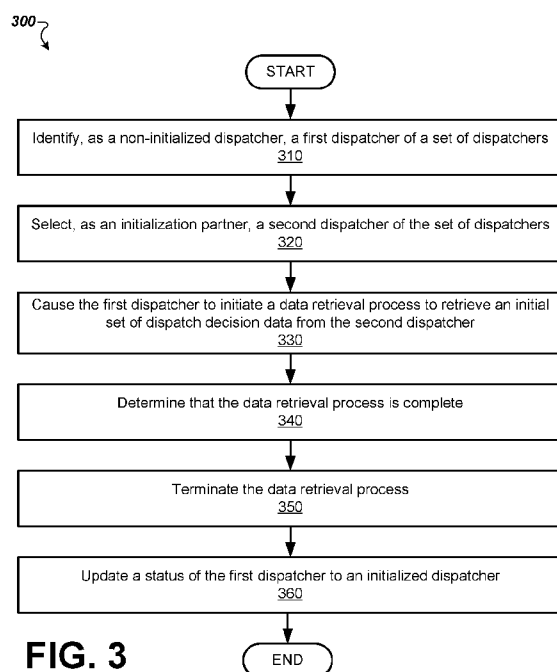




- (51) **International Patent Classification:**
G05B 19/418 (2006.01) *G05B 23/02* (2006.01)
- (21) **International Application Number:**
PCT/US2024/036211
- (22) **International Filing Date:**
28 June 2024 (28.06.2024)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
18/222,089 14 July 2023 (14.07.2023) US
- (71) **Applicant:** **APPLIED MATERIALS, INC.** [US/US];
3050 Bowers Avenue, Santa Clara, California 95054 (US).
- (72) **Inventors:** **HOAD, Paul D.**; 3050 Bowers Avenue, Santa Clara, California 95054 (US). **CHINNUSAMY, Shankar**; 3050 Bowers Avenue, Santa Clara, California 95054 (US). **PALANIAPPAN, Prasath**; 3050 Bowers Avenue, Santa Clara, California 95054 (US). **MOHAN, Jayapradeep**; 3050 Bowers Avenue, Santa Clara, California 95054 (US).
- (74) **Agent:** **PORTNOVA, Marina et al.**; Lowenstein Sandler LLP, One Lowenstein Drive, Roseland, New Jersey 07068 (US).

- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.
- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) **Title:** REDUCING IMPACT OF DISPATCHER INITIALIZATION ON DISPATCHING SYSTEM PERFORMANCE



(57) **Abstract:** A method of operating a set of dispatchers for a manufacturing facility, wherein each dispatcher of the set of dispatchers is a software application that makes dispatch decisions for the manufacturing facility based on facility data related to the manufacturing facility, includes identifying a first dispatcher of the set of dispatchers, wherein the first dispatcher is a non-initialized dispatcher, selecting, as an initialization partner, a second dispatcher of the set of dispatchers, and causing the first dispatcher to initiate a data retrieval process with the second dispatcher to retrieve, from the second dispatcher, an initial set of facility data related to the manufacturing facility.

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

- *with international search report (Art. 21(3))*

REDUCING IMPACT OF DISPATCHER INITIALIZATION ON DISPATCHING SYSTEM PERFORMANCE

TECHNICAL FIELD

[001] The present disclosure relates to semiconductor device fabrication, and, more particularly, to reducing impact of dispatcher initialization on dispatching system performance.

BACKGROUND

[002] Products can be produced by performing one or more manufacturing processes using manufacturing equipment. For example, semiconductor manufacturing equipment can be used to produce semiconductor devices (e.g., substrates) via semiconductor manufacturing processes. The manufacturing equipment can, according to a process recipe and via a substrate processing tool, deposit multiple layers of film on the surface of the substrate and can perform an etch process to form the intricate pattern in the deposited film. Sensors can be used to determine manufacturing parameters of the manufacturing equipment during the manufacturing processes and metrology equipment can be used to determine property data of the products that were produced by the manufacturing equipment, such as the overall thickness of the layers on the substrate.

SUMMARY

[003] The following is a simplified summary of the disclosure in order to provide a basic understanding of some aspects of the disclosure. This summary is not an extensive overview of the disclosure. It is intended to neither identify key or critical elements of the disclosure, nor delineate any scope of the particular implementations of the disclosure or any scope of the claims. Its sole purpose is to present some concepts of the disclosure in a simplified form as a prelude to the more detailed description that is presented later.

[004] In an aspect of the disclosure, a method of operating a set of dispatchers for a manufacturing facility is provided. Each dispatcher of the set of dispatchers is a software application that makes dispatch decisions for the manufacturing facility based on facility data related to the manufacturing facility. The method includes identifying a first dispatcher of the set of dispatchers, wherein the first dispatcher is a non-initialized dispatcher, selecting, as an initialization partner, a second dispatcher of the set of dispatchers, and causing the first

dispatcher to initiate a data retrieval process with the second dispatcher to retrieve, from the second dispatcher, an initial set of facility data related to the manufacturing facility.

[005] In another aspect of the disclosure, a system for operating a set of dispatchers for a manufacturing facility is provided. Each dispatcher of the set of dispatchers is a software application that makes dispatch decisions for the manufacturing facility based on facility data related to the manufacturing facility. The system includes a memory and a processing device, operatively coupled to the memory, to perform operations including identifying a first dispatcher of the set of dispatchers, wherein the first dispatcher is a non-initialized dispatcher, selecting, as an initialization partner, a second dispatcher of the set of dispatchers, and causing the first dispatcher to initiate a data retrieval process with the second dispatcher to retrieve, from the second dispatcher, an initial set of facility data related to the manufacturing facility.

[006] A further aspect of the disclosure includes a non-transitory computer-readable storage medium is provided. The non-transitory computer-readable storage medium includes instructions that, when executed by a processing device operatively coupled to a memory, performs operations for operating a set of dispatchers for a manufacturing facility. Each dispatcher of the set of dispatchers is a software application that makes dispatch decisions for the manufacturing facility based on facility data related to the manufacturing facility. The operations include identifying a first dispatcher of the set of dispatchers, wherein the first dispatcher is a non-initialized dispatcher, selecting, as an initialization partner, a second dispatcher of the set of dispatchers, and causing the first dispatcher to initiate a data retrieval process with the second dispatcher to retrieve, from the second dispatcher, an initial set of facility data related to the manufacturing facility.

BRIEF DESCRIPTION OF THE DRAWINGS

[007] The present disclosure is illustrated by way of example, and not by way of limitation in the figures of the accompanying drawings.

[008] **FIG. 1** is a block diagram of an example computing system, in accordance with some implementations of the present disclosure.

[009] **FIG. 2** is a diagram of an example system that can initialize dispatchers of a dispatching system of a manufacturing facility, in accordance with some implementations of the present disclosure.

[0010] FIGS. 3-5 are flowcharts of example methods for initializing dispatchers of a dispatching system of a manufacturing facility, according to aspects of the present disclosure.

[0011] FIG. 6 is a block diagram illustrating a computer system, according to certain implementations.

DETAILED DESCRIPTION

[0012] Described herein are technologies directed to reducing impact of dispatcher initialization on dispatching system performance. A dispatching system of a manufacturing facility can be used to make dispatch decisions that control operation of processing tools of the manufacturing facility. In some implementations, a manufacturing facility include an electronic device manufacturing system. An electronic device manufacturing system can include an electronic device processing system including multiple substrate processing tools, such as processing chambers. A processing chamber can have multiple sub-systems operating during each substrate manufacturing process (e.g., the deposition process, the etch process, the polishing process, etc.). A sub-system can be characterized as a set of sensors and controls related with an operational parameter of the processing chamber. An operational parameter can be a temperature, a flow rate, a pressure, and so forth. In an example, a pressure sub-system can be characterized by one or more sensors measuring the gas flow, the chamber pressure, the control valve angle, the foreline (vacuum line between pumps) pressure, the pump speed, and so forth. Accordingly, the processing chamber can include a pressure sub-system, a flow sub-system, a temperature subsystem, and so forth. A processing chamber can perform a manufacturing process according to a process recipe. A process recipe defines a particular set of operations to be performed for the substrate during the process and can include one or more settings associated with each operation. A process recipe can be embodied as a table of recipe settings including a set of inputs or recipe parameters (“parameters”) and processes that are manually entered by a user (e.g., process engineer) to achieve a set of target properties (e.g., on-substrate characteristics), also referred to as a set of goals. For example, a deposition process recipe can include a temperature setting for the processing chamber, a pressure setting for the processing chamber, a flow rate setting for a precursor for a material included in the film deposited on the substrate surface, etc. Accordingly, the thickness of each film layer, the depth of each etch, and so forth, can be correlated to these processing chamber settings.

[0013] A dispatching system can make dispatch decisions to improve manufacturing

productivity. For example, a dispatching system for an electronic device manufacturing system can be used to make dispatch decisions to improve manufacturing productivity across multiple substrate processing tools of the electronic device processing system. In some implementations, the dispatching system is a real-time dispatching (RTD) system that makes dispatch decisions in real-time or near real-time. Dispatching systems can enable electronic device manufacturers to develop dispatching policies to optimally fabricate various electronic devices with minimal performance bottleneck across the substrate processing tools.

[0014] For example, a dispatching system can include a manufacturing execution system (MES) communicably coupled to a set of substrate processing tools. The MES can gather raw facility data from various components of the manufacturing facility, such as a set of processing tools. The dispatching system can further include an MES database that can store raw facility data. In some implementations, the MES database is a relational database.

[0015] The dispatching system can further include a data processing component to process the raw facility data to generate processed facility data, also referred to as facility data. The dispatching system can further include a repository that can store the facility data. In some implementations, the repository is a temporal repository.

[0016] The facility data can be used by the dispatching system to coordinate and optimize electronic device processing tasks to meet production goals. For example, the facility data can be used to track individuals lots and substrates (e.g., wafers) throughout electronic device fabrication, manage process recipes used by substrate processing tools to fabricate electronic devices, monitor the status of the substrate processing tools, perform yield management to improve overall yield of the electronic device processing system, etc.

