DATA AND PROCESS ENGINEERING RESEARCH LAB AT KGU
COLLABORATIVE PEOPLE SYSTEMS
DATA & PROCESS ENGINEERING RESEARCH LAB AT KGU KYONGGI UNIVERSITY

HPC Asia 2024: International Conference on High Performance Computing
In Asia-Pacific Region, Jan. 25-27, 2024, Nagoya, JAPAN

KGU KYONGGI UNIVERSITY

A MapReduce-based Inter-Organizational and Distributed Process Log Clustering Framework

Thanh-Hai Nguyen *
nguyenthanhhai@kgu.ac.kr
Data and Process Engineering Research Lab.
Department of Computer Science
Graduate School of KYONGGI UNIVERSITY
Suwon-si, Gyeonggi-do, South Korea

Eun-Bee Cho & Batt Chaya †
{eunbee0508, tsbaachka95}@kgu.ac.kr
Data and Process Engineering Research Lab.
Department of Public Safety Bigdata
Graduate School of KYONGGI UNIVERSITY
Suwon-si, Gyeonggi-do, South Korea

Sang-Eun Ahn ¶
sangeun1804@kgu.ac.kr
Contents Convergence Software R.I.
KYONGGI UNIVERSITY
Suwon-si, Gyeonggi-do, South Korea

Dinh-Lam Pham ‡
phamdinhlam@kgu.ac.kr
Contents Convergence Software R,I,
KYONGGI UNIVERSITY
Suwon-si, Gyeonggi-do, South Korea

Kyoung-Sook Kim §
khmjmc@kgu.ac.kr
Contents Convergence Software R.I.
KYONGGI UNIVERSITY
Suwon-si, Gyeonggi-do, South Korea

Kwanghoon Pio Kim ‖
kwang@kgu.ac.kr
Data and Process Engineering Research Lab.
Division of AI Computer Science and Engineering
KYONGGI UNIVERSITY
Suwon-si, Gyeonggi-do, South Korea

ABSTRACT

In this paper, we propose a distributed process log clustering framework that collects and classifies the distributed process execution event logs recorded by the distributed operations of the inter-organizational business process management system. The proposed framework is implemented onto the MapReduce-based distributed processing framework and applied to the SICN-oriented process mining system. Note that it ought to be operable and applicable as a preprocessing tool of the process mining and deep learning frameworks and systems. We call the collected datasets of the distributed process event logs as the process BIG-Logs. especially. The framework proposed and implemented in this paper undertakes the splitting, mapping, shuffling, and reducing operations of the MapReduce's preprocessing functionality, and it is embedded into the SICN-oriented process mining system as one of the essential components of a specific process mining algorithm, which is the gradual ρ-Algorithm, and a predictive process monitoring algorithm to be developed in the authors' research group. We assume that the process BIG-Logs are formatted in the IEEE XES standard data format and recorded from executing all the fragmented workcases instantiated from an inter-organizational business process model. Also, we assume that the underlying inter-organizational business process model is defined by the structured information control net process modeling methodology, and that the fragmentation approach is done in vertical, horizontal, or hybrid.

