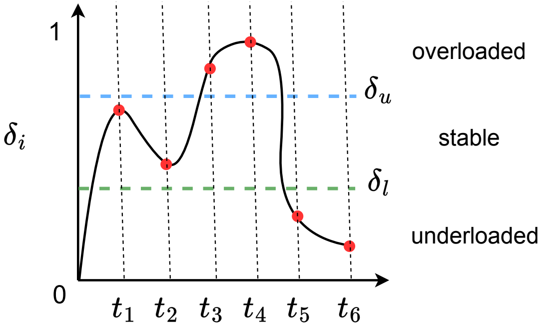
t_3 : State$(\delta_i(t_3))$ = State$(\delta_i(t_2))$?

Reactive approach : RA-SPS

SCIENCES SORBONNE UNIVERSITÉ

Planning algorithm

Analyse if the current state is equal to the precedent state

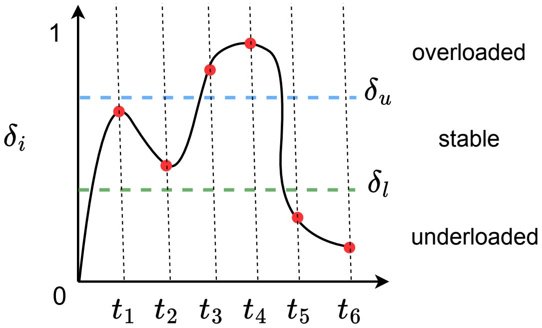


t_4 : State($\delta_i(t_4)$) = State($\delta_i(t_3)$) ? Is the machine overloaded ?
Activate k replicas of the operator O_i

Reactive approach : RA-SPS

Planning algorithm

Analyse if the current state is equal to the precedent state

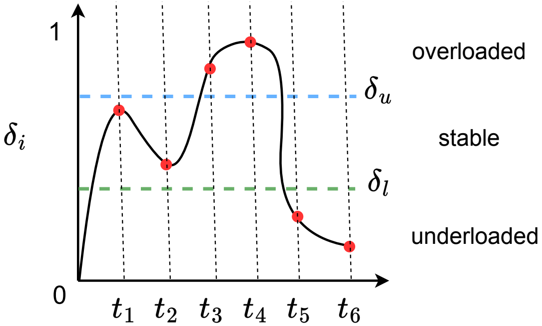


t_5 : $\text{State}(\delta_i(t_5)) = \text{State}(\delta_i(t_4))$?

Reactive approach : RA-SPS

Planning algorithm

Analyse if the current state is equal to the precedent state



t_6 : State($\delta_i(t_6)$) = State($\delta_i(t_5)$) ?

Deactivate k replicas of the operator O_i

Our Adaptive SPS
Predictive approach (PA-SPS)

Predictive Approach : PA-SPS

Proposal

Estimate the number of active replicas for operator according to the predicted input rate

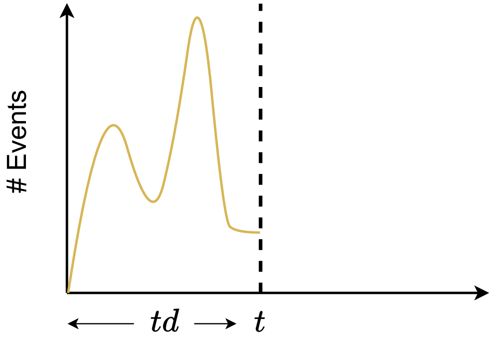
Predictive Approach : PA-SPS

Predictor model is based in :

- number of events received
- dependency among operators
- number of queued events