[0017] More specifically, the dispatching system can further include a set of dispatchers. A dispatcher is a software application that manages the scheduling and execution of tasks performed by processing tools in a manufacturing facility. For example, a task can be a substrate process performed by a substrate processing tool of an electronic device manufacturing system. In a manufacturing facility, there can be multiple processing tools and multiple lots that may need to be processed at the same time. Thus, the set of dispatchers can include multiple dispatchers in order to concurrently handle multiple dispatching requests received at approximately the same time.

[0018] For example, a dispatcher can optimize resource utilization, prioritize tasks, determine tasks execution order to maximize throughput, distribute workload across processing tools to

optimize efficiency (e.g., load balancing), monitor a state of the electronic device processing system, etc. In particular, a dispatcher can make dispatch decisions regarding task scheduling and execution to optimize task execution (e.g., improve throughput). A dispatch decision defines an action that should happen next in the manufacturing facility. For example, a dispatch decision can select a processing tool into which a substrate should be placed for processing. Examples of dispatch decisions that can be performed in an electronic device processing system can include “where a wafer lot should be processed next,” “which wafer lot should be picked for an idle substrate processing tool,” etc. The state of an electronic device processing system can include status of the substrate processing tools, status of the electronic devices (e.g., locations and/or processing states of substrates), status of processing tasks being performed by the substrate processing tools, etc.

[0019] To make dispatch decisions, a dispatcher can utilize an in-memory rule execution engine that processes a set of dispatch rules based on dispatch decision data. For example, dispatch decision data can include facility data related to the manufacturing facility, such as an electronic device manufacturing system. Dispatch decision data can include data reflecting a state of the electronic device manufacturing system and/or factors that can affect dispatching (e.g., task scheduling and execution among the substrate processing tools). Examples of dispatch decision data can include, for example, lot information, substrate processing tool information (e.g., substrate processing tool capability and/or availability), route information, process recipe information, production goals, etc. Examples of dispatch rules can include, and are not limited to, select the highest priority wafer lot to work on next, select a wafer lot that uses the same set up which the tool is currently configured for, package items when a purchase order is complete, ship items when packaging is complete, etc. For example, the dispatch decision to be made may be “where a wafer lot should be processed next,” and the dispatch rule that may be used to make the decision may be to “select the highest priority wafer lot to work on next and select a wafer lot that uses the same set up which the substrate processing tool is currently configured for.”

[0020] It may be the case that a dispatcher has shut down and needs to be restarted. When a dispatcher shuts down, the dispatcher can lose the dispatch decision data stored in-memory. Thus, the dispatcher may need to be initialized (“primed”) as part of the restart process to enable the dispatcher to make dispatch decisions. Initializing the dispatcher can include retrieving an initial set of dispatch decision data that is stored in-memory. For example, the initial set of dispatch decision data can include an initial set of facility data. The initial set of

dispatch decision data can define a state of the manufacturing facility and/or factors. A similar process can be performed when a new dispatcher is being added to the dispatching system.

[0021] In some dispatching systems, a dispatcher is initialized by retrieving the initial set of dispatch decision data directly from the repository. However, retrieving the initial set of dispatch decision data from the repository can place a load on the repository. The load placed on the repository can cause performance bottlenecks and/or can prevent updates to facility data stored in the repository (e.g., real-time or near real-time updates to data). This, in turn, can prevent other dispatchers of the dispatching system from being able to make dispatch decisions in real-time or near real-time. Accordingly, initializing a dispatcher by retrieving an initial set of dispatch decision data from a repository can negatively impact the ability of a dispatching system to make accurate dispatch decisions within an electronic device manufacturing system.

[0022] Aspects and implementations of the present disclosure address these and other shortcomings of the existing technology by reducing impact of dispatcher initialization on dispatching system performance. A dispatching system described herein can include a dispatch server that can manage dispatcher initialization. In particular, the dispatch server can enable a non-initialized dispatcher to be initialized by retrieving dispatch decision data from an initialized dispatcher, referred to as an initialization partner. For example, a non-initialized dispatcher can be a new dispatcher or a dispatcher that has been restarted. By initializing or priming a non-initialized dispatcher from an initialization partner, instead of from the repository, implementations described herein can reduce load placed on the repository without needing to isolate the dispatch system and/or interfere with repository data updates. Further details regarding the dispatching system will be described below with reference to **FIGS. 1-6**.

[0023] Aspects of the present disclosure result in technological advantages. For example, implementations described herein can reduce overhead on the repository of a dispatching system, which can reduce the impact of dispatcher initialization on dispatching system performance and improve computational resource consumption.

[0024] **FIG. 1** depicts an illustrative computer system (“system”) 100, according to aspects of the present disclosure. In some implementations, system 100 can be included as part of an electronic device manufacturing system for manufacturing electronic devices (e.g.,

processing substrates). System 100 includes a client device 110, substrate processing tools 122-1 through 122-*N*, set of sensors 124, dispatch server controller 126, metrology equipment 130, data store 140, dispatching system 150 and predictive system 160. Client device 110, processing system 210, metrology equipment 130, data store 140, dispatching system 150 and predictive system 160 can be communicably coupled to each other via network 170. In some implementations, network 170 is a public network. In some implementations, network 170 is a private network. Network 170 can include one or more wide area networks (WANs), local area networks (LANs), wired networks (e.g., Ethernet network), wireless networks (e.g., an 802.11 network or a Wi-Fi network), cellular networks (e.g., a Long-Term Evolution (LTE) network), routers, hubs, switches, server computers, cloud computing networks, and/or a combination thereof.

[0025] Processing tools 122-1 through 122-*N* can be used to fabricate or manufacture electronic devices. Each of processing tools 122-1 through 122-*N* can perform at least one substrate process for processing a substrate (e.g., a wafer) using a substrate processing tool. Examples of substrate processes include a deposition process to deposit one or more layers of film on a surface of the substrate, an etch process to form a pattern on the surface of the substrate, etc.

[0026] Each of processing tools 122-1 through 122-*N* can perform a substrate process according to a process recipe. A process recipe defines a particular set of operations to be performed for a substrate during the process and can include one or more settings associated with each operation. For example, a deposition process recipe can include a temperature setting for the processing chamber, a pressure setting for the processing chamber, a flow rate setting for a precursor for a material included in the film deposited on the substrate surface, etc. In some implementations, set of substrate processing tools 122 includes one or more processing chambers. For example, the one or more processing chambers can include at least one of: one or more deposition chambers, one or more etch chambers, one or more lithography chambers, etc.

[0027] Set of sensors 124 can be configured to obtain data associated with a substrate being processed by set of substrate processing tools 122. In some implementations, set of sensors 124 can be part of a sensor system that includes sensor server 140 and a sensor identifier reader (e.g., front opening unified pod (FOUP) radio frequency identification (RFID) reader). For example, a substrate processing tool can include one or more sensors configured to generate spectral or non-spectral data associated with a substrate before, during, and/or after a

substrate process is performed for the substrate. In some implementations, spectral data generated by set of sensors 124 can indicate a concentration of one or more materials deposited on a surface of a substrate. Sensors that can generate spectral data can include reflectometry sensors, ellipsometry sensors, thermal spectra sensors, capacitive sensors, and so forth. Sensors that can generate non-spectral data can include temperature sensors, pressure sensors, flow rate sensors, voltage sensors, etc. For example, a sensor can be a temperature sensor, a pressure sensor, a chemical detection sensor, a chemical composition sensor, a gas flow sensor, a motion sensor, a position sensor, an optical sensor, or any and other type of sensors. At least one sensor of set of sensors 124 can include a light source to produce light (or any other electromagnetic radiation), direct it towards a target, such as a component of system 100 or a substrate, a film deposited on the substrate, etc., and detect light reflected from the target. A sensor of set of sensors 124 can be located inside of an electronic device processing system (e.g., within any of the chambers including the loading stations, on one or more robots, on a robot blade, between the chambers, and so one) and/or outside of an electronic device processing system (e.g., to test ambient temperature, pressure, gas concentration, etc.). Sensor data received over a period of time (e.g., corresponding to at least part of a recipe or run) can be referred to as trace data (e.g., historical trace data, current trace data, etc.) received from different sensors over time. Sensor data can include a value of one or more of temperature (e.g., heater temperature), spacing (SP), pressure, high frequency radio frequency (HFRF), voltage of electrostatic chuck (ESC), electrical current, material flow, power, voltage, etc. Sensor data can be associated with or indicative of manufacturing parameters such as hardware parameters, such as settings or components (e.g., size, type, etc.) the electronic device processing system, or process parameters of the electronic device processing system. The sensor data can be provided while at least one of processing tools 122-1 through 122-N is performing substrate processing (e.g., equipment readings when processing products).

[0028] In some implementations, the electronic device processing system can include a set of sub-systems to enable and/or control one or more substrate processing. For example, the set of sub-systems can include at least one of: a pressure sub-system, a flow sub-system, a temperature sub-system, etc. Each sub-system can include a set of components. A set of components can include at least one of: a pressure pump, a vacuum, a gas deliver line, a plasma etcher, actuators etc. In some implementations, operation of the set of sub-systems can be controlled based at least in part on sensor data.

[0029] In some implementations, metrology equipment 130 includes a metrology server (e.g., a metrology database, metrology folders, etc.) and a metrology identifier reader (e.g., FOUNTRONICS reader for metrology system). Metrology equipment 130 can generate metrology data associated with substrates that are processed by at least one processing tools 122-1 through 122-*N*. The metrology data can include a value of film property data (e.g., wafer spatial film properties), dimensions (e.g., thickness, height, etc.), dielectric constant, dopant concentration, density, defects, etc. In some implementations, the metrology data can further include a value of one or more surface profile property data (e.g., an etch rate, an etch rate uniformity, a critical dimension of one or more features included on a surface of the substrate, a critical dimension uniformity across the surface of the substrate, an edge placement error, etc.). The metrology data can be of a finished or semi-finished product. The metrology data can be different for each substrate. Metrology data can be generated using, for example, reflectometry techniques, ellipsometry techniques, transmission electron microscope (TEM) techniques, etc. In some implementations, metrology equipment 130 is included in the electronic device processing system. For example, metrology equipment 130 can be operatively coupled to at least one processing tools 122-1 through 122-*N* to generate metrology data for a substrate before, during, and/or after a substrate process performed while the substrate remains in the processing chamber. In some instances, metrology equipment 130 can be referred to as in-situ metrology equipment. In another example, metrology equipment 130 can be coupled to another station of manufacturing equipment 124. For example, metrology equipment can be coupled to a transfer chamber, a load lock, a factory interface, etc.