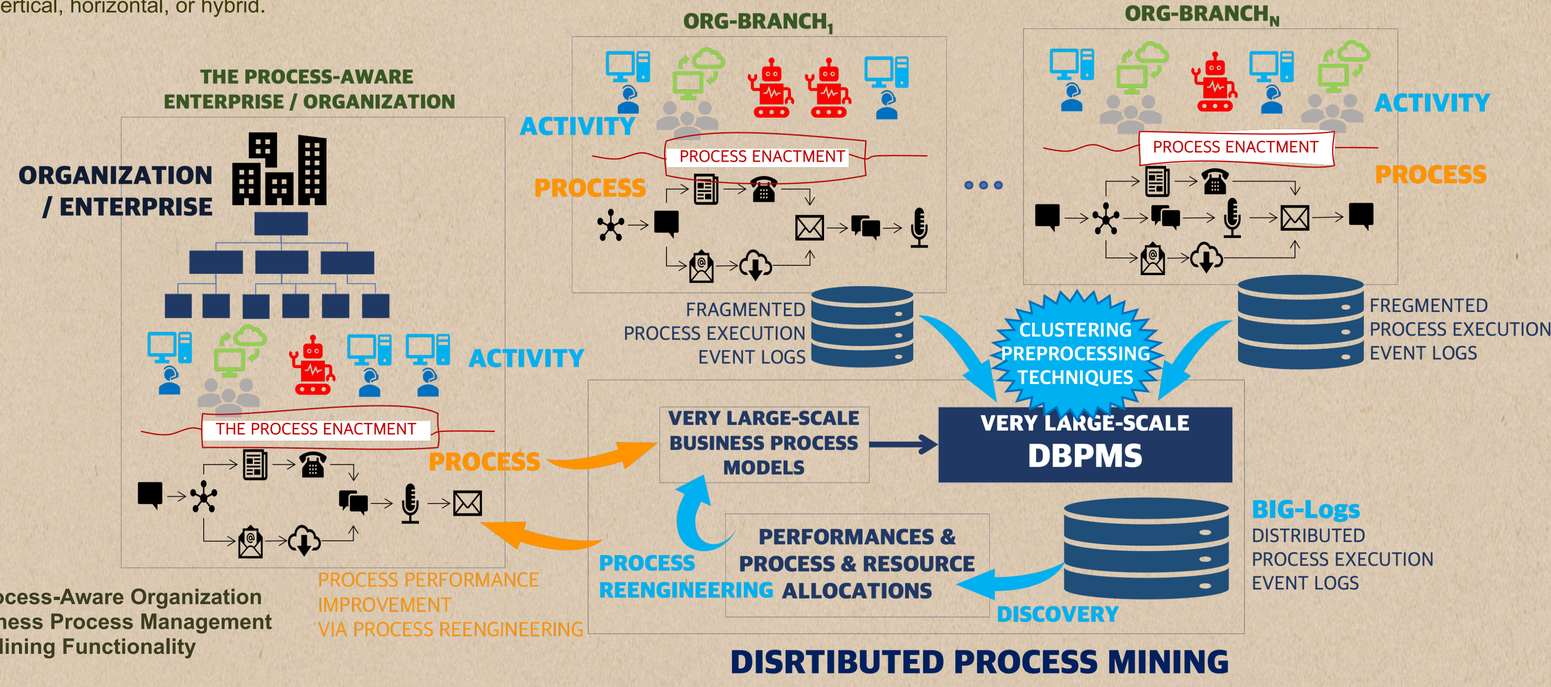

Fig. 1. A Situational View of the Process-Aware Organization Supported by the Distributed Business Process Management System with Distributed Process Mining Functionality

1 Research Goal and Scope

With regarding to the workflow and business process automation technologies on process-aware enterprises and organizations, a recent requirement is surely **the emerging concept of process fidelity** [1][2]. The process fidelity implies a sort of experimental mining studies [1][3][5] for measuring the discrepancies between the planned process model at build-time and the enacted process model at run-time. It is also related with the core activity in the process life-cycle management that supports reengineering and redesigning those deployed processes models running on the process-aware enterprises and organizations. The technical solution for realizing the process fidelity concept is adopting the process mining platforms and systems that provide the process discovery functionality as well as the process rediscovery functionality exploring the control-flow knowledge from the archives of the process models and their execution event log histories.

Another recent requirement is about enacting **the large scale and inter-organizational business process models** [3]. The fidelity and life-cycle management of these large scale and inter-organizational process models can be maintained by the process mining functionality, too. The teaser figure of Fig. 1 is to illustrate a situational view of enacting the large scale and inter-organizational process models. Assume that a specific inter-organizational process model is deployed and enacted across all the distributed branches of the process-aware organization, and its execution history is recorded and managed as the valuable assets of Bigdata. As illustrated in the figure, the essential preprocessing function for adopting the process mining functionality is firstly to collect all the partial process execution event logs [2] that are vertically, horizontally, or hybrid fragmented and dispersed over all the branches of a large scale organization. The next is to rearrange such event logs collected to form traces of execution event sequences for the process instances (or workcases) spawned from a corresponding process model. And the last step of the preprocessing function is to classify these process execution event log traces into a group of trace patterns, each of which is holding the identical activity execution event sequence.