Predictive Approach : PA-SPS

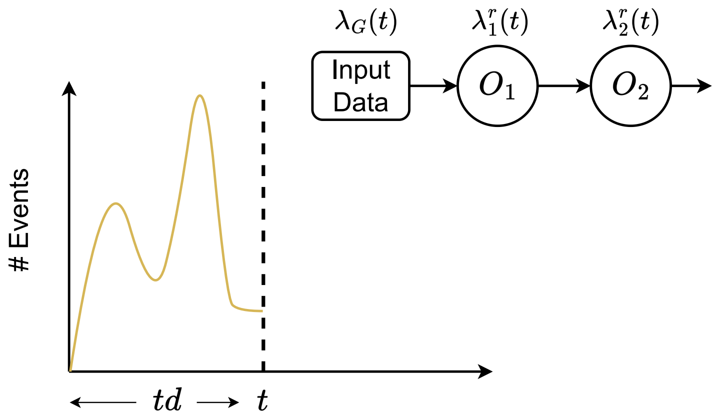
$\lambda_G(t)$: number of events sent by input data during t



Predictive Approach : PA-SPS

$\lambda_G(t)$: number of events sent by input data during t

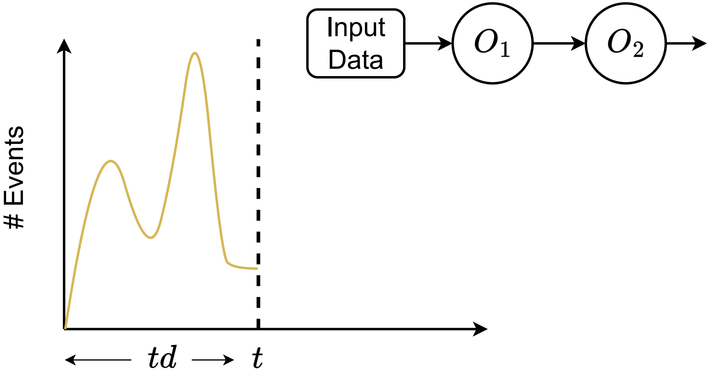
$\lambda_i^r(t)$: number of events received by O_i during t



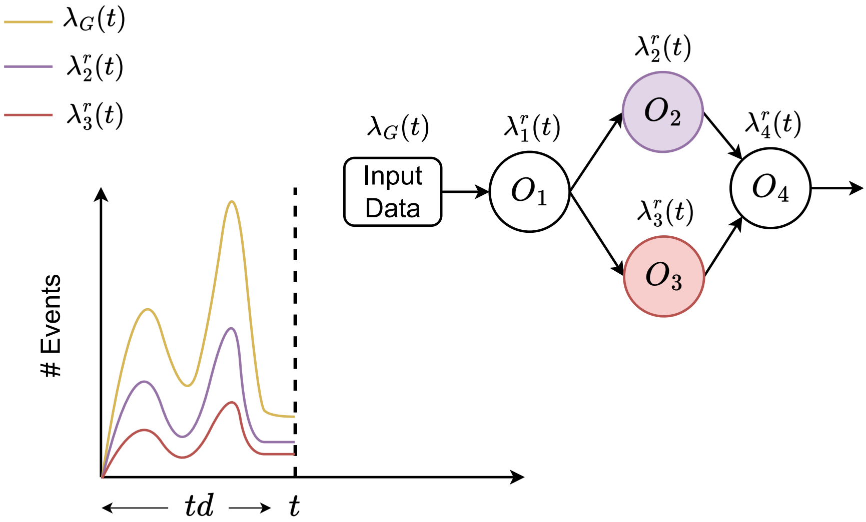
Predictive Approach : PA-SPS

- $\lambda_G(t)$: number of events sent by input data during t
- $\lambda_i^r(t)$: number of events received by O_i during t