[0030] Client device 110 can include a computing device such as personal computers (PCs), laptops, mobile phones, smart phones, tablet computers, netbook computers, network connected televisions (“smart TVs”), network-connected media players (e.g., Blu-ray player), a set-top box, over-the-top (OTT) streaming devices, operator boxes, etc. In some implementations, the metrology data can be received from the client device 110. Client device 110 can display a graphical user interface (GUI), where the GUI enables the user to provide, as input, metrology measurement values for processed substrates. Client device 110 can include user interface (UI) 112, application 114, and corrective action component 116. In implementations, a “user” can be represented as a single individual. However, other implementations of the disclosure encompass a “user” being an entity controlled by a

plurality of users and/or an automated source. For example, a set of individual users federated as a group of administrators can be considered a “user.”

[0031] Application 114 can be a computer program configured to provide maintenance, services, analytics, and predictive technologies performed by one or more evaluation systems (e.g., machine-learning models, inference engines, heuristics models, algorithms, physics-based engine, etc.). One or more evaluation systems (e.g., a machine-learning model) can be generated by predictive system 160. UI 112 can receive user input (e.g., via a Graphical User Interface (GUI) displayed via client device 110) associated with application 114. In some implementations, user interface 112 can be presented via a web browser (not shown) and application 114 can be hosted on an application server (not shown). Alternatively, the client device 110 includes a local (mobile or desktop) application 114 that provides user interface 112. In some implementations, user interface 112 can communicate with the application 114 via network 130.

[0032] In some implementations, input data can be sent to or processed by application 114. Corrective action component 116 can be part of application 114 or a separate system (e.g., program, application, etc.). In some implementations, corrective action component 116 receives input data from at least one component of system 100, determines a corrective action based on the input data, and causes the corrective action to be performed. Corrective action component 116 can receive user input (e.g., via a Graphical User Interface (GUI) displayed via the client device 110) of an indication associated with substrate processing. For example, responsive to receiving an indication that sensor data satisfied a threshold criterion (e.g., exceeded or fell below a fault detection limit), correction action module 116 can perform one or more corrective action (e.g., increase power, decrease flowrate, etc.). The corrective actions can be stored in a fault pattern library in data store 140.

[0033] In some implementations, corrective action component 116 transmits the indication to predictive system 160 (or any other service provided by application 114), receives output (e.g., predictive data) from predictive system 160, determines a corrective action based on the output, and causes the corrective action to be implemented. In some implementations, corrective action component 116 receives an indication of a corrective action from predictive system 160 and causes the corrective action to be implemented. Each client device 110 can include an operating system that allows users to one or more of generate, view, or edit data (e.g., indication associated with manufacturing equipment 124, corrective actions associated with manufacturing equipment 124, etc.).

[0034] Data store 140 can be a memory (e.g., random access memory), a drive (e.g., a hard drive, a flash drive), a database system, or another type of component or device capable of storing data. Data store 140 can include multiple storage components (e.g., multiple drives or multiple databases) that can span multiple computing devices (e.g., multiple server computers). Data store 140 can store data associated with substrate processing during a substrate process. For example, data store 140 can store data collected by set of sensors 124 before, during, or after a substrate process. Data stored by data store 140 can include historical process data (e.g., process data generated for a prior substrate processed at the manufacturing system) and/or current process data (e.g., process data generated for a current substrate processed at the manufacturing system). Data store 140 can also store spectral data or non-spectral data associated with a portion of a substrate being processed.

[0035] Data store 140 can also store contextual data associated with one or more substrates processed at the manufacturing system. Contextual data can include a recipe name, recipe step number, preventive maintenance indicator, operator, etc. Contextual data can refer to historical contextual data (e.g., contextual data associated with a prior process performed for a prior substrate) and/or current process data (e.g., contextual data associated with current process or a future process to be performed for a prior substrate). The contextual data can further include identify sensors that are associated with a particular sub-system of a processing chamber.

[0036] Data store 140 can also store task data. Task data can include one or more sets of operations to be performed for the substrate during a deposition process and can include one or more settings associated with each operation. For example, task data for a deposition process can include a temperature setting for a processing chamber, a pressure setting for a processing chamber, a flow rate setting for a precursor for a material of a film deposited on a substrate, etc. In another example, task data can include controlling pressure at a defined pressure point for the flow value. Task data can refer to historical task data (e.g., task data associated with a prior process performed for a prior substrate) and/or current task data (e.g., task data associated with current process or a future process to be performed for a substrate).

[0037] In some implementations, data store 140 can store statistical data. Statistical data can include statistics representative of the raw facility data, e.g., mean data (average), range data, standard deviation data, maximum and minimum data, median data, mode data, etc. Mean data can include a measured averages of two or more values. For example, mean data can be used to determine the average heater temperature, the processing chamber pressure, the

average flowrate of a gas, etc., during a step(s), a specific time duration, an entire process recipe, etc. Range data can include the middle observation in a set of data (e.g., a median temperature during a step). Range data can include the difference between a maximum value and a minimum value of a set of values (e.g., the range of the heater pressure during a process recipe). The standard deviation is measure of the amount of variation or dispersion of a set of values.

[0038] In some implementations, data store 140 can be configured to store data that is not accessible to a user of the manufacturing system. For example, process data, spectral data, contextual data, etc. obtained for a substrate being processed at the manufacturing system is not accessible to a user (e.g., an operator) of the manufacturing system. In some implementations, all data stored at data store 140 can be inaccessible by the user of the manufacturing system. In other or similar implementations, a portion of data stored at data store 140 can be inaccessible by the user while another portion of data stored at data store 140 can be accessible by the user. In some implementations, one or more portions of data stored at data store 140 can be encrypted using an encryption mechanism that is unknown to the user (e.g., data is encrypted using a private encryption key). In other or similar implementations, data store 140 can include multiple data stores where data that is inaccessible to the user is stored in one or more first data stores and data that is accessible to the user is stored in one or more second data stores.

[0039] In some implementations, data store 140 can be configured to store data associated with known fault patterns. A fault pattern can be a one or more values (e.g., a vector, a scalar, etc.) associated with one or more issues or failures associated with a processing chamber sub-system. In some implementations, a fault pattern can be associated with a corrective action. For example, a fault pattern can include parameter adjustment steps to correct the issue or failure indicated by the fault pattern. For example, the predictive system or the corrective action module can compare a determined fault pattern (determined from data obtained from one or more sensors of a sensor cluster) to a library of known fault patterns to determine the type of failure experienced by a sub-system, the cause of the failure, the recommended corrective action to correct the fault, and so forth.

[0040] Data store 140 can be part of or operationally connected to sensor server 140. Sensor server 140 can include one or more physical machines (e.g., server machines, desktop computers, etc.) that each include one or more processing devices communicatively coupled to memory devices and input/output (I/O) devices. The processing devices can include a

computer, microprocessor, logic device or other device or processor that is configured with hardware, firmware, and software to carry out some of the implementations described herein.

[0041] Dispatching system 150 can include dispatch server 152 and set of dispatchers 154. In some implementations, dispatching system is an RTD system that makes dispatch decisions in real-time or near real-time. Operation of dispatch server 152 can be controlled by dispatch server controller 126. Although shown as components of dispatching system 150, each of dispatch server 152 and each dispatcher of set of dispatchers 154 can be included in one or more other computing devices, such as a rackmount server, a router computer, a server computer, a personal computer, a mainframe computer, a laptop computer, a tablet computer, a desktop computer, a GPU, an ASIC, etc. Each component can execute instructions to perform any one or more of the methodologies and/or implementations described herein. The instructions can be stored on a computer readable storage medium, which can include the main memory, static memory, secondary storage and/or processing device (during execution of the instructions. Dispatch server 152 can maintain a record of which dispatchers of set of dispatchers 154 are currently initialized (“initialized dispatchers”). As will now be described below with reference to **FIG. 2**, dispatch server 152 can manage initialization of a dispatcher of set of dispatchers 154 that is not initialized (“non-initialized dispatcher”) by initiating data retrieval from an initialized dispatcher instead of a repository.

[0042] **FIG. 2** is a diagram of an example system 200 that can initialize dispatchers of a dispatching system of an electronic device manufacturing system, in accordance with some implementations of the present disclosure. As shown, system 200 includes electronic device processing system (“processing system”) 210 and dispatching system 150, as described above with reference to **FIG. 1**. In some implementations, dispatching system 150 is a remote system. For example, dispatching system 150 can be a cloud-based system.

[0043] For example, processing system 210 can include substrate processing tools (“processing tools” 122-1 through 122-*N* and dispatcher server controller 126, as described above with reference to **FIG. 1**. Processing system 210 can include any suitable number of substrate processing tools in accordance with implementations described herein.

[0044] Dispatch system 150 can include dispatch server 152 and set of dispatchers 154, as described above with reference to **FIG. 1**. Additionally, dispatch system 150 can include MES 220, MES database (DB) 230, MES data processing component (“processing component”) 240, and repository 250. MES 220 can generate raw MES data for storage in

MES DB 230. Processing component 240 can process raw MES data to generate facility data for storage in repository 250. For example, processing raw MES data can include performing an extract, transform, load (ETL) process to extract raw MES data from MES DB 230, transform the raw MES data into transformed data, and load the transformed data into repository 250. The ETL process can be used to combine raw MES data obtained from multiple data sources into a single transformed dataset for storage in repository 250. Transforming raw MES data can include improving data quality by cleansing the raw MES data.

