Towards Structured Performance Analysis of Industry 4.0 Workflow Automation Resources

Ajay Kattepur
Embedded Systems and Robotics
TCS Research & Innovation, Bangalore, India.
ajay.kattepur@tcs.com

ABSTRACT

Automation and the use of robotic components within business processes is in vogue across retail and manufacturing industries. However, a structured way of analyzing performance improvements provided by automation in complex workflows is still at a nascent stage. In this paper, we consider the common Industry 4.0 automation workflow resource patterns and model them within a hybrid queuing network. The queuing stations are replaced by *scale up*, *scale out* and *hybrid scale* automation patterns, to examine improvements in end-to-end process performance. We exhaustively simulate the throughput, response time, utilization and operating costs at higher concurrencies using Mean Value Analysis (MVA) algorithms. The queues are analyzed for cases with multiple classes, batch/transactional processing and load dependent service demands. These solutions are demonstrated over an exemplar use case of automation in Industry 4.0 warehouse automation workflows. A structured process of automation workflow performance analysis will prove valuable across industrial deployments.

CCS CONCEPTS

• Mathematics of computing → Queueing theory; **• Applied computing → Business process modeling**; **Supply chain management**; **• Computer systems organization → Robotic components**.

KEYWORDS

Workflow Resource Patterns; Queuing Network; Mean Value Analysis; Industry 4.0 Warehouse Automation.

ACM Reference Format:
Ajay Kattepur. 2019. Towards Structured Performance Analysis of Industry 4.0 Workflow Automation Resources. In *Tenth ACM/SPEC International Conference on Performance Engineering (ICPE '19), April 7–11, 2019, Mumbai, India.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3297663.3309671

1 INTRODUCTION

Industrial automation [1] has pervaded multiple industries with autonomous robots, Internet of Things (IoT) [2] and software systems replacing human participants in the retail and manufacturing industries. Industry 4.0 [3] has further emphasized these requirements, with robotics and process automation systems intended to replace mundane and repetitive industrial tasks. Integrating Artificial Intelligence [4] into robotic automation is also mandated, which would enable autonomous and smart deployments.

One of these applications is in the warehouse inventory management space [5], where routine tasks of procurement, product picking and placement may be performed more efficiently by autonomous robots. Deployments using the Kiva System [6] by Amazon[1], is one such example, which has shown to improve picking efficiency in large warehouses. While these systems are readily integrated into traditional business processes [7], overall performance improvements are yet to be formally characterized. In addition, models which can handle traditional human participants, software automation and robotics in an integrated framework are needed.

In this paper, we intend to study the end-to-end performance improvements provided by automation systems, when integrated into traditional business processes [7]. We draw inspiration from workflow resource patterns [9], that model task creation, assignment and execution in workflows. While such models have been proposed for software processes with human participants, we extend these models for automation resources involving coordination among robotic agents, human participants and business processes.

Workflow resource patterns are used in conjunction with queuing network models [10] to accurately characterize the end-to-end performance of complex processes. Using the example of an automation workflow in Industry 4.0 warehouses, scenarios such as First-In-First-Out (FIFO) order fulfillment, batch processing, load dependency and dynamic variation in resource requirements are modeled. This is, in turn, studied with a hybrid queuing network that can handle multiple classes, load dependency and scaling up/out of resource patterns. This queuing network is evaluated using Mean Value Analysis (MVA) [10] algorithms to estimate throughput, latency and resource bottlenecks under different conditions. This allows us to propose accurate resource patterns to maximize throughput or minimize cost under various operational conditions. This has to be extended to cases where automation resources may *scale up*, *scale out* or *hybrid scale* in order to maintain execution for varying demand rates. The use of MVA and queuing network models allows estimation of throughput and utilization levels at higher loads, which can lead to runtime adaptation.

ICPE '19, April 7–11, 2019, Mumbai, India

ACM ISBN 978-1-4503-6239-9/19/04…$15.00
https://doi.org/10.1145/3297663.3309671

[1] https://www.amazonrobotics.com/

We demonstrate these approaches on Industry 4.0 warehouse automation systems, where picking and stowaway tasks may be replaced with autonomous robotic elements. Using hybrid queuing network models, cases when transactional or batch jobs are processed by the warehouse are evaluated. Through simulations, it is seen that 100–200% improvements in throughput and 70% reduction in cost per transaction handled by the warehouse, when efficient automation workflow patterns are employed. Granular performance analysis provides a systematic technique to study automation workflows.

