Cluster Tool Performance Analysis using Graph Database

Shiuan-Hau Huang, Hsin-Ping Yen, Yan-Hsiu Liu, Kuang-Hsien Tseng, Ji-Fu Kung, Chia-Chun Lin, Yi-Ting Li, Yung-Chih Chen, and Chun-Yao Wang, *Member, IEEE*

2021 IEEE 34th International System-on-Chip Conference (SOCC) | 978-1-6654-2931-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/SOCC52499.2021.9739223

***Abstract*—A cluster tool is the place where wafers are processed. To increase the productivity of wafers, semiconductor manufacturers would like to maximize the utilization of cluster tools. However, a cluster tool may not achieve its optimal utilization rate due to the schedule conflict under different combinations of process recipes. By exploiting the data in the log files of a cluster tool, we may find out the reasons of low utilization rate. In this paper, we propose an approach of cluster tool performance analysis to automatically analyze the root cause of low utilization rate in cluster tools. Experimental results show the effectiveness of the proposed approach.**

I. Introduction

Single wafer processing technology is widely used in most semiconductor manufacturing processes including etching, deposition, photo-lithography, inspection, etc. This is a technique that processes wafers sequentially and has been increasingly used in cluster tools.

Fig.1 illustrates a cluster tool, which consists of the following components: Load Port, Load Lock, Robot Arm, Process Module (PM, also known as chamber). These components can be divided into internal and external regions by the Load Lock. The external region is the Load Port, and it is used to load and unload wafers in batch. The internal region has a Robot Arm and PMs, and is used to process wafers sequentially. The Load Lock is used to adjust air pressure and temperature for wafer movement between internal and external regions.

Due to limited space, the cluster tool has no buffer in the internal region. The cluster tool uses the Load Port in the external region as a buffer to store the wafers and transfers the wafers to be processed.

We briefly introduce how a cluster tool works. The Load Port contains a batch of wafers, usually 25 wafers. The wafers are transferred to Load Lock sequentially. The Load Lock opens door to catch the wafer. When the Load Lock closes the door, it pumps air pressure and transfers the wafer to the PM using the Robot Arm. The wafer is then processed at the designated PM based on its recipe. Once a wafer is unloaded from the Load Port to the Load Lock, it can only be returned to the Load Port after all the processing steps are completed.

Although the structure of a cluster tool is not complicated, its scheduling is complex. First, the cluster tool has a variety of different wafer flow modes, such as serial, parallels or multiple wafer flow modes. Second, PMs also have restrictions on the waiting time about wafer. This is because the remaining chemicals and high temperature in a PM both affect the quality of wafers waiting to be processed. Third, the cluster tool can be either single-arm or dual-arm cluster tools. Since dual-arm cluster tools can handle a large number of wafers, they usually have a higher complexity than single-arm cluster tools in scheduling. Last, because there is no buffer among PMs, deadlock is prone to occur in such an environment, making scheduling more complicated. For example, if a single-arm cluster tool transfers a wafer from the Load Lock to the PM, and the PM is processing another wafer simultaneously, it will cause a deadlock due to resource conflict. That is, the Robot Arm cannot send a wafer to the PM that is processing another wafer. The deadlock has a great impact on the entire scheduling such that it has to be avoided.

There have been numerous works on cluster tool scheduling, like using petri nets or reinforcement learning [6]- [13] to improve the productivity of single-arm or dual-arm cluster tools. Most of them use scheduling algorithms to achieve the maximum productivity for the identical wafer recipe. With the advances of semiconductor manufacturing technology, however, the wafer recipes and manufacturing process requirements are more diverse. Hence, there is a higher requirement for cluster tools.

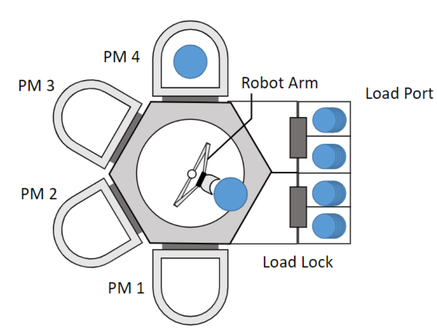

Fig. 1. Structure of a cluster tool.