Conclusively, the main research challenge of this paper is to develop a process log clustering framework based upon the MapReduce platform [4], and verify its functional correctness by applying to a large scale and inter-organizational process enactment event log Bigdata that is made up of three fragmented process log datasets. The authors' research group has successfully developed the SICN-oriented process mining algorithm and system [2] that can eventually hook the implemented process log clustering framework proposed in this paper as the preprocessing function.

2 The MapReduce-based Process Log Clustering Algorithm and System

This section introduce the algorithmic description of the inter-organizational process log clustering framework. Especially, we adopt the MapReduce platform [4] for gathering and classifying the fragmented process logs distributed over all the branches of the process-aware inter-organization. Note that the MapReduce takes the list of records formed in a (Key, Value) pair, as inpput. The following is the pseudo-coded algorithm:

```
Require: A Group of Fragmented
Datasets of Inter-Organizational
Process Enactment Event Logs
in XES-Format, T
Ensure: A List of TWC-Patterns, P
Ensure: Occ. of TWCP, ∀Op(p ∈ P)
procedure MAIN(T)
   class ω-Mapper; class ω-Reducer;
end procedure
```

```
class ω-Mapper:
  Organizing key & value;
  method Map(fTWC-key a,fTWC-value t)
    for ∀ t do
      clustering p ← t;
      call Emit(pattern p, count weight);
    end for
```

```
class ω-Reducer:
  method Reduce(key p, values W)
    Op ← 0;
    for ∀ w ∈ W do
      Op ← Op + w;
      call Emit (pattern p, count Op);
    end for
```

The procedural classes of **ω-Mapper** and **ω-Reducer** takes, as input, a list of records organized with the temporal work-cases in a pair of **Key (fTWC-key, the line number)** and **Value (fTWC-value, the event trace)**. The method **Map( )** function reads the list of **(fTWC-key, fTWC-value)** pairs and puts the intermediate result out in the form of **(pattern-key, count-value)** through the method **Emit( )** function. These intermediate results consist of a list of **(pattern-key, count-values[ ])** pairs clustered with the same value based on pattern-key. The method **Reduce( )** function performs an aggregate operation on lists of **(pattern-key, pattern-values)** pairs, and returns the final result, that is, each pattern (temporal work-case pattern) and its occurrence. The total occurrence **(pattern *p*, count O*p*)** is output through the **Emit( )** function. Actually, prior to executing the method **Reduce( )** function, the mapper performs clustering on the intermediate results based on the pattern-key, which is achieved through sorting on the pattern-key values. Then, the **Combine( )** function is internally executed prior to delivery to each task that will perform the method **Reduce( )** function on the **(pattern-key, pattern-values[ ])** lists clustered with sorting. Note that the **Combine( )** function is intended to reduce the I/O cost required to transfer data from the mapper to the reducer by performing the aggregate operation in advance in the map-step.

3 Experimental Verification and Conclusion

As a consequence, we successfully implemented the inter organizational process log clustering algorithm and framework with operating upon the temporal work-case driven MapReduce platform, and plugged this implemented framework in the SICN-oriented process mining system [2][5] as the preprocessing function. By using the SICN-oriented process mining system, we carried out an verificational experiment with fragmenting the original Customer Summary process log dataset [1] into three groups of horizontally distributed temporal work-case traces. As illustrated in Fig. 2, we horizontally fragmented the Customer Summary process dataset, which is containing the total 43,808 temporal work-case traces, into three fragmented process event log datasets; These three horizontally fragmented process event log datasets contain 14,000, 14,000, and 15,808 temporal work-case traces, respectively. After fulfilling the mapping operations and the reducing operations of **ω-Mapper** and **ω-Reducer**, at last we successfully obtained a clustered process dataset, named as Process **BIG-Log**, holding the total 59 temporal work-case pattern-clusters and each pattern-cluster is containing a certain amount of the temporal work-case traces, all of which have the identical activity-event sequences. Note that the top five temporal work-case pattern-clusters, in terms of the holding number of temporal work-case traces, have 31806, 5004, 1328, 2525, and 778 temporal work-case traces, respectively. The experimental results are depicted in Fig. 3 through four screens captured from the implemented algorithm and the SICN-oriented process mining system embedding the implemented algorithm as its preprocessing tool, as well. The captured screens at the right-hand side of the figure shows four SICN-oriented process models that are discovered from four selected groups (4, 3, 19, and 35 temporal work-case patterns) of temporal work-case patterns out of 59 different types of temporal work-case patterns after being clustered from the three fragmented process log datasets by the mapper and the reducer. While on the other hand, the left-hand side also shows the four captured screens displaying the statistical information, such as the number of traces, the number of trace-types (temporal work-case patterns), the number of events, and so on, discovered from the Customer Summary process **BIG-log** dataset.