The Principal Contributions of this paper are:

(1) Systematic analysis of workflow resource models using a network of queuing centers.
(2) Thorough performance analysis when automation entails parallel processing, superior service demands or delegation of workload.
(3) Mean Value Analysis for higher concurrency loads with multiple classes, load dependent service demands and probabilistic completion rates considered.
(4) Accurate characterization of automation resource patterns, under various operational environments – leading to suggestions on scaling up/out.
(5) An Industry 4.0 warehouse automation case study, demonstrating structured performance improvements.

The rest of this paper is organized as follows: Section 2 provides an overview of Industry 4.0 warehouse automation resources and autonomous robots that operate in them. Resource models using both queuing networks and workflow patterns are studied in Section 3. The application of performance laws and Mean Value Analysis techniques to evaluate the queuing network models are described in Section 4. In Section 5, simulations are performed to analyze single class and multi class models in warehouse deployments. This is followed by related work and conclusions of the paper.

2 INDUSTRY 4.0 AUTOMATION

In this section, we provide an overview of activities involved in Industry 4.0 warehouses and various order-fulfillment strategies. An overview of autonomous robots that are deployed in picking and delivery tasks are also provided.

2.1 Industrial Warehouses

Multi-party, multi-supplier warehouses [5] have been used in the retail and manufacturing industries as buffers for varying demands. In addition, they may serve ancillary activities such as packaging, labeling and localized distribution. Fig. 1 provides an overview of various activities taking place in multi supplier warehouses. Stock procurement deliveries are periodically received that may be put-away in forward or reserve locations. This stock is then consumed by orders that are periodically received. In such warehouses, it is important to analyze the end-to-end efficiency and throughput, when subjected to varying demand rates. Use of IoT and automation systems may also be scaled up, depending on performance deterioration in certain cases.

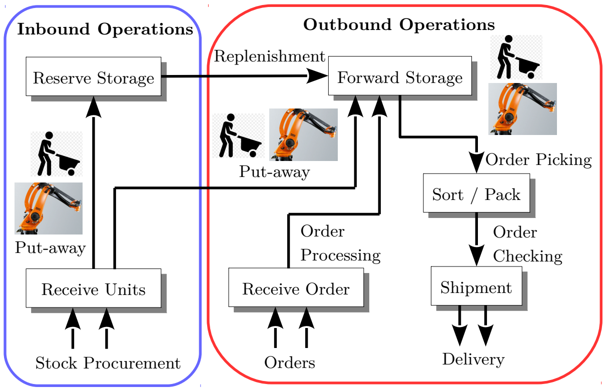


Figure 1: Industrial Warehouse Operation Processes.

In Industry 4.0 warehouses [6][8], there is increased demand to make use of robots such as KUKA KMR [2] to automate tasks such as put-away and order picking. Such robotic systems may replace or work hand-in-hand with human participants to complete tasks. While significant work has gone into inventory and supply chain optimization [11], we intend to use the abstraction of robotic automation components and workflow performance analysis to study warehouse operations. This combines both workflow processes with performance modeling allowing extensions to other deployments.

2.2 Robotic Automation Agents

In order to model the robotic components in warehouses, we make use of the *Intelligent Agent* [4] abstraction. Typical agent actions, for instance with a order picking robot in an Industry 4.0 warehouse, include:

(1) *Goals*: Understanding goals of each task and subtask, such as, placing correct parts into correct bins within the given time constraints.
(2) *Perception*: Object identification and obstacle detection using camera and odometry sensors that sense the robot's environment.
(3) *Actions*: Identifying granular actionable subtasks, such as, moving to particular location, picking up parts of orders or sorting objects. Constraints may be placed on the robot capabilities, motion plans and accuracy in performing such actions.
(4) *Knowledge Base*: The knowledge base coordinates the appropriate action in relation to an individual robot's perception. The knowledge base should also include descriptions of domain ontology, task templates, algorithmic implementations and resource descriptions.

In order to effectively study the effect of introducing such participants on performance measures such as end-to-end throughput or latency, a structured approach to model resource allocation in complex workflows are needed. These are modeled within a queuing network, described next.