This work is supported in part by the Ministry of Science and Technology of Taiwan under MOST 109-2221-E-007-082-MY2, MOST 109-2221-E-155-047-MY2, and MOST 109-2224-E-007-005, NTHU 110A0003EX.

S.-H. Huang, H.-P. Yen, C.-C. Lin, Y.-T. Li and C.-Y. Wang are with the Department of Computer Science, National Tsing Hua University, Hsinchu 300044, Taiwan R.O.C.

Y.-C. Chen is with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei 106335, Taiwan R.O.C.

Y.-H. Liu, K.-H. Tseng and J.-F. Kung are with the United Microelectronics Corporation, Hsinchu 30078, Taiwan R.O.C.

To achieve the maximal productivity, not only good algorithms are needed to schedule wafers, but the parameter setting of the cluster tool also matters. A cluster tool usually has a lot of parameter settings. For example, the transfer time of the Robot Arm, the just in time (JIT)[1], the opening time of air pressure gate, and the order of wafers to be processed. A suitable setting needs to be formulated for different wafer types and schedules. A cluster tool may not be able to complete the steps required by the scheduling algorithm due to improper or outdated settings, which affect the overall productivity. However, these settings are too complicated to be evaluated as the perfect setting in turns of throughput. In most cases, engineers observed the log files of the cluster tool to figure out improper settings and then modified the settings if necessary. However, it is quite challenging for engineers to read these detailed log files about every operation of the cluster tool. Relying on engineers to find out the improper parameter settings manually is not very effective.

In this paper, we propose an analysis system that can quickly find the segmented log files that represent productivity drop across different cluster tools and recipes. We also save the segmented log files and the relationship among log files in the database. The data saved in the database can be used to analyze the root causes of productivity drops.

We also compare the performance differences between using relational database and graph database for storing data. The experimental results show that the query performance is better when using a graph database as compared to using a relational database.

II. Preliminaries

In this section, we review the background of databases.

Relational database was proposed in the 1970s. It is a data structure with a collection of data item tables, and all data item tables are described and organized by the relational model between the tables. Since then, relational database has become a primary data structure for both academic and commercial pursuits.

In the relational database, each table schema has a key to identify a primary column used for identifying a row. The relationships between data are saved in another table, and relational database uses an external key to link the primary key for indicating which data of the primary key is recorded.

However, as the amount of data and the connections among data increase, relational database is not efficient in searching data. This is because the relational databases store the relationships among data in the form of table. Relational databases take much time to look up tables when they search data through relations. Therefore, when the times of searches increase, the performance will be degraded.

Recently, another storage structure, graph database, was proposed to eliminate the shortcomings of relational databases when storing data that is with complex relationships. In a graph database, information is represented by nodes, edges, and attributes. Nodes represent data, and edges represent relationships between nodes. Attributes can be added on both nodes and edges to express their specific characteristics. Unlike the relational database, the graph database uses edges to store the relationships between data such that the complexity of querying data can be reduced to O(1). There have been numerous works on comparing relational database and graph database [1]- [5]. The comparisons reveal that when data has the following characteristics, using a graph database to store data is more appropriate than a relational database.

1) The number of relationships among data is greater than the number of data.
2) Querying data with multi-level relationship is more frequently.
3) Having lots of many-to-many relationships.

III. Problem Formulation

To improve productivity, engineers have to figure out which wafer in the cluster tool has longer waiting time, which leads to inefficiency of the cluster tool. However, examining these data manually is very time-consuming since engineers usually convert the data into a Gantt chart for review based on their experience.

This paper aims to develop an analysis system to detect the productivity drop due to the abnormally long waiting time in the cluster tool. We use the log file obtained to represent the behavior of wafer in the cluster tool. Then, we store the raw data in the log file as nodes and relationships among nodes to analyze the root cause of productivity drop.

A. Log File

The log file records the wafer ID, recipe, physical location, and the duration for each wafer. TABLE I shows the format of a log file. The wafer transportation process starts from the Load Port and returns to the Load Port in the end. The difference for different wafers is that the order of PM given by the recipe is various.

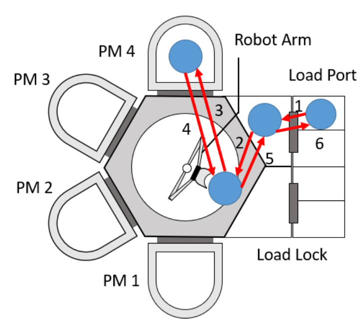

Fig. 2. A wafer path in a cluster tool.