$$\lambda_G(t) = \lambda_1^r(t) = \lambda_2^r(t)$$

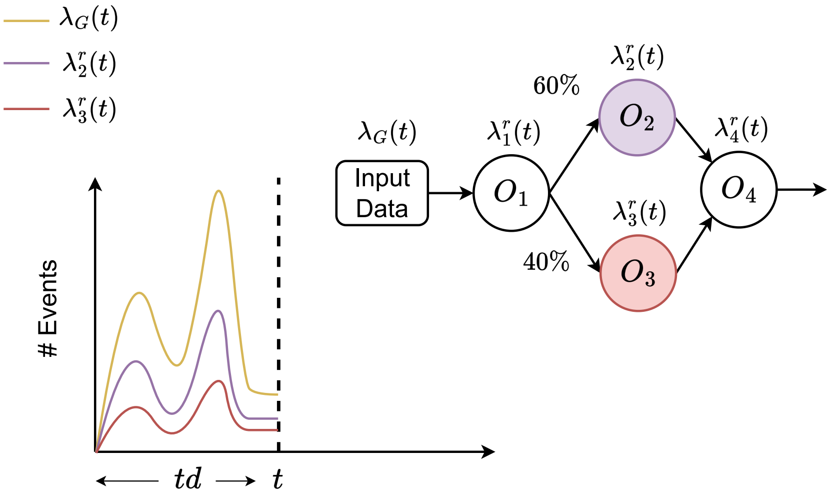


Predictive Approach : PA-SPS



Predictive Approach : PA-SPS

$\theta_i(t)$: percentage of events processed of $\lambda_G(t)$ during t

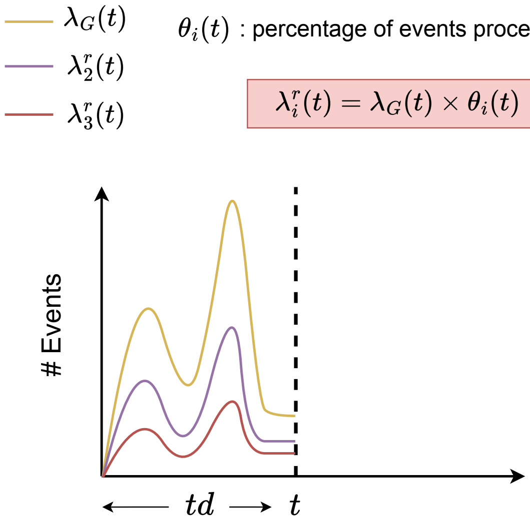


Predictive Approach : PA-SPS

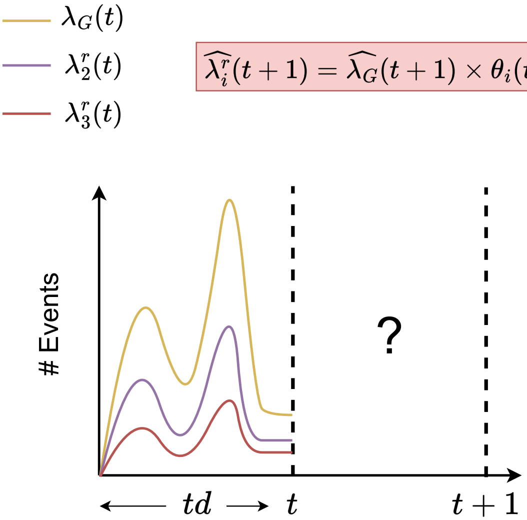
SCIENCES SORBONNE UNIVERSITÉ

$\theta_i(t)$: percentage of events processed of $\lambda_G(t)$ during t

$$\boxed{\lambda_i^r(t) = \lambda_G(t) \times \theta_i(t)}$$



Predictive Approach : PA-SPS



$$\boxed{\widehat{\lambda_i^r}(t+1) = \widehat{\lambda_G}(t+1) \times \theta_i(t)}$$

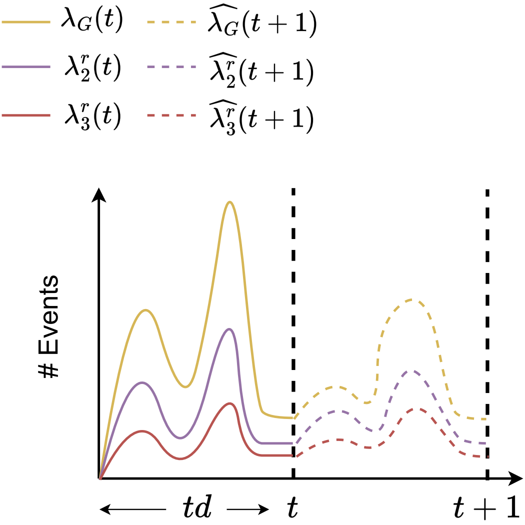
Predictive Approach : PA-SPS

Predictive model is used to predict the number of events sent by the input data during the next time interval