[2] https://www.kuka.com/en-in/products/mobility/mobile-robots/kmr-iiwa

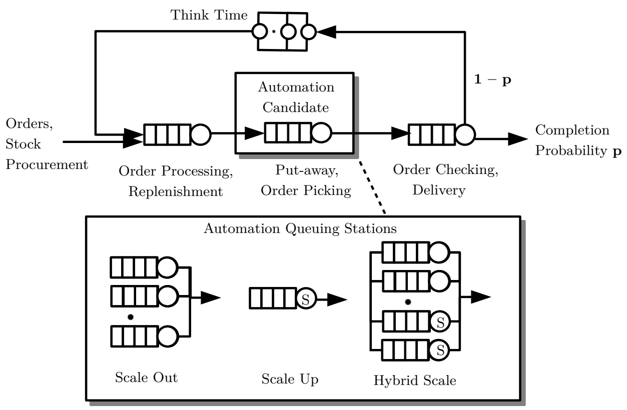


Figure 2: Queuing Network for Warehouse Automation.

3 RESOURCE MODELING

We describe hybrid queuing network models that may be used to model various stations in industrial warehouses. The stations are mapped to workflow resource patterns, that formally describe workflow execution tasks in industrial settings.

3.1 Queuing Model

Queuing network models [10] have often been used to model manufacturing, enterprise and software performance [12]. While both open and closed queues are used in particular domains, a hybrid modeling approach has been proposed in [13], that can handle combinations of both these systems.

Fig. 2 provides a hybrid queuing network model for activities that are undertaken in automated warehouses (refer to Fig. 1). Note that we unify the order processing and replenishment activities within a single $M/G/1$ queuing station, with orders causing reduction in inventory ($-n$ products) and procurement/replenishment causing increase in inventory ($+m$ products). Further, the hybrid model introduces a completion probability p, that may be tuned depending on warehouse deployments. A low value of p, resembles a closed queuing network (for instance, when orders and procurements are processed as batches). High value of p resembles an open queuing network (for instance, with first-in first-out transactional orders). An additional *think time* is incorporated into the model, that reflects the time spent in the system, without consuming resources.

While optimizing inventory levels to take care of varying demand has been well studied [5][11], we concentrate on replacing bottleneck queuing resources by superior queuing stations in Fig. 2:

(1) *Scale Out*: This involves parallelizing tasks to multiple resources. For instance, if one human agent is handling tasks, this would mean adding more of similar agents to meet increasing demand.
(2) *Scale Up*: This involves replacing a queuing station with a superior one (marked Ⓢ), such as a robotic picking agent with higher individual throughput.
(3) *Hybrid Scale*: A hybrid approach that parallelizes tasks and replaces a few of the agents with superior ones.

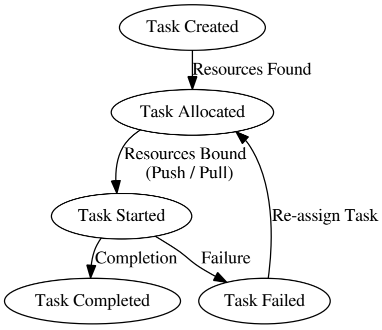


Figure 3: Workflow Task Creation, Allocation, Completion.

Such an approach for specifying scaling has been used in elastic cloud based infrastructures [14] to support auto-scaling features. The queuing stations are used for performance and operational analysis of the automation workflow resources.

3.2 Workflow Resource Patterns

Workflow resource patterns [9] have been proposed as an exhaustive list of patterns seen in task allocation in business processes [7]. These patterns are used for task creation, allocation, monitoring and completion involving multiple modular components, typically observed in workflow execution. Fig. 3 provides an overview of activities provided in resource modeling. Once a task is created (e.g. locate and pick a carton of cookies from warehouse), it may be allocated to one/many resources that may be humans or automation robotic agents. The task may be allocated (push) or may be bid (pull) by resource agents available. These tasks can be individually allocated to human participants or to robotic agents. Once the task is allocated to a robotic agent, it is able to analyze and identify sub-tasks to complete the goal (using a knowledge base). Non-completed tasks may be re-allocated to other resources. Such granular decomposition of workflow tasks using intelligent robotic agents has been studied in [15].

Multiple patterns have been proposed in [9] to formally model activities in Fig. 3. To map the queuing stations in Fig. 2 to workflow resource patterns, we provide the analysis in Table 1. The individual task creation, allocation, push/pull resources and detour patterns are mapped to our queuing stations. We comment on a few cases in Table 1:

(1) *Default* case: This is mapped to $M/G/1$ queuing stations in Fig. 3. Task creation/allocation patterns in Table 1 that are typically offered to a single resource are mapped to this case, for instance 1. Direct Allocation and 11. Automatic Execution. Execution and Detour patterns in Table 1 that are treated in a single queuing station are also mapped here such as 32. Suspension-Resumption and 39. Chained Execution.
(2) *Scale out* case: This is mapped to multiple parallel queuing stations in Fig. 3 and handles concurrent resources. Table 1 maps task creation patterns such as 5. Separation of Duties, push patterns such as 13. Distribution by Offer – Multiple Resources and execution/detour patterns such as 43. Additional Resources.
(3) *Scale up* case: This is mapped to heterogeneous queuing stations in Fig. 3, which can provide superior service demands.