[0045] Dispatch system 150 can include dispatch server (“DS”) 152 and set of dispatchers 154. Dispatch server 152 can manage initialization of non-initialized dispatchers of set of dispatchers 154. For example, a non-initialized dispatcher can be a new dispatcher included in set of dispatchers 154 that needs to be initialized (e.g., primed), or a dispatcher of set of dispatchers 154 that has been restarted and needs to be re-initialized (e.g., re-primed)

[0046] For example, upon identifying non-initialized dispatcher 260, DS 152 can select dispatcher 262 as an initialization partner for non-initialized dispatcher 260. An initialization partner for the non-initialized dispatcher can be selected using any suitable method. For example, DS 152 can select an idle dispatcher of the set of dispatchers as an initialization partner for the non-initialized dispatcher. If the set of dispatchers includes multiple idle dispatchers, then DS 152 can select an idle dispatcher that has been most recently updated (e.g., includes the most up-to-date dispatch decision data). Accordingly, initialized dispatcher 262 can be an idle dispatcher of set of dispatchers 154 that has been most recently updated.

[0047] DS 152 can cause non-initialized dispatcher 260 to be initialized using initialized dispatcher 262 as the initialization partner. More specifically, DS 152 can send, to non-initialized dispatcher 260, a notification that causes non-initialized dispatcher 260 to initiate data retrieval from initialized dispatcher 262. In some implementations, DS 152 sends the notification to non-initialized dispatcher 260 using an application programming interface (API) call. For example, the API call can be a representational state transfer (REST or RESTful) API call. The notification can include an identifier of initialized dispatcher 262 as the initialization partner. For example, the identifier of initialized dispatcher 262 can include an address of initialized dispatcher 262. In some implementations, the identifier of initialized dispatcher 262 includes a host name and a port number of initialized dispatcher 262. DS 152 can further cause initialized dispatcher 262 to halt all other dispatch activity until after initialization of non-initialized dispatcher 260 has been completed.

[0048] Upon receiving the notification from DS 152, non-initialized dispatcher 260 can initiate data retrieval to retrieve an initial set of dispatch decision data from initialized dispatcher 262 as the initialization partner. The initial set of dispatch decision data can include an initial set of facility data. Examples of dispatch decision data that can be included in an initial set of dispatch decision data retrieved from initialized dispatcher 262 include import files from import cache, subscribed repository table columns, cached reports (e.g., cached rules), managed cache data, unmanaged cache data, etc.

[0049] It may not be necessary for non-initialized dispatcher 260 to retrieve all dispatch decision data from initialized dispatcher 262 for initialization. For example, the initial set of dispatch decision data can include a set of base data that places non-initialized dispatcher 260 into a state that sufficiently minimizes the amount of dispatch decision data that needs to be transferred between repository 250 and dispatcher 260. The set of base data can be updated over time during normal operation of dispatcher 260 after initialization is complete.

[0050] In some implementations, retrieving the dispatch decision data from initialized dispatcher 262 includes determining whether a threshold condition related to the data retrieval has been satisfied, and terminating the data retrieval to obtain the set of base data in response to determining that the threshold condition has been satisfied.

[0051] For example, determining whether the threshold condition has been satisfied can include determining whether an amount of dispatch decision data retrieved from initialized dispatcher 262 is greater than or equal to a threshold amount of dispatch decision data. Illustratively, the threshold amount of dispatch decision data can be defined as a percentage of a total amount of dispatch decision data. As another example, determining whether the threshold condition has been satisfied can include determining whether an elapsed amount of time of data retrieval is greater than or equal to a threshold amount of time. The threshold amount of time can serve as a proxy for the threshold amount of dispatch decision data. Illustratively, the threshold amount of time can be defined as an average amount of time that is needed to obtain the threshold amount of dispatch decision data during the data retrieval process. In some implementations, the elapsed amount of time is on the order of minutes.

[0052] After non-initialized dispatcher 260 has obtained at least the set of base data and becomes an initialized dispatcher, now-initialized dispatcher 260 can send a notification to DS that initialization is complete. In some implementations, DS 152 can cause the initialized dispatcher to terminate the data retrieval. For example, DS 152 can send a termination

command to now-initialized dispatcher 260, and now-initialized dispatcher 260 can terminate the data retrieval in response to receiving the termination command. DS 152 can update an initialization status of now-initialized dispatcher 260 from non-initialized to initialized. In turn, now-initialized dispatcher 260 can serve as an initialization partner for another non-initialized dispatcher of set of dispatchers 154.

[0053] Now-initialized dispatcher 260 can store dispatch decision data in memory (e.g., cache). Upon receiving a dispatch request, now-initialized dispatcher 260 can process the dispatch request using the dispatch decision data to make a dispatch decision. For example, the dispatch decision data can be stored in cache in order to quickly retrieve and process the dispatch decision data to make the dispatch decision.

[0054] In some implementations, at least one machine learning (ML) model is trained to manage dispatcher initialization. For example, an ML model can be used to predict, based on input data related to a state of system 100 and/or system 200, an optimal selection of an initialization partner from set of dispatchers 154. As another example, an ML model can be used to control, based on input data related to the state of system 100 and/or system 200, an amount of dispatch decision data retrieved from an initialization partner during a data retrieval process. As yet another example, an ML model can be used to control, based on input data related to a state of system 100 and/or system 200, an elapsed amount of time to perform the data retrieval process.

[0055] **FIG. 3** is a flowchart of a method 300 for initializing a dispatcher of a dispatching system of a manufacturing facility, in accordance with some implementations of the present disclosure. Method 300 is performed by processing logic that can include hardware (circuitry, dedicated logic, etc.), software (such as is run on a general-purpose computer system or a dedicated machine), firmware, or some combination thereof. In some implementations, method 300 is performed by a dispatch server, such as dispatch server 152 of **FIGS. 1-2**. In some implementations, one or more operations of method 300 can be performed by one or more other components of an electronic device manufacturing system.

[0056] At operation 310, processing logic identifies, as a non-initialized dispatcher, a first dispatcher of a set of dispatchers. Each dispatcher of the set of dispatchers can be a software application that makes dispatch decisions for a manufacturing facility based on facility data related to the manufacturing facility. In some implementations, the manufacturing facility includes an electronic device manufacturing system. For example, a dispatch decision can

relate to processing of at least one substrate using at least one processing tool of the electronic device manufacturing system. In some implementations, the first dispatcher is a new dispatcher of the set of dispatchers that needs to be initialized (e.g., primed). In some implementations, the first dispatcher is a dispatcher that has been restarted and needs to be re-initialized (e.g., re-primed).

[0057] At operation 320, processing logic selects, as an initialization partner, a second dispatcher of the set of dispatchers. The second dispatcher can be selected as an initialization partner using any suitable method. For example, selecting the second dispatcher as an initialization partner can include selecting an idle dispatcher of the set of dispatchers as the initialization partner. If the set of dispatchers includes multiple idle dispatchers, then selecting the second dispatcher as an initialization partner can include selecting, as the initialization partner, an idle dispatcher of the set of dispatchers that has been most recently updated (e.g., includes the most up-to-date dispatch decision data). Accordingly, the second dispatcher selected as an initialization partner can be a most recently updated idle dispatcher of the set of dispatchers.

[0058] At operation 330, processing logic causes the first dispatcher to initiate a data retrieval process to retrieve an initial set of dispatch decision data from the second dispatcher. The initial set of dispatch decision data can include an initial set of facility data. In some implementations, the initial set of dispatch decision data includes at least one of: import files from import cache, subscribed repository table columns, cached reports (e.g., cached rules), managed cache data, unmanaged cache data, etc. In some implementations, the initial set of facility data describes at least one of: a set of substrate processing tools of the manufacturing facility, substrate routes between substrate processing tools of the set of substrate processing tools, a set of process recipes, etc.

[0059] Causing the first dispatcher to initiate the data retrieval process can include sending a notification to the first dispatcher that causes the first dispatcher to initiate the data retrieval process. In some implementations, sending the notification to the first dispatcher includes using an API call. For example, the API call can be a REST or RESTful API call. The notification can include an identifier of the second dispatcher as the initialization partner. For example, the identifier of the second dispatcher can include an address of the second dispatcher. In some implementations, the identifier of the second dispatcher includes a host name and a port number of the second dispatcher. Processing logic can cause the second

dispatcher to halt all other dispatch activity until after initialization of the first dispatcher has been completed.

[0060] In some embodiments, the initial set of dispatch decision data includes a set of base data that places the first dispatcher into a state that sufficiently minimizes the amount of dispatch decision data that needs to be transferred between a repository storing dispatch decision data and the first dispatcher. The set of base data can be updated over time by obtaining updated dispatch decision data from the repository during normal operation of the first dispatcher after initialization is complete.

[0061] In some implementations, retrieving the initial set of dispatch decision data from the second dispatcher includes determining whether a threshold condition related to the data retrieval has been satisfied, and terminating the data retrieval to obtain the set of base data in response to determining that the threshold condition has been satisfied. For example, determining whether the threshold condition has been satisfied can include determining whether an amount of dispatch decision data retrieved from the second dispatcher is greater than or equal to a threshold amount of dispatch decision data. Illustratively, the threshold amount of dispatch decision data can be defined as a percentage of a total amount of dispatch decision data. As another example, determining whether the threshold condition has been satisfied can include determining whether an elapsed amount of time of data retrieval is greater than or equal to a threshold amount of time. The threshold amount of time can serve as a proxy for the threshold amount of dispatch decision data. Illustratively, the threshold amount of time can be defined as an average amount of time that is needed to obtain the threshold amount of dispatch decision data during the data retrieval process. In some implementations, the elapsed amount of time is on the order of minutes.

[0062] At operation 340, processing logic determines that the data retrieval process is complete and, at operation 350, processing logic terminates the data retrieval process. In some implementations, terminating the data retrieval process includes causing the first dispatcher to terminate the data retrieval. For example, terminating the initialization process can include sending a termination command to the first dispatcher to terminate the data retrieval process.