Predictive models applied:

- Basic
- Linear regression
- Fast Fourier Transform
- Artificial Neural Network
- Random Forest

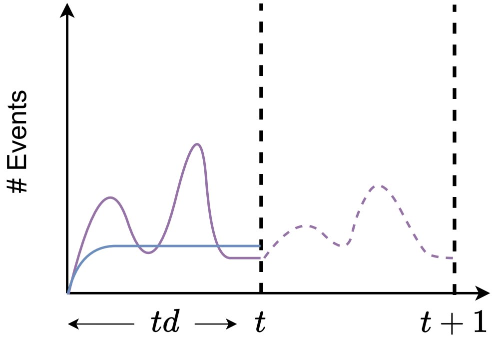
Predictive Approach : PA-SPS



Predictive Approach : PA-SPS

$\lambda_2^r(t)$ $\widehat{\lambda_2^r}(t+1)$

$\mu_2(t)$: number of events processed by O_i during t



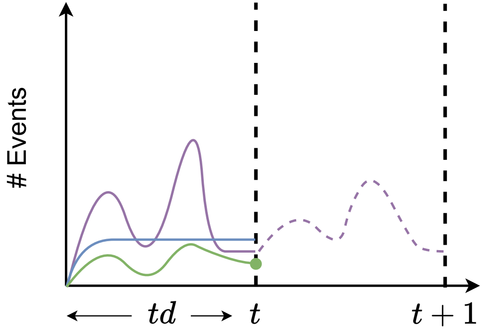
Predictive Approach : PA-SPS

SCIENCES SORBONNE UNIVERSITÉ

$\lambda_2^r(t)$ $\widehat{\lambda_2^r}(t+1)$

$\mu_2(t)$

$q_2(t)$: queue of events received and not processed by O_i **at the end of** t



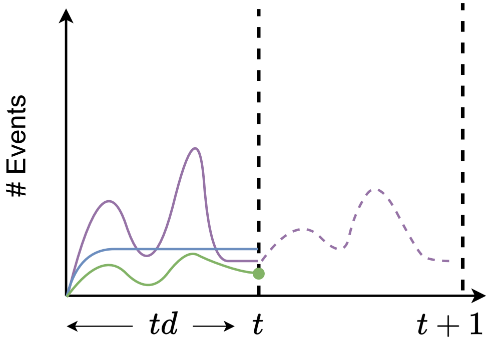
Predictive Approach : PA-SPS

$\lambda_2^r(t)$ $\widehat{\lambda_2^r}(t+1)$

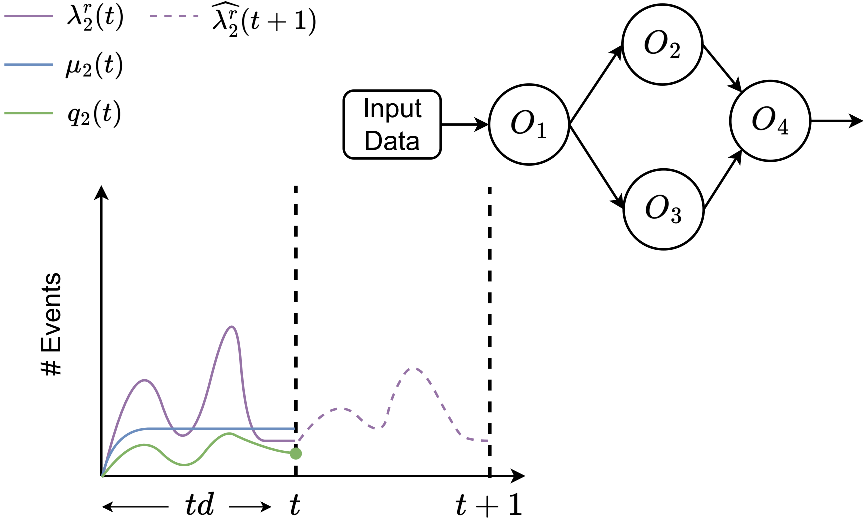
$\mu_2(t)$

$q_2(t)$

$$\widehat{\lambda_i^q}(t+1) = |q_i(t)|$$



Predictive Approach : PA-SPS



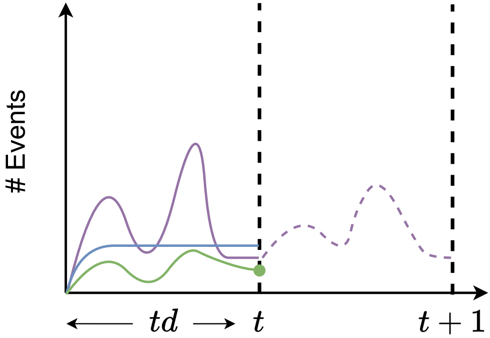
Predictive Approach : PA-SPS

$\lambda_2^r(t)$ $\widehat{\lambda_2^r}(t+1)$