Fig. 2 illustrates a path where a wafer usually moves in a cluster tool. The path of a wafer movement is as follows: Load Port → Load Lock → Robot Arm → PM → Robot Arm → Load Lock → Load Port.

[1]Grab the next wafer in advance so that when the current wafer is completed, the next wafer can be processed immediately.

TABLE I
LOG FILE FORMAT.

Data ID	Wafer ID	Recipe	Fromloc	Toloc	Fromtime	Totime	Duration (s)
1	WAFER 1	A	Robot Arm	PM 1	12:00:00	12:00:10	10
2	WAFER 1	A	PM 1	PM 1	12:00:10	12:01:00	50
3	WAFER 2	B	Robot Arm	PM 2	12:00:20	12:00:30	10
4	WAFER 2	B	PM 2	PM 2	12:00:30	12:01:10	40
5	WAFER 1	A	PM 1	Robot Arm	12:01:00	12:01:40	40
6	WAFER 2	B	PM 2	Robot Arm	12:01:10	12:01:20	10
7	WAFER 2	B	Robot Arm	Load Lock	12:01:20	12:01:30	10
8	WAFER 1	A	Robot Arm	Load Lock	12:01:40	12:01:50	10

The log file divides the wafer processing in the PM into two raw data. According to TABLE I, the raw data in the log file from one PM to the same PM represents the wafer processing within a PM. The raw data in the log file from PM to the Robot Arm represents the time that the wafer waits for Robot Arm to pick it up after processing.

Engineers usually convert these raw data into a Gantt chart, and examine the Gantt chart to find out the improper parameter settings of the cluster tool based on their experience. For example, Fig. 3 is a Gantt chart drawn from the data in TABLE I.

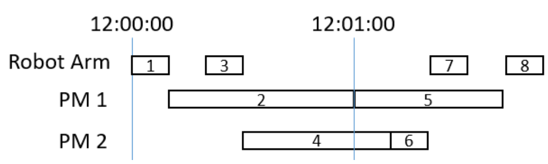

Fig. 3. The Gantt chart of data in TABLE I. The number on the bar is the data ID in TABLE I.

According to Fig. 3, we can find that the raw data of ID 5 spent more time from PM 1 to the Robot Arm. This is because the cluster tool was set incorrectly, which violates the first-in-first-out rule. As a result, the wafer 2 that was completed later was transferred to the Robot Arm earlier than the wafer 1. In this case, the Robot Arm sent out the wafer 2 prior to the wafer 1, and the wafer 1 needed to be idle in PM 1.

In the log file, each raw data has a value of duration, which represents the time period the wafer spent between the two locations (Fromloc and Toloc). Under the condition that a batch of 25 wafers, we have found that the distribution of duration is normal. Hence, when a raw data is with a significant increase in duration, we consider it as an outlier. We can use these outliers to figure out the improper parameter settings in the cluster tool.

For example, TABLE II shows the duration of each processing step from the Load Port to the PM and back to the Load Port for a batch of 25 wafers. The wafer was moved based on the recipe during operation, and the duration of each step under a recipe usually does not change a lot. For example, the duration of processing a wafer in a PM is almost the same as shown in Column 5 of TABLE II.

Fig. 4 illustrates the graph of distribution data about TABLE II normalized by the average. In Fig. 4, the data from PM to RA has more outliers. As we can see in TABLE II, wafer 18 spent almost six times of the minimum duration (10s) from PM to the Robot Arm. This is because this duration represents the time of the wafer in the PM that waits for Robot Arm to be available to pick it up, and it is unpredictable unfortunately. If the Robot Arm cannot be used due to a certain reason, this duration might exceed the normal case. The only Robot Arm in the cluster tool results in resource competition such that longer waiting time occurs more frequently.

TABLE II
DURATION OF RAW DATA IN THE LOG FILE.