[0063] At operation 360, processing logic updates a status of the first dispatcher to an initialized dispatcher. For example, a dispatch server can maintain a record of which dispatchers of the set of dispatchers are initialized dispatchers. By doing so, the dispatch

server can select the first dispatcher to be a future initialization partner for a future non-initialized dispatcher.

[0064] In some implementations, at least one ML model is trained to manage dispatcher initialization. For example, an ML model can be used to predict, based on input data related to a state of the electronic device manufacturing system, an optimal selection of the second dispatcher as an initialization partner from the set of dispatchers. As another example, an ML model can be used to control, based on input data related to the state of electronic device manufacturing system, an amount of dispatch decision data retrieved from an initialization partner during a data retrieval process. As yet another example, an ML model can be used to control, based on input data related to a state of the electronic device manufacturing system, an elapsed amount of time to perform the data retrieval process. Further details regarding operations 310-360 are described above with reference to **FIGS. 1-2** and will be described in further detail below with reference to **FIGS. 4-5**.

[0065] **FIG. 4** is a flowchart of a method 400 for initializing a dispatcher of a dispatching system of a manufacturing facility, in accordance with some implementations of the present disclosure. Method 400 is performed by processing logic that can include hardware (circuitry, dedicated logic, etc.), software (such as is run on a general-purpose computer system or a dedicated machine), firmware, or some combination thereof. In some implementations, method 400 is performed by a non-initialized dispatcher, such as dispatcher 260 of **FIG. 2**. In some implementations, one or more operations of method 400 can be performed by one or more other components of an electronic device manufacturing system.

[0066] At operation 410, processing logic receives a notification to initiate a data retrieval process to retrieve an initial set of dispatch decision data from a dispatcher, and at operation 420, processing logic initiates the data retrieval process in response to receiving the notification. The dispatcher can be a software application that makes dispatch decisions for a manufacturing facility based on facility data related to the manufacturing facility. In some implementations, the manufacturing facility includes an electronic device manufacturing system. For example, a dispatch decision can relate to processing of at least one substrate using at least one processing tool of the electronic device manufacturing system.

[0067] More specifically, the notification can be received from a dispatcher server and the dispatcher can be an initialized dispatcher. In some implementations, the notification is received using an API call. For example, the API call can be a REST or RESTful API call.

The notification can include an identifier of the dispatcher as an initialization partner. For example, the identifier of the dispatcher can include an address of the dispatcher. In some implementations, the identifier of dispatcher includes a host name and a port number of the dispatcher.

[0068] The initial set of dispatch decision data can include an initial set of facility data. In some implementations, the initial set of dispatch decision data includes at least one of: import files from import cache, subscribed repository table columns, cached reports (e.g., cached rules), managed cache data, unmanaged cache data, etc. In some implementations, the initial set of facility data describes at least one of: a set of substrate processing tools of the manufacturing facility, substrate routes between substrate processing tools of the set of substrate processing tools, a set of process recipes, etc.

[0069] At operation 430, processing logic terminates the data retrieval process. In some embodiments, the initial set of dispatch decision data retrieved from the dispatcher includes a set of base data that places a non-initialized dispatcher into a state that sufficiently minimizes the amount of dispatch decision data that needs to be transferred between a repository storing dispatch decision data and the non-initialized dispatcher. The set of base data can be updated over time by obtaining updated dispatch decision data from the repository during normal operation of the non-initialized dispatcher after initialization is complete.

[0070] In some implementations, retrieving the dispatch decision data from the dispatcher includes determining whether a threshold condition related to the data retrieval has been satisfied, and terminating the data retrieval to obtain the set of base data in response to determining that the threshold condition has been satisfied. For example, determining whether the threshold condition has been satisfied can include determining whether an amount of dispatch decision data retrieved from the dispatcher is greater than or equal to a threshold amount of dispatch decision data. Illustratively, the threshold amount of dispatch decision data can be defined as a percentage of a total amount of dispatch decision data. As another example, determining whether the threshold condition has been satisfied can include determining whether an elapsed amount of time of data retrieval is greater than or equal to a threshold amount of time. The threshold amount of time can serve as a proxy for the threshold amount of dispatch decision data. Illustratively, the threshold amount of time can be defined as an average amount of time that is needed to obtain the threshold amount of dispatch decision data during the data retrieval process. Further details regarding operations

410-430 are described above with reference to **FIGS. 1-3** and will be described in further detail below with reference to **FIG. 5**.

[0071] **FIG. 5** is a flowchart of a method 500 for initializing a dispatcher of a dispatching system of a manufacturing facility, in accordance with some implementations of the present disclosure. Method 500 is performed by processing logic that can include hardware (circuitry, dedicated logic, etc.), software (such as is run on a general-purpose computer system or a dedicated machine), firmware, or some combination thereof. In some implementations, method 500 is performed by a dispatcher selected as an initialization partner, such as dispatcher 262 of **FIG. 2**. In some implementations, one or more operations of method 500 can be performed by one or more other components of an electronic device manufacturing system.

[0072] At operation 510, processing logic halts dispatch activity. For example, a dispatch server can cause the dispatch activity to be halted.

[0073] At operation 520, processing logic sends an initial set of dispatch decision data to a non-initialized dispatcher. More specifically, the initial set of dispatch decision data can be sent to the non-initialized dispatcher during a data retrieval process to initialize the non-initialized dispatcher. The initial set of dispatch decision data can include an initial set of facility data. In some implementations, the initial set of dispatch decision data includes at least one of: import files from import cache, subscribed repository table columns, cached reports (e.g., cached rules), managed cache data, unmanaged cache data, etc. In some implementations, the initial set of facility data describes at least one of: a set of substrate processing tools of the manufacturing facility, substrate routes between substrate processing tools of the set of substrate processing tools, a set of process recipes, etc.

[0074] At operation 530, processing logic resumes dispatch activity. For example, the dispatch server can cause the dispatch activity to be resumed after completion of the data retrieval process. Further details regarding operations 510-530 are described above with reference to **FIGS. 1-4**.

[0075] **FIG. 6** is a block diagram illustrating a computer system 600, according to certain implementations. In some implementations, computer system 600 can be connected (e.g., via a network, such as a Local Area Network (LAN), an intranet, an extranet, or the Internet) to other computer systems. Computer system 600 can operate in the capacity of a server or a client computer in a client-server environment, or as a peer computer in a peer-to-peer or

distributed network environment. Computer system 600 can be provided by a personal computer (PC), a tablet PC, a Set-Top Box (STB), a Personal Digital Assistant (PDA), a cellular telephone, a web appliance, a server, a network router, switch or bridge, or any device capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that device. Further, the term "computer" shall include any collection of computers that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methods described herein.

[0076] In a further aspect, computer system 600 can include processing device 602, volatile memory 604 (e.g., random-access memory (RAM)), non-volatile memory 606 (e.g., read-only memory (ROM) or electrically-erasable programmable ROM (EEPROM)), and data storage device 616, which can communicate with each other via bus 608.

[0077] Processing device 602 can be provided by one or more processors such as a general purpose processor (such as, for example, a complex instruction set computing (CISC) microprocessor, a reduced instruction set computing (RISC) microprocessor, a very-long instruction word (VLIW) microprocessor, a microprocessor implementing other types of instruction sets, or a microprocessor implementing a combination of types of instruction sets) or a specialized processor (such as, for example, an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), a digital signal processor (DSP), or a network processor).

[0078] Computer system 600 can further include network interface device 622 (e.g., coupled to network 674). Computer system 600 also can include video display unit 610 (e.g., an LCD), alphanumeric input device 612 (e.g., a keyboard), cursor control device 614 (e.g., a mouse), and signal generation device 620.

[0079] In some implementations, data storage device 616 can include non-transitory computer-readable storage medium 624 on which can store instructions 626 encoding any one or more of the methods or functions described herein, including methods or functions implemented by dispatch server 152 and/or dispatcher 260.

[0080] Instructions 626 can also reside, completely or partially, within volatile memory 604 and/or within processing device 602 during execution thereof by computer system 600, hence, volatile memory 604 and processing device 602 can also constitute machine-readable storage media.

[0081] While computer-readable storage medium 624 is shown in the illustrative examples as a single medium, the term "computer-readable storage medium" shall include a single medium or multiple media (e.g., a centralized or distributed database, and/or associated caches and servers) that store the one or more sets of executable instructions. The term "computer-readable storage medium" shall also include any tangible medium that is capable of storing or encoding a set of instructions for execution by a computer that cause the computer to perform any one or more of the methods described herein. The term "computer-readable storage medium" shall include, but not be limited to, solid-state memories, optical media, and magnetic media.

[0082] The methods, components, and features described herein can be implemented by discrete hardware components or can be integrated in the functionality of other hardware components such as ASICs, FPGAs, DSPs or similar devices. In addition, the methods, components, and features can be implemented by firmware modules or functional circuitry within hardware devices. Further, the methods, components, and features can be implemented in any combination of hardware devices and computer program components, or in computer programs.

[0083] Unless specifically stated otherwise, terms such as "receiving," "performing," "providing," "obtaining," "causing," "accessing," "determining," "adding," "using," "training," or the like, refer to actions and processes performed or implemented by computer systems that manipulates and transforms data represented as physical (electronic) quantities within the computer system registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices. Also, the terms "first," "second," "third," "fourth," etc. as used herein are meant as labels to distinguish among different elements and cannot have an ordinal meaning according to their numerical designation.

[0084] Examples described herein also relate to an apparatus for performing the methods described herein. This apparatus can be specially constructed for performing the methods described herein, or it can include a general purpose computer system selectively programmed by a computer program stored in the computer system. Such a computer program can be stored in a computer-readable tangible storage medium.

[0085] The methods and illustrative examples described herein are not inherently related to any particular computer or other apparatus. Various general purpose systems can be used in

accordance with the teachings described herein, or it can prove convenient to construct more specialized apparatus to perform methods described herein and/or each of their individual functions, routines, subroutines, or operations. Examples of the structure for a variety of these systems are set forth in the description above.