Table 1: Workflow Resource Patterns [9] Mapped to Queuing Scale Up, Scale Out and Hybrid Scale Stations.

Queuing Station	Task Creation, Task Allocation Patterns	Push Patterns	Pull Patterns	Task Execution, Detour Patterns
Default (M/G/1)	1. Direct Allocation 11. Automatic Execution 40. Configurable Unallocated Work Item Visibility 41. Configurable Allocated Work Item Visibility			32. Suspension–Resumption 33. Skip 36. Commencement on Creation 38. Piled Execution 39. Chained Execution
Scale out	5. Separation of Duties 6. Case Handling	13. Distribution by Offer – Multiple Resources 16. Round Robin Allocation 18. Early Distribution 19. Distribution on Enablement 20. Late Distribution	24. System-Determined Work Queue Content	29. Deallocation 30. Stateful Reallocation 31. Stateless Reallocation 37. Commencement on Allocation 42. Simultaneous Execution 43. Additional Resources
Scale up	3. Deferred Allocation 7. Retain Familiar 8. Capability-based Allocation 9. History-based Allocation 10. Organizational Allocation	12. Distribution by Offer – Single Resource 14. Distribution by Allocation – Single Resource 15. Random Allocation	21. Resource-Initiated Allocation 22. Resource-Initiated Execution – Allocated Work Item 23. Resource-Initiated Execution – Offered Work Item	29. Deallocation 30. Stateful Reallocation 31. Stateless Reallocation 27. Delegation 28. Escalation 34. Redo 35. Pre-Do
Hybrid scale	2. Role-Based Allocation 4. Authorization	17. Shortest Queue	25. Resource-Determined Work Queue Content 26. Selection Autonomy	42. Simultaneous Execution 43. Additional Resources 27. Delegation 28. Escalation 34. Redo 35. Pre-Do

Table 1 maps task creation patterns such as 8. Capability-based Allocation, push patterns such as 12. Distribution by Offer – Single Resource, pull patterns such as 21. Resource-Initiated Allocation and execution/detour patterns such as 28. Escalation.

(4) *Hybrid Scale* case: This is mapped to a combination of multiple and heterogeneous resources in Fig. 3. Table 1 maps task creation patterns such as 2. Role-based Allocation, pull patterns such as 25. Resource-Determined Work Queue Content and execution/detour patterns such as 42. Simultaneous Execution.

Such mapping allows us to analyze the performance of complex workflows, using the abstraction of our hybrid queuing model. Performance analysis of automation workflows, using the queuing stations, are analyzed next.

4 PERFORMANCE ANALYSIS

In this section, we summarize a few of the performance laws that are of interest in our analysis. This is followed by Mean Value Analysis (MVA) that are developed for single class, multi class, load dependent, load independent and probabilistic completion rates.

4.1 Performance Laws

In order to perform operational analysis of our queuing models, we specify the notations in Table 2. We briefly review them here; an interested reader is referred to [10] for further details. Concepts such as throughput, service demand, latency and concurrency are incorporated into these metrics. Resources refer to human, robotic or software agents that are assigned tasks to be completed.

- **Utilization Law**: Utilization is the fraction of time the resource is busy.

$$\mathrm{U}_i = \mathrm{X}_i \cdot \mathrm{S}_i \tag{1}$$

Table 2: Notations for Performance Analysis.

Symbol	Notation
Q_i	Number of jobs in queuing station *i*
U_i	Utilization of queuing station *i*
X_i	Throughput of queuing station *i*
R_i	Response time of queuing station *i*
V_i	Average number of visits to queuing station *i*
S_i	Service demand of queuing station *i*
X	Throughput of the system
N	Average number of tasks in the queuing system
R	Average response time of the queuing system
Z	Mean think time of a task

- **Service Demand Law**: Total average service time spent at resource *i*, denoted S_i.

$$\mathrm{S}_i = \frac{\mathrm{U}_i}{\mathrm{X}} \tag{2}$$

- **Little's Law**: If there are N orders in the system, each with think times Z (time waiting between interactions with the system) and the system processes at the throughput rate X producing a wait time R, the following relationship applies:

$$\mathrm{N} = \mathrm{X} \cdot (\mathrm{R} + \mathrm{Z}) \tag{3}$$

We make use of the service demand law and Little's law in deriving service demands required in proceeding sections.