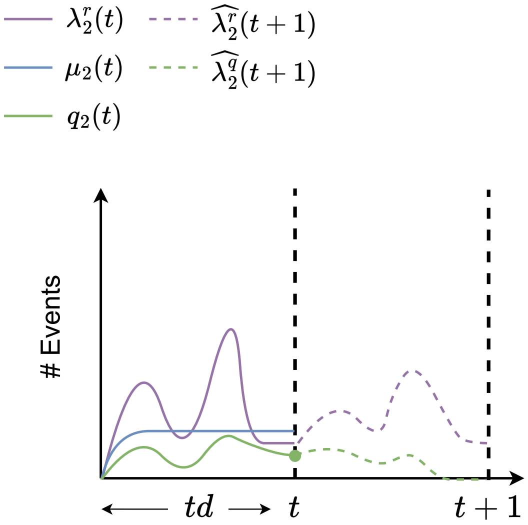
$\mu_2(t)$

$q_2(t)$

$$\widehat{\lambda_i^q}(t+1) = |q_i(t)| + \sum_{p \in pred(O_i)} \widehat{\lambda_p^q}(t+1) \times \theta_p(t)$$



Predictive Approach : PA-SPS



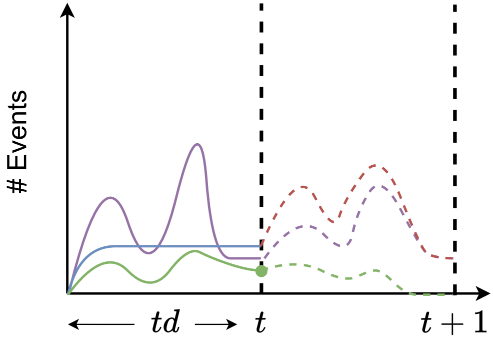
Predictive Approach : PA-SPS

$\lambda_2^r(t)$ $\widehat{\lambda_2^r}(t+1)$

$\mu_2(t)$ $\widehat{\lambda_2^q}(t+1)$

$q_2(t)$ $\widehat{\lambda_2}(t+1)$

$$\boxed{\widehat{\lambda_i}(t+1) = \widehat{\lambda_i^r}(t+1) + \widehat{\lambda_i^q}(t+1)}$$



Predictive Approach : PA-SPS

Prediction of number of active replicas

$$r_i(t+1) = \frac{\widehat{\lambda}_i(t+1) \times et_i}{td}$$

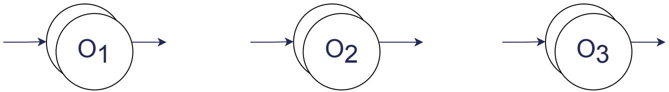
PA-SPS : Prediction of active replicas

Planning algorithm

$$r_i(t+1) = \frac{\widehat{\lambda_i}(t+1) \times et_i}{td}$$

$\widehat{\lambda_1}(t+1) = 10\ events$ $\quad$ $\widehat{\lambda_2}(t+1) = 10\ events$ $\quad$ $\widehat{\lambda_3}(t+1) = 10\ events$