Conclusively, we proposed a MapReduce-based inter-organizational process log clustering framework, which is one of the preprocessing techniques to be ultimately used in process mining technology to discover and rediscover process models from the distributed and fragmented execution event logs of large-scale inter-organizational process models. As a result, in order to verify the feasibility of the proposed clustering framework, we successfully implemented the process log clustering algorithm using the Hadoop-based MapReduce platform and applied to the actual process model execution event log dataset to verify the functional correctness of the proposed clustering framework as a preprocessing technique of the SICN-oriented process mining system.

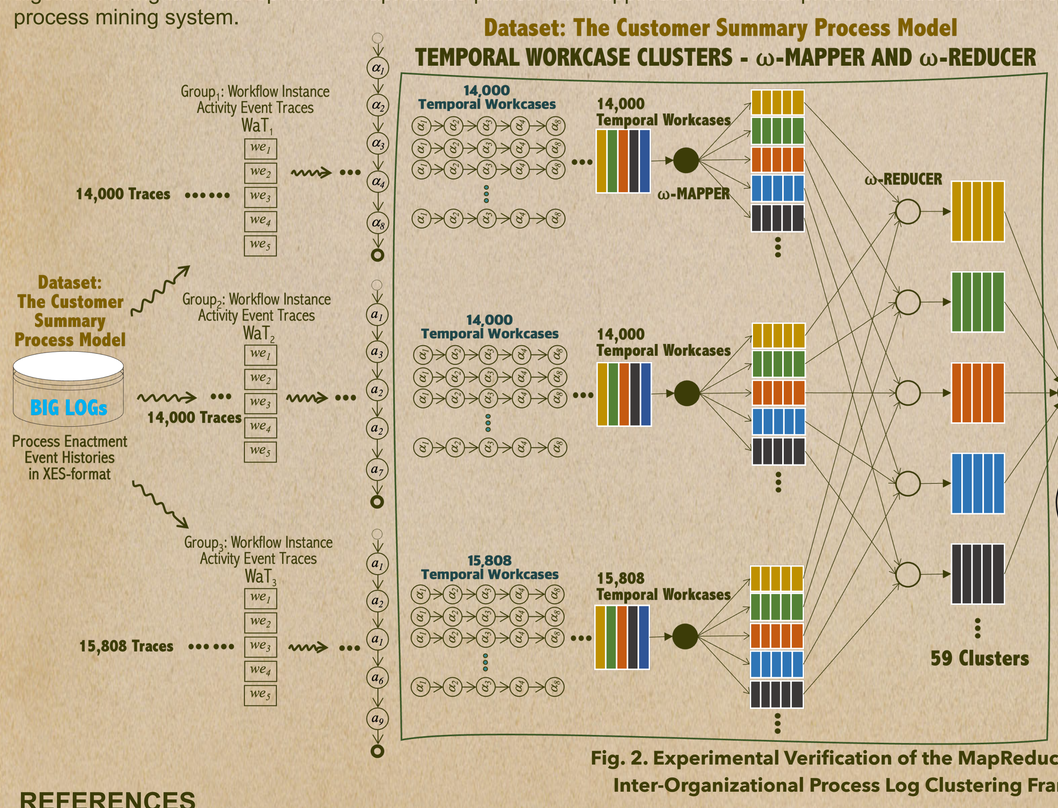

Fig. 2. Experimental Verification of the MapReduce-based Inter-Organizational Process Log Clustering Framework