Wafer ID	LP-LL	LL-RA	RA-PM	PM-PM	PM-RA	RA-LL	LL-LP
WAFER 1	5	285	8	507	11	8	52
WAFER 2	4	34	8	505	11	16	47
WAFER 3	5	242	9	507	10	9	51
WAFER 4	11	49	7	505	11	8	47
WAFER 5	5	260	9	507	26	9	51
WAFER 6	9	45	7	505	33	8	49
WAFER 7	5	237	8	507	11	9	51
WAFER 8	13	47	8	505	10	10	47
WAFER 9	15	276	9	507	10	9	51
WAFER 10	10	45	7	505	11	11	47
WAFER 11	5	261	8	507	10	9	72
WAFER 12	12	45	7	506	25	17	46
WAFER 13	4	277	9	507	10	9	51
WAFER 14	14	45	7	505	20	7	48
WAFER 15	5	232	8	507	11	8	51
WAFER 16	11	50	8	505	10	8	47
WAFER 17	4	261	9	507	17	9	82
WAFER 18	**17**	**47**	**8**	**505**	**59**	**8**	**47**
WAFER 19	5	288	8	507	11	8	51
WAFER 20	5	54	8	505	10	8	46
WAFER 21	5	261	9	507	10	9	51
WAFER 22	9	62	7	505	11	8	46
WAFER 23	5	241	8	507	11	8	47
WAFER 24	12	57	8	505	14	8	39
WAFER 25	5	261	8	508	10	10	37

LP: Load Port LL: Load Lock RA: Robot Arm PM: Process Module

Although we can use Gantt charts and outliers to find out the productivity drop due to improper parameter settings of the cluster tool, it is time-consuming to check the entire log file. Hence, we propose a new analysis system to analyze the data in the log file and the interactions among the data.

B. The Proposed Analysis System

We briefly introduce the proposed analysis system. First, we develop a storage model to save raw data in the log file as nodes and relationships. Then we analyze the reasons of productivity

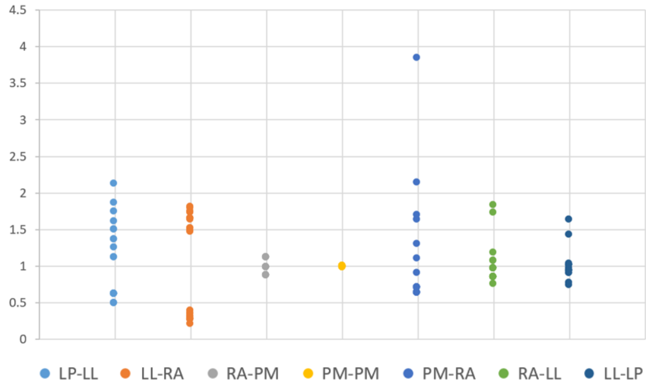

Fig. 4. A distribution graph of data in TABLE II normalized by the average.

drop, and examine whether other cluster tools encounter the productivity drop due to the same reason.

To analyze the relationship among raw data, we identify three relationships that need to be stored as follows: wafer relationship, location relationship, and overlap relationship.

The wafer relationship represents the relationship among the raw data of the same wafer at different locations. According to the wafer relationship, we can find out the moving path of one wafer in the cluster tool and the corresponding time stamps. The characteristic of the wafer relationship is that all the related raw data have the same wafer ID.

The location relationship represents the relationship of different wafers at the same physical location. According to the location relationship, we can know the sequence of wafers running at the same physical location in the cluster tool, and the time period of each physical location that does not have any wafers. The characteristic of the location relationship is that all the related raw data have the same *Fromloc*.

The overlap relationship represents the relationship among the raw data that are overlapped in the timeline. According to the overlap relationship, we can find out how the raw data affect each other and compete for resources. The characteristic of the overlap relationship is that all the related raw data have overlaps in the timeline.

The nodes involved in these three relationships all follow the order of time, from earlier raw data pointing to the later raw data. Using these three relationships, we can examine the interactions among the raw data, and figure out the reason of productivity drop. The operations of wafers in the cluster tool can be represented by a graph. Fig. 5 depicts a graph that presents the three relationships from the data in TABLE I, where the nodes are data ID and the directed edges indicate the precedence of data ID. We can identify sub-graphs that represent the violations of rules causing productivity drop, and then examine whether other cluster tools encounter the productivity drop due to the same reason.

As mentioned, the data ID 5 in TABLE I spent much more time due to the first-in-first-out rule violation in the cluster tool. Hence, in the proposed analysis system, we first focus on the sub-graph representing the first-in-first-out rule violation in the cluster tool.