4.2 Mean Value Analysis

Mean value analysis (MVA) [10] has been applied with considerable success in the case of closed queuing networks in order to predict performance at higher work loads. We make use of mean value analysis models to analyze the performance of the queuing automation models in Fig. 2. The exact MVA algorithm [10] starts with an empty network; it then increases the number of customers by 1 at each iteration until there are the required number (N) of customers in the system. For each queuing station $k = 1, \ldots, K$, the waiting time R_k is computed using the static input service demands S_k and

Algorithm 1: Exact Mean Value Analysis (MVA) Algorithm, Single Class, Constant/Load Dependent Service Demand, Probabilistic Completion Rate.

```
   Input: Set of queuing stations $k \in K$; Corresponding Service demands $S_k$ and
          Visit counts $V_k$; Number of concurrent users N; Think time Z;
          Probability of task completion $p$
   Output: Throughput $X^n$ with increasing concurrency $n \in N$; Response time
          $R^n$ with increasing concurrency $n \in N$;
1  for $k \leftarrow 1$ to $K$ do
2      Initialize queue at each station: $Q_k \leftarrow 0$
3      Initialize utilization at each station: $U_k \leftarrow 0$
4  for $n \leftarrow 1$ to $N$ do
5      for $k \leftarrow 1$ to $K$ do
6          Response time at each station:
           $R_k = \begin{cases} S_k \cdot (1 + Q_k), & \text{Load Independent Case} \\ S_k \cdot f(U_k^n) \cdot (1 + Q_k), & \text{Load Dependent Case} \end{cases}$
7      Total response times using visit counts: $R^n = \sum_{k=1}^{K} V_k \cdot R_k$
8      Throughput with Little's Law: $X^n = \frac{n}{R^n + Z}$
9      for $k \leftarrow 1$ to $K$ do
10         Update queues at each station: $Q_k = (1 - p) \cdot X^n \cdot V_k \cdot R_k$
11         Update utilization at each station: $U_k^n = \frac{Q_k}{1+Q_k}$
12 return $X^n, R^n, U_k^n$
```

the number of jobs in the queue Q_k. The system throughput is then computed using the sum of waiting times at each node and Little's law (eq. 3). Finally, Little's law is applied to each queue to compute the updated mean queue lengths.

Algorithm 1 provides an outline of Mean Value Analysis applied to single class models. This may be mapped to a single category of products handled by Industry 4.0 warehouses. Note that we calculate the response time using two models (Line 6 in Algorithm 1):

$$R_k = \begin{cases} S_k \cdot (1 + Q_k), & \text{Load Independent Case} \\ S_k \cdot f(U_k^n) \cdot (1 + Q_k), & \text{Load Dependent Case} \end{cases} \quad (4)$$

where, the load independent case has constant service demand S_k, while the load dependent case has service demands that vary as a function of utilization at each concurrent load level $S_k \cdot f(U_k^n)$. Load dependent service demands are particularly realistic when human agents are involved, with superior service times seen with greater demand loads [16]. We further introduce the probability of completion p, which is used to append queue lengths during each iteration (Line 10 in Algorithm 1):

$$Q_k = (1 - p) \cdot X^n \cdot V_k \cdot R_k \quad (5)$$

This allows us to simulate both closed (low p) and partially open (high p) queuing models (as in Fig. 2).

Algorithm 2 provides the Multi-Class MVA model, which makes use of c classes of orders. This is crucial when there are multiple types of orders having different rates and guarantee, while making use of shared resources. The queue length at each station is a combination of all the flows that are served (Line 12 in Algorithm 2):

$$Q_k = (1 - p) \cdot \sum_{c=1}^{C} X_c^n \cdot V_{c,k} \cdot R_{c,k} \quad (6)$$