[0086] The above description is intended to be illustrative, and not restrictive. Although the present disclosure has been described with references to specific illustrative examples and implementations, it will be recognized that the present disclosure is not limited to the examples and implementations described. The scope of the disclosure should be determined with reference to the following claims, along with the full scope of equivalents to which the claims are entitled.

CLAIMS

What is claimed is:

1. A method of operating a set of dispatchers for a manufacturing facility, wherein each dispatcher of the set of dispatchers is a software application that makes dispatch decisions for the manufacturing facility based on facility data related to the manufacturing facility, the method comprising:

identifying a first dispatcher of the set of dispatchers, wherein the first dispatcher is a non-initialized dispatcher;

selecting, as an initialization partner, a second dispatcher of the set of dispatchers; and
causing the first dispatcher to initiate a data retrieval process with the second dispatcher to retrieve, from the second dispatcher, an initial set of facility data related to the manufacturing facility.

2. The method of claim 1, wherein the first dispatcher is one of: a new dispatcher of the set of dispatchers, or a dispatcher that has been restarted.

3. The method of claim 1, wherein selecting the second dispatcher comprises identifying a most recently updated idle dispatcher of the set of dispatchers.

4. The method of claim 1, wherein causing the first dispatcher to initiate the initialization process with the second dispatcher comprises sending, to the first dispatcher, a notification comprising an identifier of the second dispatcher as the initialization partner, and wherein the identifier comprises a host name and a port number of the second dispatcher.

5. The method of claim 1, wherein the initial set of facility data comprises at least one of: import files from import cache, subscribed repository table columns, cached reports, managed cache data, or unmanaged cache data.

6. The method of claim 1, wherein the initial set of facility data describes at least one of: a set of substrate processing tools of the manufacturing facility, substrate routes between substrate processing tools of the set of substrate processing tools, or a set of process recipes.

7. The method of claim 1, further comprising:
 - determining that the data retrieval process is complete;
 - responsive to determining that the data retrieval process is complete, terminating the data retrieval process; and
 - updating a status of the first dispatcher to an initialized dispatcher.
8. The method of claim 1, further comprising:
 - initiating the first dispatcher using the initial set of facility data to obtain an initialized first dispatcher; and
 - after initiating the first dispatcher using the initial set of dispatch decision data, using the initiated first dispatcher to cause a substrate to be dispatched to a substrate processing tool for processing.
9. The method of claim 1, further comprising:
 - receiving, by the first dispatcher from a dispatch server, a notification to initiate the data retrieval process;
 - initiating, by the first dispatcher, the data retrieval process to retrieve the initial set of facility data from the second dispatcher;
 - determining that the data retrieval process is complete; and
 - responsive to determining that the data retrieval process is complete, terminating the data retrieval process.
10. The method of claim 9, wherein determining that the data retrieval process is complete comprises determining whether a threshold condition related to the data retrieval has been satisfied.
11. The method of claim 10, wherein determining that the threshold condition has been satisfied comprises determining that an amount of facility data retrieved from the second dispatcher is greater than or equal to a threshold amount of data.
12. A system for operating a set of dispatchers for a manufacturing facility, wherein each dispatcher of the set of dispatchers is a software application that makes dispatch decisions for the manufacturing facility based on facility data related to the manufacturing facility, the system comprising:

a memory; and
a processing device, operatively coupled to the memory, to perform operations comprising:
 identifying a first dispatcher of the set of dispatchers, wherein the first dispatcher is a non-initialized dispatcher;
 selecting, as an initialization partner, a second dispatcher of the set of dispatchers; and
 causing the first dispatcher to initiate a data retrieval process with the second dispatcher to retrieve, from the second dispatcher, an initial set of facility data related to the manufacturing facility.

13. The system of claim 12, wherein the first dispatcher is one of: a new dispatcher of the set of dispatchers, or a dispatcher that has been restarted.

14. The system of claim 12, wherein selecting the second dispatcher comprises identifying a most recently updated idle dispatcher of the set of dispatchers.

15. The system of claim 12, wherein causing the first dispatcher to initiate the initialization process with the second dispatcher comprises sending, to the first dispatcher, a notification comprising an identifier of the second dispatcher as the initialization partner, and wherein the identifier comprises a host name and a port number of the second dispatcher.

16. The system of claim 12, wherein the initial set of facility data comprises at least one of: import files from import cache, subscribed repository table columns, cached reports, managed cache data, or unmanaged cache data.

17. The system of claim 12, wherein the initial set of facility data at least describes at least one of: a set of substrate processing tools of the manufacturing facility, substrate routes between substrate processing tools of the set of substrate processing tools, or a set of process recipes.

18. The system of claim 12, wherein the operations further comprise:
 determining that the data retrieval process is complete;

responsive to determining that the data retrieval process is complete, terminating the data retrieval process; and

updating a status of the first dispatcher to an initialized dispatcher.

19. The system of claim 12, wherein the operations further comprise:
initiating the first dispatcher using the initial set of facility data to obtain an initialized first dispatcher; and

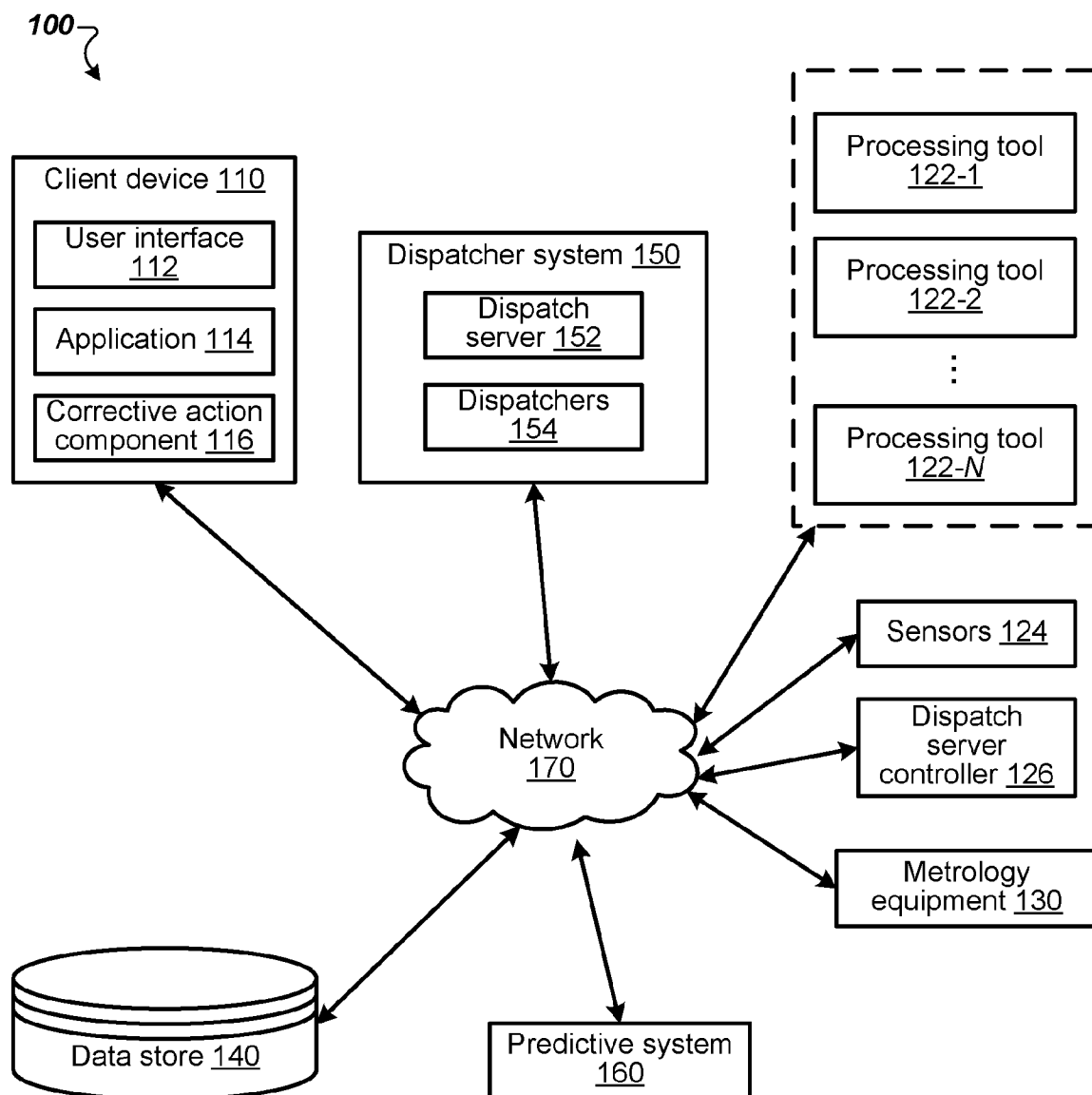
after initiating the first dispatcher using the initial set of dispatch decision data, using the initiated first dispatcher to cause a substrate to be dispatched to a substrate processing tool for processing.

20. The system of claim 12, wherein the operations further comprise:
receiving, by the first dispatcher from a dispatch server, a notification to initiate the data retrieval process;

initiating, by the first dispatcher, the data retrieval process to retrieve the initial set of facility data from the second dispatcher;

determining that the data retrieval process is complete, wherein determining that the data retrieval process is complete comprises determining whether a threshold condition related to the data retrieval has been satisfied; and

responsive to determining that the data retrieval process is complete, terminating the data retrieval process.