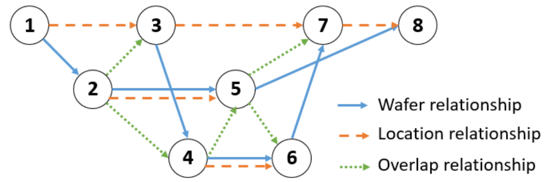

Fig. 5. Graph representation for data in TABLE I.

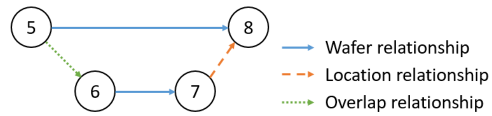

Fig. 6. The sub-graph representing the first-in-first-out rule violation.

The first-in-first-out rule violation is caused by the situation that a lately completed wafer is sent out earlier than an early completed one. Hence, the early completed wafer has to wait in the PM. The sub-graph with respect to the first-in-first-out rule violation is shown in Fig. 6.

In Fig. 6, the overlap relationship between node 5 and node 6 represents that the starting time of node 5 is earlier than that of node 6. However, node 7 points to node 8 through the location relationship, which means that Robot Arm transfers the wafer in node 7 first, then the wafer in node 8. This scenario violates the first-in-first-out rule because wafer 1 completes its operation earlier than wafer 2.

TABLE III shows another set of raw data in a log file having productivity drop due to resource competition. We find that the raw data of ID 1 spent more time from PM 1 to the Robot Arm. Fig. 7 illustrates a sub-graph about the data in TABLE III.

Fig. 7. The sub-graph representing the resource competition.

In Fig. 7, node 2 represents the Robot Arm transfers wafer 2 to the Load Lock. Therefore, the location relationship from node 2 to node 3 and the wafer relationship from node 1 to node 3 represent that when wafer 1 has completed the process in PM 1 and waits for Robot Arm to unload it, the Robot Arm transfers wafer 2 instead of wafer 1 to Load Lock. Hence wafer 1 has to wait until Robot Arm is available. The sub-graph in Fig. 7 represents the resource competition that multiple wafers in the PMs wait for the only Robot Arm in the cluster tool to unload them simultaneously.

TABLE IV shows another set of raw data in a log file having productivity drop due to PM waits for Load Lock pumping.

TABLE III
RAW DATA OF PRODUCTIVITY DROP FOR RESOURCE COMPETITION

Data ID	Wafer ID	Recipe	Fromloc	Toloc	Fromtime	Totime	Duration(s)
1	WAFER 1	A	PM 1	Robot Arm	12:00:00	12:00:20	20
2	WAFER 2	B	PM 2	Robot Arm	12:00:05	12:00:15	10
3	WAFER 1	A	Robot Arm	Load Lock	12:00:20	12:00:30	10

TABLE IV
RAW DATA OF PRODUCTIVITY DROP FOR PM WAITS FOR LOAD LOCK PUMP.

Data ID	Wafer ID	Recipe	Fromloc	Toloc	Fromtime	Totime	Duration(s)
1	WAFER 1	A	PM 1	PM 1	12:00:00	12:00:30	30
2	WAFER 1	A	PM 1	Robot Arm	12:00:30	12:01:10	40
3	WAFER 2	A	Load Lock	Robot Arm	12:00:40	12:01:10	40
4	WAFER 1	A	Robot Arm	Load Lock	12:01:10	12:01:20	10
5	WAFER 2	A	Robot Arm	PM 1	12:01:20	12:01:30	10

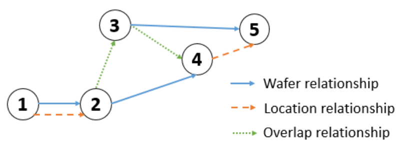

Fig. 8. Graph representation for data in TABLE IV.

The graph representation of TABLE IV is shown in Fig. 8. According to TABLE IV, we find that the raw data of ID 2 spent more time from PM 1 to the Robot Arm.

In Fig. 8, node 3 represents that the Load Lock is pumping such that other wafers cannot pass through it before the end of the pumping operation. Therefore, the overlap relationships from node 2 to node 3, and node 3 to node 4 represent that node 2 has to wait at PM 1 for the Load Lock to complete the pumping operation.