Algorithm 2: Exact Mean Value Analysis (MVA) Algorithm, Multi Class, Constant/Load Dependent Service Demand, Probabilistic Completion Rate.

```
   Input: Set of queuing stations $k \in K$; Corresponding Service demands $S_k$ and
          Visit counts $V_k$; Number of concurrent users N; Think time Z; Number
          of classes $C$ with population of each class $n_1, n_2, \dots, n_C$, Probability of
          task completion $p$;
   Output: Throughput $X^n$ with increasing concurrency $n \in N$; Response time
          $R^n$ with increasing concurrency $n \in N$;
1  for $k \leftarrow 1$ to $K$ do
2      Initialize queue at each station: $Q_k \leftarrow 0$
3      Initialize utilization at each station: $U_k \leftarrow 0$
4  for $n \leftarrow 1$ to $\sum_{c=0}^{C} N_c$ do
5      for $c \leftarrow 1$ to $C$ do
6          for $k \leftarrow 1$ to $K$ do
7              Response time at each station:
               $R_{c,k} = \begin{cases} S_{c,k} \cdot (1 + Q_k), & \text{Load Independent Case} \\ S_{c,k} \cdot f(U_k^n) \cdot (1 + Q_k), & \text{Load Dependent Case} \end{cases}$
8      for $c \leftarrow 1$ to $C$ do
9          Total response times using visit counts: $R_c^n = \sum_{k=1}^{K} V_{c,k} \cdot R_{c,k}$
10         Throughput with Little's Law: $X_c^n = \frac{n_c}{R_c^n + Z_c}$
11     for $k \leftarrow 1$ to $K$ do
12         Update queues at each station: $Q_k = (1 - p) \cdot \sum_{c=1}^{C} X_c^n \cdot V_{c,k} \cdot R_{c,k}$
13         Update utilization at each station: $U_k^n = \frac{Q_k}{1+Q_k}$
14 return $X_c^n, R_c^n, U_k^n$
```

We analyze the performance of these systems using simulations in the next section. We emphasize that these models combine both traditional business process workflows with queuing network analysis and industrial robotic automation.

5 SIMULATION RESULTS

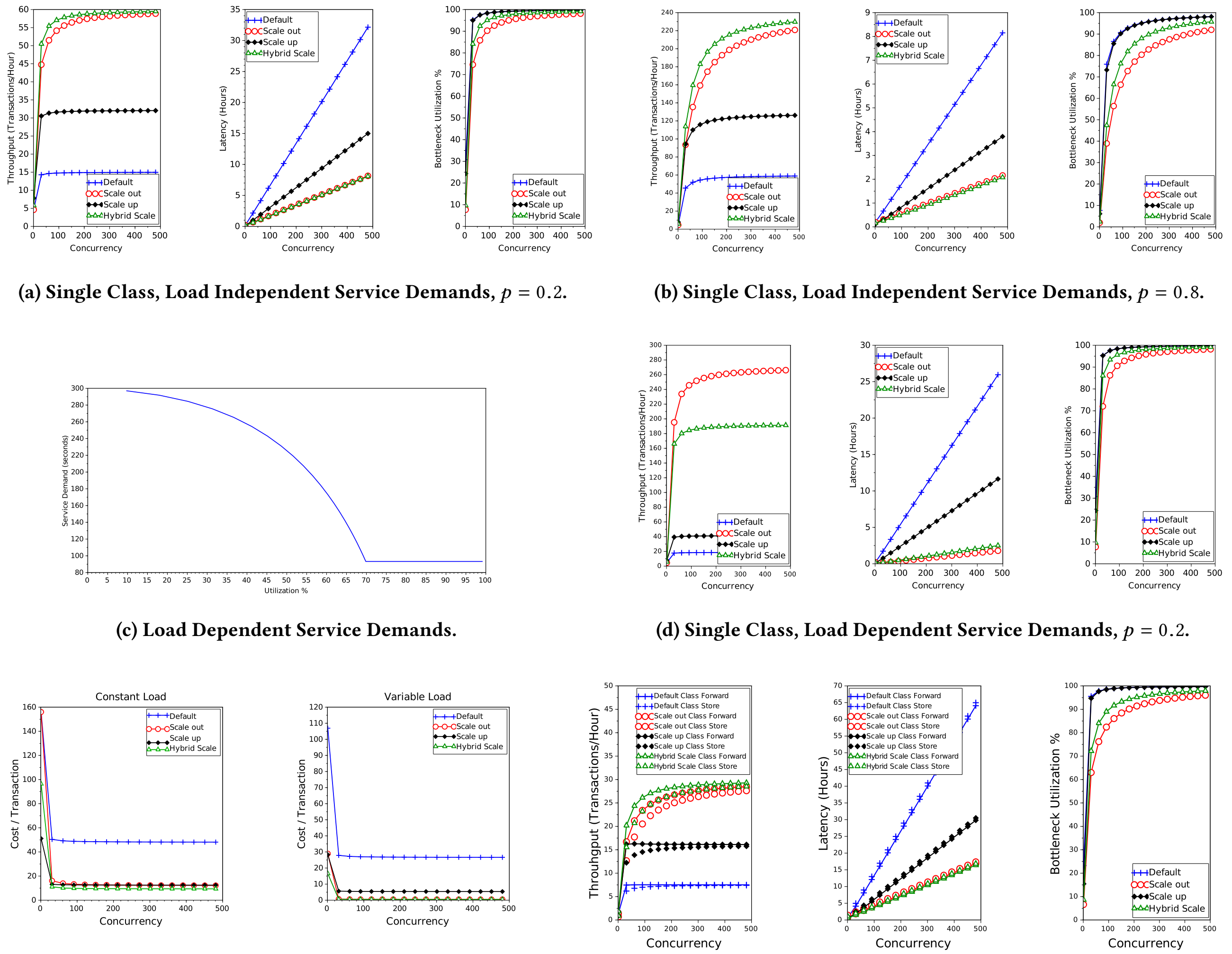
In order to study the performance improvements provided by automation in warehouses, we employ simulation settings taken from realistic datasets. Based on the experimental data provided in [5], the following are typical time-frames in warehouse logistic operations: Travel (55%), Item Search (15%), Item Extraction (10%) and Additional Overheads (20%). Applying the experimental results in [8], which provides 35 seconds as the mean time for a picking robot search and extract feature, the mean time for for the end-to-end automated pickup process is set at 140 seconds. This is an improvement over the mean time of 300 seconds taken by human agents for warehouse procurement [6]. Using these settings, we simulate the hybrid queuing model in Fig. 2 with various queuing stations. These settings are used to analyze performance characteristics under various environments.