$et_1 = 200\ ms$ $\quad$ $et_2 = 100\ ms$ $\quad$ $et_3 = 400\ ms$



$r_1(t) = 2$ $\quad$ $r_2(t) = 2$ $\quad$ $r_3(t) = 2$

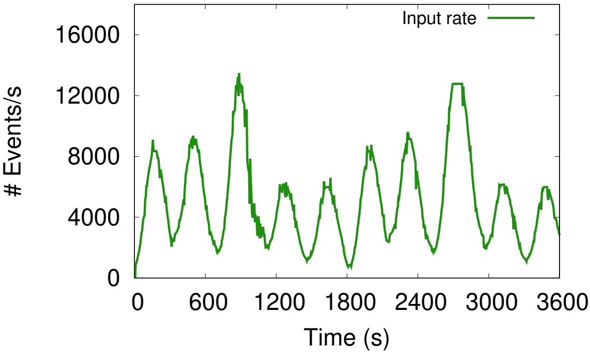
$r_1(t+1) = 2$ $\quad$ $r_2(t+1) = 1$ $\quad$ $r_3(t+1) = 4$

$$td = 1000 ms$$

Experiments

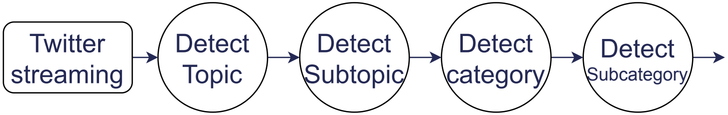
Environment : Input data scenario

A traffic model from real Twitter data related to COVID pandemic



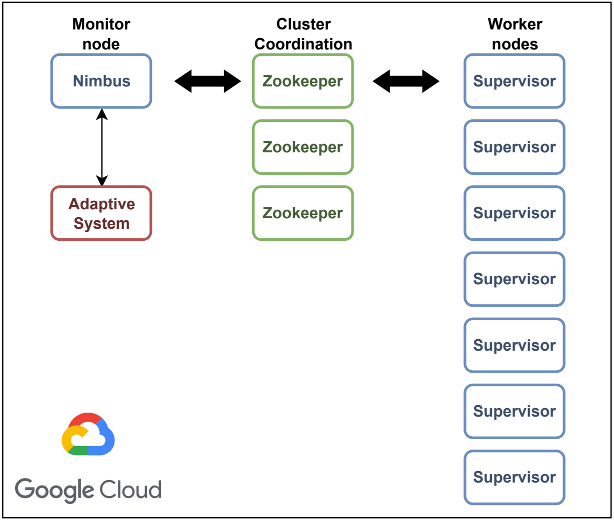
Environment : Application

Twitter linear app to classify tweets



Environment : Infrastructure

Google Cloud Platform as infrastructure for the deployment



Evaluation metrics

Saved nodes *[Lombardi et al. 2018]*

- Proportion of resources saved with respect to a statically over-provisioned configuration

Throughput degradation *[Lombardi et al. 2018]*

- Difference between the input rate and the output rate

Latency

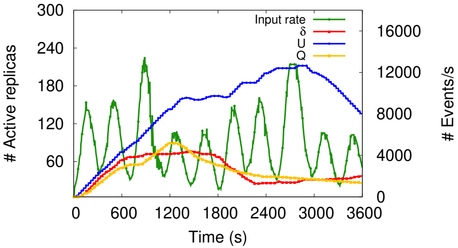
- Average time taken by an event between the moment it enters and leaves the SPS

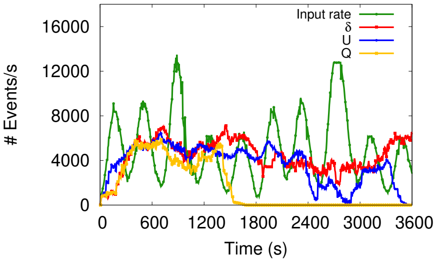
Difference in the number of processed events

- Difference between the total number of processed events and the total number of received events

Performance Results
Reactive approach (RA-SPS)