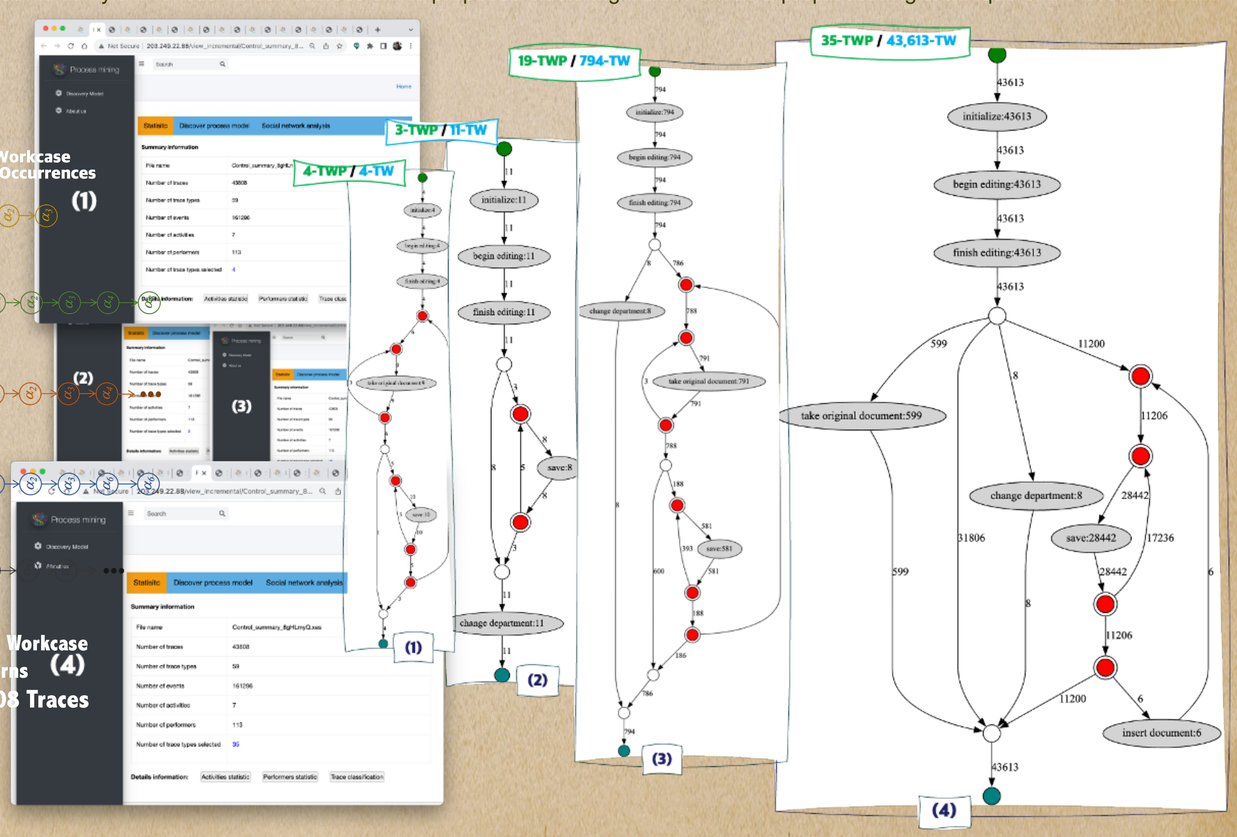

Fig. 3. Experimental Results of the SICN-oriented Process Mining System

REFERENCES

[1] Centre for Research Data 4TU. 2012, 2013, 2014, 2015, 2016, 2017, 2018. BPI Challenges. BPM.
[2] Kyoung-Sook Kim, Dinh-Lam Pham, and Kwanghoon Pio Kim. 2021. ρ-Algorithm: A SICN-Oriented Process Mining Framework. IEEE Access 9 (2021), 139852–139875.
[3] Kwanghoon Pio Kim. 2012. A Model-Driven Workflow Fragmentation Framework for Collaborative Workflow Architectures and Systems. Journal of Network and Computer Applications 35, 1 (2012), 97–110.
[4] K.-H. Lee, W. J. Park, K. S. Cho, and W. Ryu. 2013. The MapReduce framework for Large-scale Data Analysis: Overview and Research Trends. Electronics and telecommunications trends, 28, 6(2013),156–166.
[5] Kyoung-Sook Kim, et. al.. 2022. Experimental verification and validation of the SICN-oriented process mining algorithm and system. Journal of King Saud University - Computer and Information Sciences, 34,10, 9793–9813.

ACKNOWLEDGEMENTS
This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant No. NRF-2022R1A2C2093002).
The research outcomes in this paper belong to the Data and Process Engineering Research Lab. in Kyonggi University.

NRF 한국연구재단 Ministry of Education