Table 3 provides the service demands that are used in our simulation setup, with the *Default* Order picking/put-away times being set at 300 seconds. As specified in Fig. 2, Table 3 sets the *Scale Up* service demand value to 140 seconds (representing automated pickers), *Scale Out* with four (human) agents to 75 seconds and the *Hybrid Scale* to a combination of humans and robotic entities. We use these settings to simulate the MVA Algorithms provided in Algorithms 1 and 2 in Scilab[3]. We simulate various scenarios that are typically seen in warehouse automation deployments.

[3] https://www.scilab.org/

Table 3: Mean Service Demands for Warehouse Activities.

Queuing Station	Default	Scale Out	Scale Up	Hybrid Scale
Order Processing / Replenishment	60 sec.	60 sec.	60 sec.	60 sec.
Put-away / Order Picking	300 sec.	75 sec. (4 agents)	140 sec.	150 sec. (2 agents), 70 sec. (2 agents)
Order Checking / Fulfillment	60 sec.	60 sec.	60 sec.	60 sec.
Think Time	60 sec.	60 sec.	60 sec.	60 sec.



(a) Single Class, Load Independent Service Demands, $p = 0.2$.

(b) Single Class, Load Independent Service Demands, $p = 0.8$.

(c) Load Dependent Service Demands.

(d) Single Class, Load Dependent Service Demands, $p = 0.2$.

(e) Cost per Transaction for Load Dependent/Independent Service Demands.

(f) Multi Class, Load Independent Service Demands, $p = 0.2$.

Figure 4: Performance Analysis of Warehouse Automation.

5.1 Single Class, Load Independent Service Demands

Algorithm 1 is simulated for a single class of items, with load independent service demands. Fig. 4a provides the throughput, response time and bottleneck utilization for this network. We use $p = 0.2$, which represents a closed queuing model (orders primarily processed in batches). We notice that while the throughput with a human agent operating order put-away/picking is 15 transactions/hour, this doubles with the scale up pattern (replacing with a robot automation framework). The best improvements are seen four concurrent operators (scale-out), providing peak throughput of 58 transactions/hour. While the default configuration can support around 100 concurrent users, the scale up/hybrid scale patterns can handle greater than 200 concurrent users.

The set-up is modified to a setting with $p = 0.8$ in Fig. 4b, representing a first-in first-out transactional process. We notice that the throughput and concurrency levels supported in Fig. 4b are superior to that of Fig. 4a. Superior improvements are seen with hybrid scale and scale out stations.