**FIG. 1**

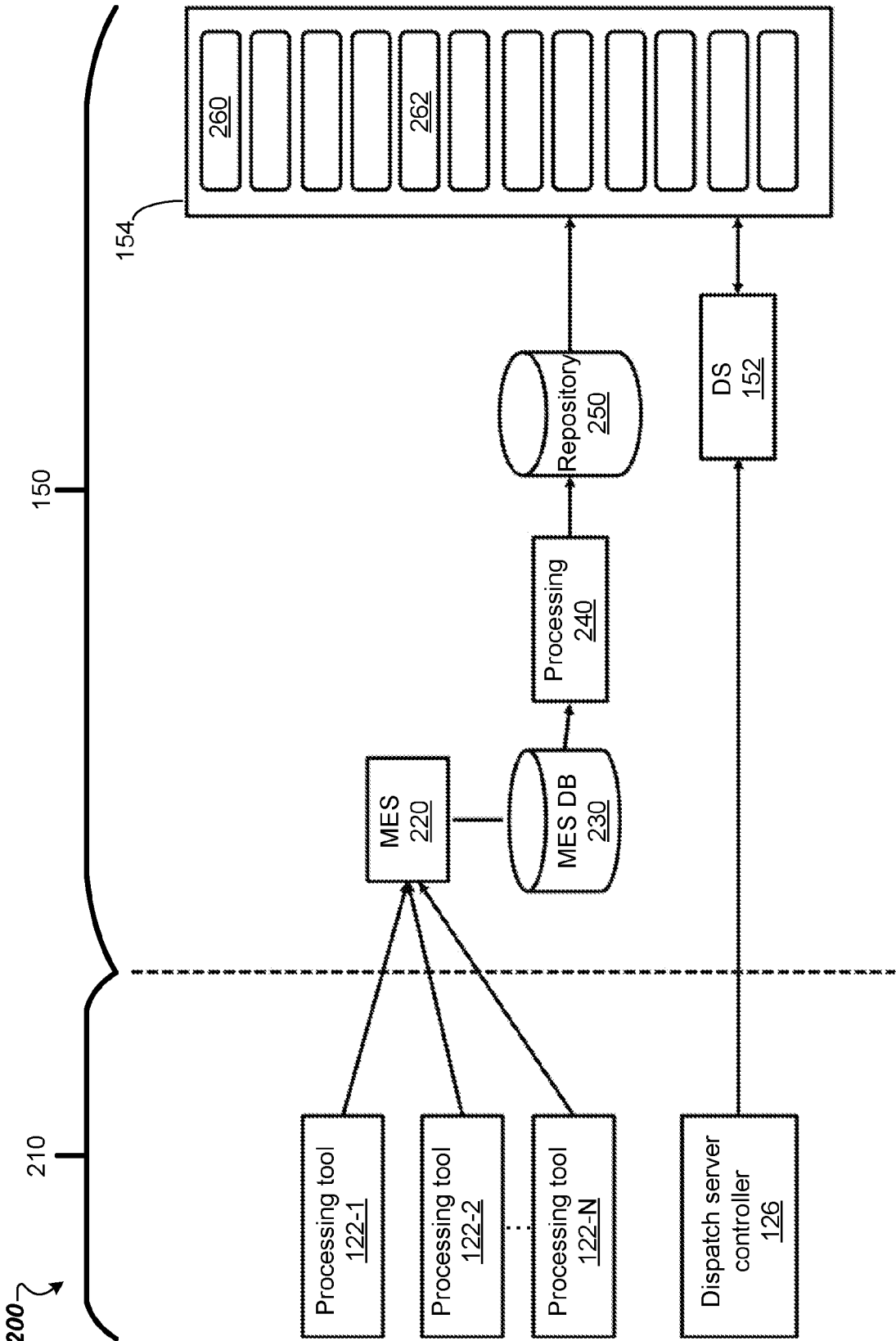
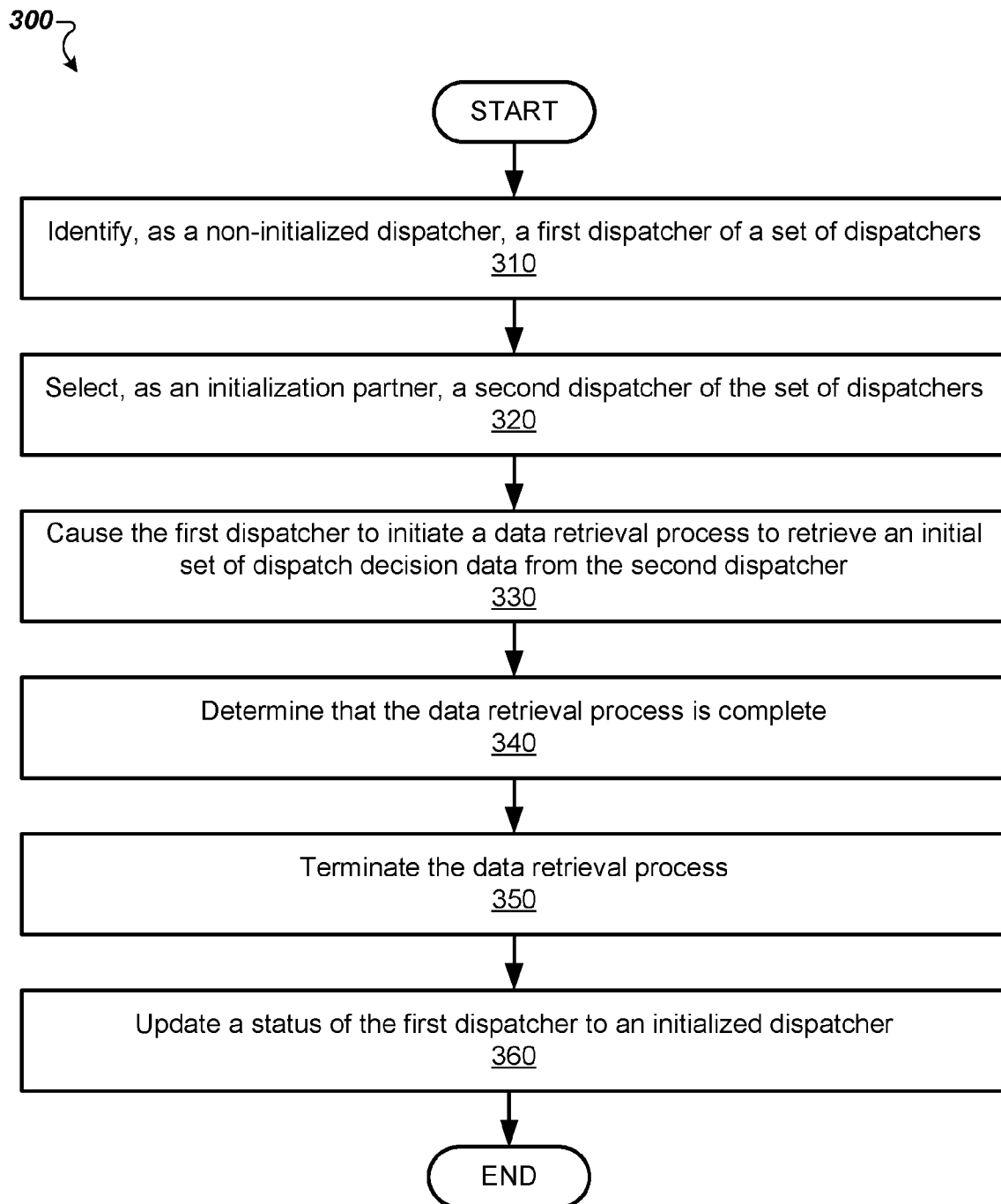
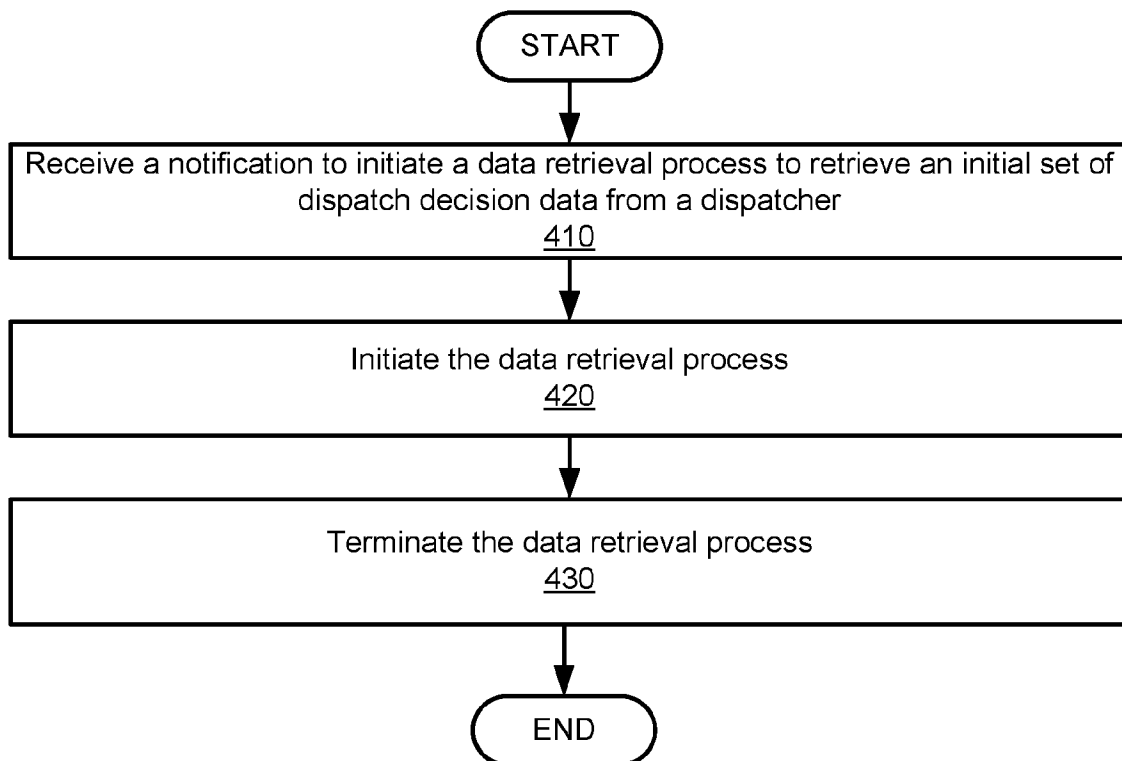