Performance Results : Reactive approach (RA-SPS)





- Q queues a lot of events
- U overprovides active replicas
- δ is more stable

Performance Results : Reactive approach (RA-SPS)

SCIENCES SORBONNE UNIVERSITÉ

Metric	Saved Nodes	Throughput Degradation	Diff. Proc. Events	Latency (ms)
δ	**0.3996**	**0.1092**	**0.8907**	**39687.51**
U	-0.8934	0.2597	0.7402	23441.39
Q	0.4975	0.6830	0.3169	28799.60

- δ process more events that U and Q
- δ does not consider dependency, so there is cascading problem

Performance Results
Predictive approach (PA-SPS)

Performance Results : Predictive approach (PA-SPS)

SCIENCES SORBONNE UNIVERSITÉ

Pred. Model	Saved Resources	Throughput Degradation	Diff. Proc. Events	Latency (ms)
ANN	**0.475**	**0.070**	**1.000**	**355.490**
FFT	0.519	0.189	1.000	1023.380
LR	0.533	0.195	1.000	663.030
RF	0.538	0.227	0.996	583.921
Basic	0.560	0.325	1.000	1295.490

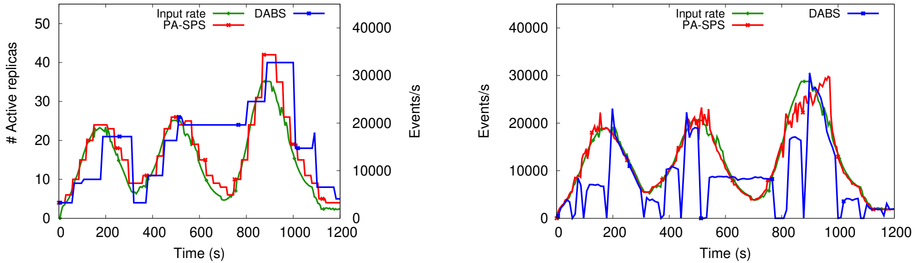
- ANN has a low latency and is more stable, but use more resource
- Trade-off : Resource vs Performance

Performance Results : Predictive approach (PA-SPS)

Comparison between PA-SPS and DABS

- PA-SPS :
  - Uses ANN as a predictor model
- DABS :
  - An extension of Storm
  - Uses a predictor model based on regressions

Performance Results : Predictive approach (PA-SPS)



- DABS restarts many times the application
- PA-SPS is more stable

Performance Results : Predictive approach (PA-SPS)

Adaptive SPS	Saved Resources	Throughput Degradation	Diff. Processed Events	Latency (ms)
PA-SPS	0.475	0.071	1.000	355.490
DABS	0.3962	0.2849	0.8283	1391.28

- DABS has a less efficient adaptation in contrast to PA-SPS

Conclusion

Conclusion

Adaptive SPS extending Storm

- **Pool of replicas**
- **Load-aware grouping**

Two approaches:

- **Reactive (RA-SPS)** : Use of a multi-metric
- **Predictive (PA-SPS)** : Estimate number of active replicas using different predictive models

Future work

Dynamic adaptation of parameters

- RA-SPS uses different parameters, which can be adjusted by AI

Hybrid adaptive SPS

- Use the reactive and predictive model for SPS adaptation

Physical and logical adaptation

- Consider also the modification of virtual machines

Merci beaucoup !