The mentioned examples demonstrate that the proposed analysis system identifies the sub-graphs with respect to different outliers for detecting the productivity drop.

The flow chart of the proposed analysis system is shown in Fig. 9. First, it stores the raw data in a log file of a cluster tool as nodes and constructs the graph to represent the relationships among the raw data. Next, it compares the duration of each step among the wafers under the same recipe to detect outliers. The sub-graphs with respect to outliers are classified by different reasons of productivity drop. If the sub-graph of an outlier cannot be classified into a category of known reason, we create a new category for it. When all the outliers are classified, the proposed analysis system reports the results about the root causes of productivity drop in the cluster tool.

IV. EXPERIMENTAL RESULTS

We implemented the proposed analysis system with a relational database and a graph database for comparison. The chosen relational database is MySQL Workbench version 8.0.22. For the graph database, it is Neo4j desktop version 1.4.1. The experiments were conducted on an Intel Xeon E5-2650v2 2.60 GHz CentOS 6.7 platform with 64GBytes memory.

To demonstrate that the proposed system can analyze the reason of productivity drop across different cluster tools and recipes, and show that the graph database is more suitable for the proposed analysis system than the relational database, we use nine test cases covering different log files for the experiments. The log files involve three different cluster tools for one hour, four hours, and eight hours operation.

TABLE V shows the experimental results. The columns represent the number of cluster tools, the length of log file, the number of batches, the number of involved wafers, the number of nodes and the number of relationships in the graph, the numbers of outliers and the outlier categories, and the total query time for different databases Neo4j and MySQL.

The number of overlap relationships is the largest among the three relationships. This is because a raw data ID usually has only one wafer relationship and one location relationship pointing to other raw data, but it may have many overlap relationships pointing to other raw data.

According to TABLE V, as the number of cluster tools and the length of log file increases, the number of outlier categories does not increase a lot. This is because the causes of outliers in the cluster tools are mostly the same. Fig. 10 shows the patterns of sub-graphs about these outliers. The reasons of productivity drop are various, and correspond to different sub-graphs. For example, the resource competition occurs when two PMs complete their wafers at the same time, or a PM completes its wafer while the Robot Arm is busy. The outlier node in the sub-graph of improper parameter setting has no overlap relationship, which means that no other wafers affect the outlier. Hence, it was caused by improper parameter settings in the cluster tool. The query time increases as the raw data increases. As the number of raw data increases, the number of outliers also increases. The query time of proposed system using MySQL is longer than that of Neo4j. It is because most operations in the proposed system need to be queried through

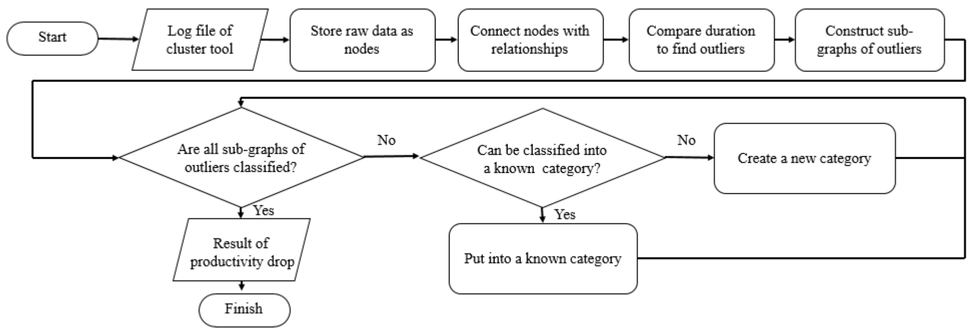

Fig. 9. Flow chart of the proposed analysis system.