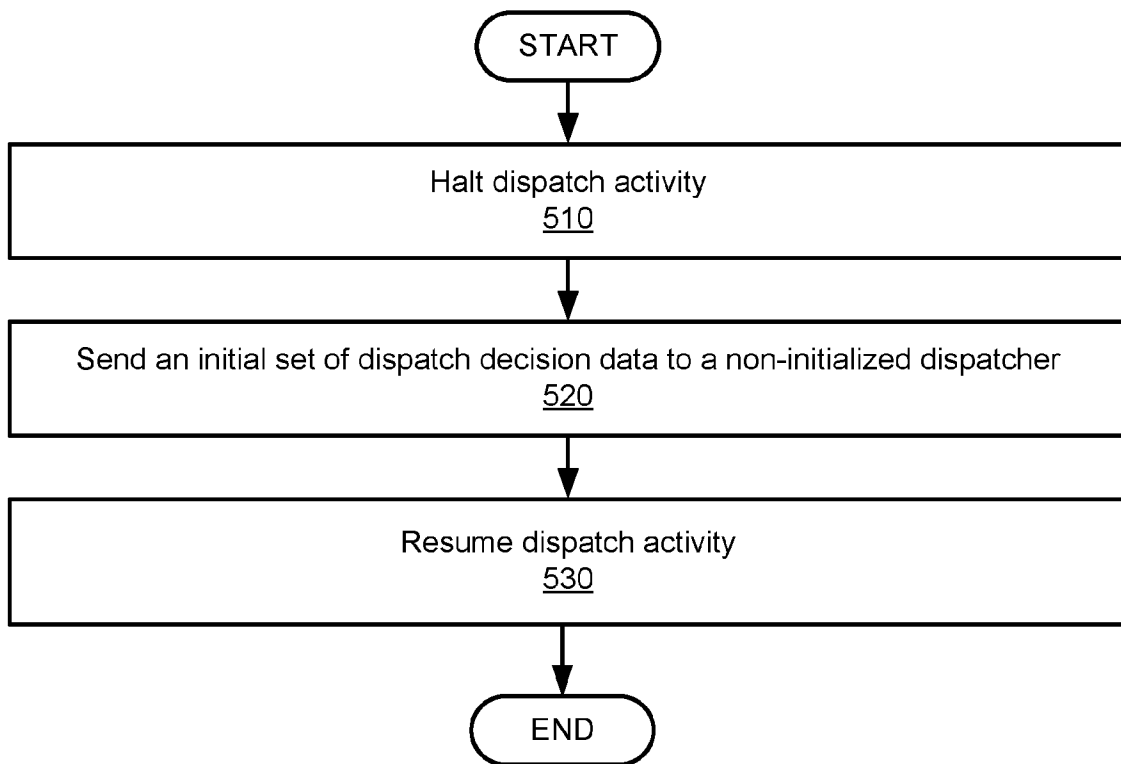
FIG. 2

**FIG. 3**

400

**FIG. 4**

500

**FIG. 5**

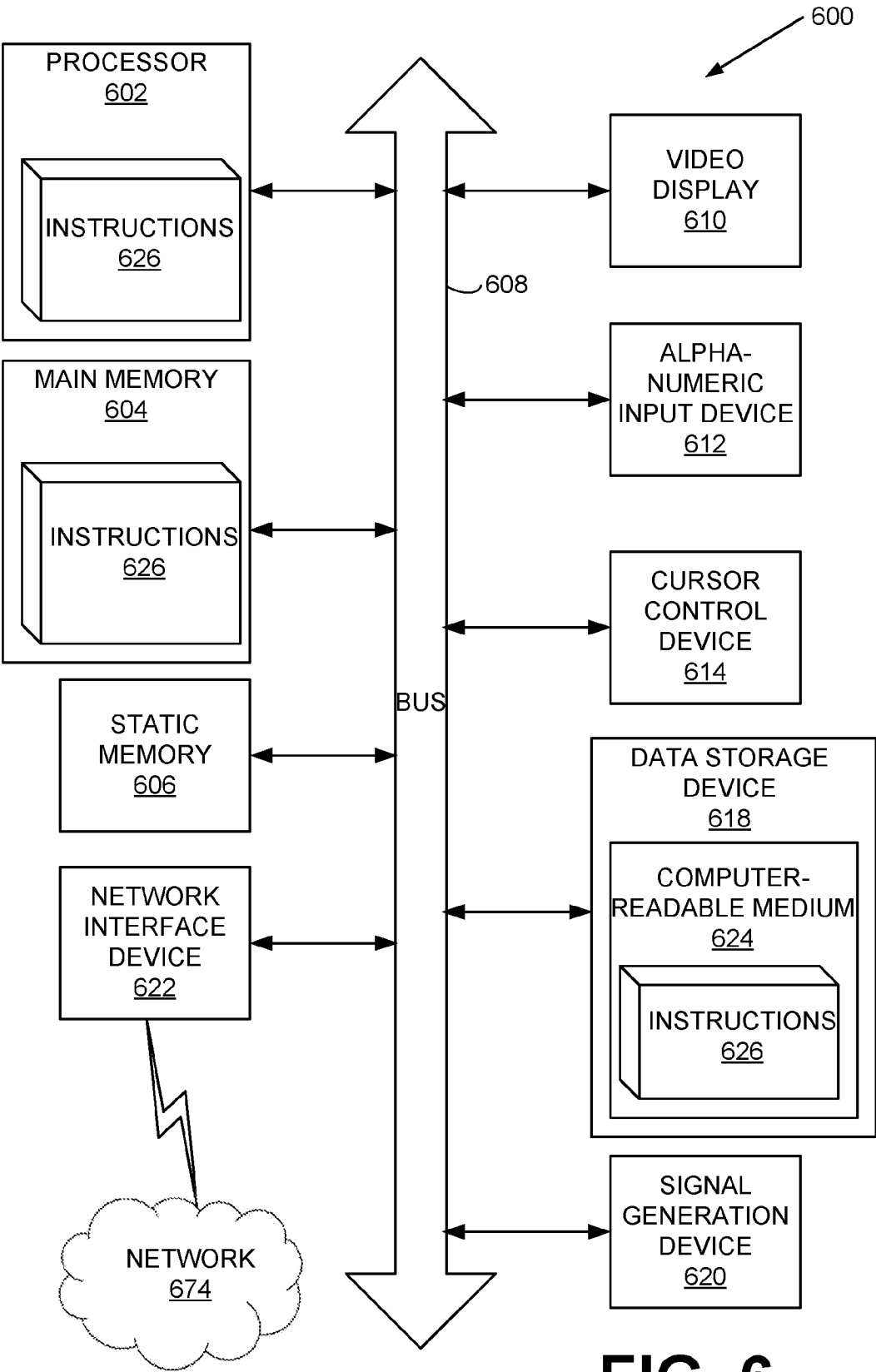


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2024/036211

A. CLASSIFICATION OF SUBJECT MATTER G05B 19/418(2006.01)i; G05B 23/02(2006.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G05B 19/418(2006.01); G05B 13/02(2006.01); G06Q 50/04(2012.01); H04L 29/06(2006.01) Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: semiconductor, wafer, manufacture, dispatcher, software, application, facility, decision, identify, select initialization, partner, retrieval, process		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2021-0263505 A1 (YANGTZE MEMORY TECHNOLOGIES CO., LTD.) 26 August 2021 (2021-08-26) paragraphs [0018]-[0081]; and figures 1-6	1-20
A	US 2005-0187649 A1 (MERRITT FUNK et al.) 25 August 2005 (2005-08-25) paragraphs [0022]-[0227]; and figures 1-12	1-20
A	EP 3598713 A1 (CODESYS HOLDING GMBH) 22 January 2020 (2020-01-22) paragraphs [0057]-[0113]; and figures 1-10	1-20
A	US 2004-0039469 A1 (FRANK LIN) 26 February 2004 (2004-02-26) paragraphs [0048]-[0089]; and figures 1-4	1-20
A	KR 10-2013-0141743 A (VMS SOLUTIONS CO., LTD.) 27 December 2013 (2013-12-27) paragraphs [0011]-[0044]; and figures 1-5	1-20
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: “A” document defining the general state of the art which is not considered to be of particular relevance “D” document cited by the applicant in the international application “E” earlier application or patent but published on or after the international filing date “L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) “O” document referring to an oral disclosure, use, exhibition or other means “P” document published prior to the international filing date but later than the priority date claimed “T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art “&” document member of the same patent family		
Date of the actual completion of the international search 30 September 2024		Date of mailing of the international search report 30 September 2024
Name and mailing address of the ISA/KR Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon 35208, Republic of Korea Facsimile No. +82-42-481-8578		Authorized officer KIM, Sung Hee Telephone No. +82-42-481-3516

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/US2024/036211

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2021-0263505	A1	26 August 2021	CN	111356988	A	30 June 2020
				CN	111356988	B	12 March 2021
				TW	202133097	A	01 September 2021
				TW	1788650	B	01 January 2023
				US	11385627	B2	12 July 2022
				WO	2021-163986	A1	26 August 2021
US	2005-0187649	A1	25 August 2005	AU	2003-270866	A1	23 April 2004
				CN	100407215	C	30 July 2008
				CN	1682165	A	12 October 2005
				EP	1546827	A1	29 June 2005
				JP	2006-501680	A	12 January 2006
				JP	4694843	B2	08 June 2011
				KR	10-1025527	B1	04 April 2011
				KR	10-2005-0061498	A	22 June 2005
				TW	200415453	A	16 August 2004
				TW	I233008	B	21 May 2005
				US	7123980	B2	17 October 2006
				WO	2004-031875	A1	15 April 2004
EP	3598713	A1	22 January 2020	EP	3598713	B1	07 June 2023
US	2004-0039469	A1	26 February 2004	US	6748282	B2	08 June 2004
KR	10-2013-0141743	A	27 December 2013	KR	10-1371006	B1	25 March 2014