Publications

SCIENCES SORBONNE UNIVERSITÉ

- [Wladdimiro et al. 2021] Daniel Wladdimiro, Luciana Arantes, Pierre Sens and Nicolas Hidalgo. "A Multi-Metric Adaptive Stream Processing System." In: NCA. IEEE, 2021.
- [Wladdimiro et al. 2022a] Daniel Wladdimiro, Luciana Arantes, Pierre Sens and Nicolas Hidalgo. "A predictive approach for dynamic replication of operators in distributed stream processing systems" In: SBAC. IEEE, 2022.
- [Wladdimiro et al. 2023a] Daniel Wladdimiro, Luciana Arantes, Pierre Sens and Nicolas Hidalgo. "PA-SPS: A Predictive Adaptive Approach for an Elastic Stream Processing System" In: JPDC. 2023. [Minor Revision]
- [Wladdimiro et al. 2022b] Daniel Wladdimiro, Luciana Arantes, Pierre Sens and Nicolas Hidalgo. "A predictive model for Stream Processing System that dynamically calibrates the number of operator replicas." In: ComPAS. Amiens, France, 2022.
- [Wladdimiro et al. 2023b] Daniel Wladdimiro, Luciana Arantes, Pierre Sens and Nicolas Hidalgo. "PRESPS: a PREdictive model to determine the number of replicas of the operators in Stream Processing Systems" In: ComPAS. Annecy, France, 2023.

References I

- Carbone, Paris et al. (2015). "Apache Flink™: Stream and Batch Processing in a Single Engine". In: *IEEE Data Eng. Bull.* 38.4, pp. 28–38.
- Cardellini, Valeria et al. (2018). "Decentralized self-adaptation for elastic Data Stream Processing". In: *Future Gener. Comput. Syst.* 87, pp. 171–185.
- Gedik, Bugra et al. (2014). "Elastic Scaling for Data Stream Processing". In: *IEEE Trans. Parallel Distrib. Syst.* 25.6, pp. 1447–1463.
- Gulisano, Vincenzo et al. (2012). "StreamCloud: An Elastic and Scalable Data Streaming System". In: *IEEE Trans. Parallel Distributed Syst.* 23.12, pp. 2351–2365.
- Heinze, Thomas et al. (2014). "Latency-aware elastic scaling for distributed data stream processing systems". In: *DEBS*. ACM, pp. 13–22.
- Kahveci, Basri and Bugra Gedik (2020). "Joker: Elastic stream processing with organic adaptation". In: *J. Parallel Distributed Comput.* 137, pp. 205–223.

References II

Kombi, Roland Kotto et al. (2019). "DABS-Storm: A Data-Aware Approach for Elastic Stream Processing". In: *Trans. Large Scale Data Knowl. Centered Syst.* 40, pp. 58–93.

Lombardi, Federico et al. (2018). "Elastic Symbiotic Scaling of Operators and Resources in Stream Processing Systems". In: *IEEE Trans. Parallel Distrib. Syst.* 29.3, pp. 572–585.

Madsen, Kasper Grud Skat, Yongluan Zhou, and Li Su (2016). "Enorm: efficient window-based computation in large-scale distributed stream processing systems". In: *DEBS*. ACM, pp. 37–48.

Mencagli, Gabriele, Massimo Torquati, and Marco Danelutto (2018). "Elastic-PPQ: A two-level autonomic system for spatial preference query processing over dynamic data streams". In: *Future Gener. Comput. Syst.* 79, pp. 862–877.

Russo, Gabriele Russo et al. (2021). "MEAD: Model-Based Vertical Auto-Scaling for Data Stream Processing". In: *CCGRID*. IEEE, pp. 314–323.

References III

Satzger, Benjamin et al. (2011). "Esc: Towards an Elastic Stream Computing Platform for the Cloud". In: *IEEE CLOUD*. IEEE Computer Society, pp. 348–355.

Toshniwal, Ankit et al. (2014). "Storm@ Twitter". In: *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, pp. 147–156.

Wladdimiro, Daniel et al. (2021). "A Multi-Metric Adaptive Stream Processing System". In: *NCA*. IEEE, pp. 1–8.

— (2022a). "A predictive approach for dynamic replication of operators in distributed stream processing systems". In: *SBAC-PAD*. IEEE.

— (2022b). "A predictive model for Stream Processing System that dynamically calibrates the number of operator replicas". In: *ComPAS*, pp. 1–8.

— (2023a). "PA-SPS: A Predictive Adaptive Approach for an Elastic Stream Processing System". In: *J. Parallel Distributed Comput.*

References IV

Wladdimiro, Daniel et al. (2023b). "PRESPS: a PREdictive model to determine the number of replicas of the operators in Stream Processing Systems". In: *ComPAS*, pp. 1–9.