TABLE V
EXPERIMENTAL RESULTS.

|Cluster Tool|	Process time (h)	|Batch|	|Wafer|	|Node|	|Wafer Rela.|	|Location Rela.|	|Overlap Rela.|	|Outlier|	|Category|	Query time (ms)	
										Neo4j	MySQL
1	1	3	21	97	76	90	281	4	2	154	230
1	4	7	116	568	452	561	2170	28	6	431	594
1	8	8	174	864	690	857	2977	36	6	796	853
2	1	6	57	251	194	237	865	7	3	347	497
2	4	14	229	1167	938	1153	4204	41	6	562	613
2	8	16	345	1757	1414	1745	5818	58	6	1098	1273
3	1	7	82	376	294	358	1049	11	3	464	572
3	4	19	305	1577	1248	1536	5344	65	6	1350	1934
3	8	25	527	2683	2152	2662	8937	89	6	2031	2953

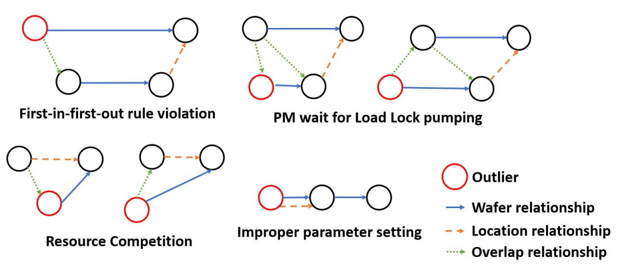

Fig. 10. Summary of sub-graph patterns representing outliers.

relationships when constructing the sub-graphs of outliers or classifying them. As compared with the graph database, the relational database requires more time to find the data through the relationship, which suggests the graph database is preferable in our system.

V. CONCLUSION

In this paper, we propose a system for the cluster tool to detect the productivity drop and analyze the corresponding root causes. The experimental results show that the proposed system works well across different cluster tools and recipes under the graph database. Having such a system, engineers can further elevate the productivity of cluster tool by re-scheduling.

REFERENCES

[1] R. Angles, "A comparison of current graph database models", *IEEE International Conf. on Data Engineering Workshops*, pp. 171–177, April 2012.

[2] R. Angles and C. Gutierrez, "Survey of graph database models", *ACM Computing Surveys*, pp.123–147, 2008.

[3] R. Cattell, "Scalable sql and nosql data stores", *SIGMOD Record*, Vol.39, No.4, pp. 12-27, May 2011.

[4] M. Ciglan, A.Averbuch and L. Hluchy, "Benchmarking traversal operations over graph databases", *International Workshop on Graph Data Management: Techniques and Applications*, pp. 186–189, April 2012.

[5] D. Dominguez-Sal, P. Urbòn-Bayes, A. Gimenèz-Vañò, S. Gomez-Villamor, N. Martìnez-Bazan, and J. Larribà-Pey, "Survey of graph database performance on the hpc scalable graph analysis benchmark", in *Web-Age Information Management*,Vol. 6185, pp. 37–48, 2010.

[6] C. Hong and T.-E. Lee, "Multi-Agent reinforcement learning approach for scheduling cluster tools with condition based chamber cleaning operations", *IEEE International Conference on Machine Learning and Applications*, pp. 885-890, 2018.

[7] C. Jung and T.-E. Lee, "An efficient mixed integer programming model based on timed petri nets for diverse complex cluster tool scheduling problems", *IEEE Trans. Semicond. Manuf.*, VOL. 25, NO. 2, pp. 186-199, May 2012.

[8] C. Jung, H.-J. Kim, and T.-E. Lee, "A branch and bound algorithm for cyclic scheduling of timed Petri net", *IEEE Trans. Autom. Sci. Eng.*, pp. 309–323, January 2015.

[9] J.-H. Lee and H.-J. Kim, "Backward sequence analysis for single-armed cluster tools", in *Proc. 29th Int. Conf. Automated Planning Scheduling*, pp. 269–272, July 2019.

[10] T.-E. Lee, H.-Y. Lee, and Y.-H. Shin, "Workload balancing and scheduling of a single-armed cluster tools", in *Proc. 5th APIEMS Conf.*, Gold Coast, QLD, Australia, pp. 1–15, December 2004.

[11] Y. Qiao, N. Wu, and M.-C. Zhou, "Scheduling of dual-arm cluster tools with wafer revisiting and residency time constraints", *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 286–300, February 2014.

[12] Y. Qiao, N. Wu, and M.-C. Zhou, "Schedulability and scheduling analysis of dual-arm cluster tools with wafer revisiting and residency time constraints based on a novel schedule", *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 3, pp. 472–484, March 2015.

[13] S. C. Wood and K. C. Saraswat, "Modeling the performance of cluster-based fabs", in *Proc. IEEE International Semiconductor Manufacturing Science Symposium*, pp. 8–14, May 1991.