5.2 Single Class, Load Dependent Service Demands

In conventional computing systems, the service demands are not constant, but rather, a function of the workload [16]. This may also be practically seen in systems involving human agents, wherein, higher rates of work output are noticed with increased workloads. We model the function of service demands using Fig. 4c, with the 300 seconds service demands tapering to 100 seconds, with higher concurrent workloads. We must note here that sustained production at lower service demands may not be feasible with humans agents, due to increased stress/fatigue levels. Robotic automation agents, however, should be able to handle the variations in service demands for prolonged periods.

Performance improvements with various resource patterns is seen in Fig. 4d, with the scale out pattern providing the highest improvements. There are considerable deviations in the performance measures seen in Fig. 4a and 4b. So, dependent on the deployment scenario, it will be crucial to select the best pattern for optimal performance improvement.

Comparing the cost per transaction in Fig. 4e, (cost being inversely proportional to service demands), we notice that variable service demands outperform the traditional constant demand performance. This suggests that load dependent service demand governors may be more advantages in the case of autonomous robotic agents. We acknowledge that capital costs employed in setting up the automation system has not been considered in this case.

5.3 Multi Class, Load Independent Service Demands

To simulate cases where multiple order "types" are processed by the warehouse stations, we introduce *forward* orders (processed immediately) and *storage* orders (processed in a delayed manner) as also shown in Fig. 1. The mean service demands for storage orders are set to double the values reported in Table 3. We notice that in such multi-class models as well, there are improvements provided by scale out and hybrid scale patterns (Fig. 4f). While there may be delays in think times that may be integrated into these models, this demonstrates that multi-class orders and suppliers may be studied in this framework. Due to multi-class interactions, the maximum throughput is limited (half of what is seen in Fig. 4a), which consequently affects the response time.

5.4 Structured Resource Pattern Analysis

Through our simulations, we can broadly classify the improvements provided through various resource patterns. Table 4 summarizes the improvements provided by various resource patterns, per unit increase in automation. We see that for single class models with constant loads, throughput may be improved between 100 − 200% and latency ∼ 60% while still reducing cost per transaction by ∼ 70%. This improvement is higher in the case of load dependent, specially with the *Scale Out* pattern providing significant improvements in throughput and cost per transaction. These can be linked to a central controller that can autoscale and modify patterns, such as those seem in elastic cloud deployments [14]. Rules for runtime adaptation techniques, for instance based on monitored concurrency levels, are given in the pseudo-code below (refers to the outputs provided in Fig. 4b):

```
Input Concurrency Level
Case based on Concurrency Level
  Concurrency Level >= 100:  Automation Resource Scale Up
  Concurrency Level >= 200:  Automation Resource Scale Out
  Concurrency Level >= 350:  Automation Resource Hybrid Scale
  Default:                   Automation Resource Default
End Case
```

Such rules are a starting point to aid in automated runtime adaptation in Industry 4.0 deployments.

Mapping this back to Table 1, an efficient set of patterns to consider in the *Scale Out* case would be: 5. Separation of Duties (Task allocation) → 13. Distribution by Offer – Multiple Resources (Push Pattern) → 24. System-Determined Work Queue Content (Execution Pattern) → 31. Stateless Reallocation (Detour Pattern). Similarly, for the *Hybrid Scale* case, the patterns to consider are: 2. Role-Based Allocation (Task allocation) → 25. Resource-Determined Work Queue Content (Pull Pattern) → 42. Simultaneous Execution (Execution Pattern) → 43. Additional Resources (Detour Pattern).

In **summary**, our work demonstrates the following:

(1) Systematic analysis of workflow resource models using mapping between workflow patterns and queuing stations (Table 1 and Fig. 2).
(2) Estimating performance at higher concurrency loads with multiple classes, load dependent service demands and probabilistic completion rates using MVA algorithms (Algorithms 1 and 2).
(3) Accurate characterization of automation resource patterns in Industry 4.0 deployments (Fig. 4).
(4) Identifying appropriate workflow resource patterns, that can improve order delivery throughput, latency and cost per transaction (Table 4).

Such accurate modeling of automation workflows with performance analysis will prove crucial in the case of multiple industrial deployments.

6 RELATED WORK

Industry 4.0 [3] requires increased automation, autonomy and adaptation among distributed entities working in factory/warehousing environments. A central entity for control and coordination in warehouses has traditionally been the Warehouse Management System (WMS) [5]. However, with increased warehouse automation such as those demonstrated with Amazon's Kiva robots [6], processes with heterogeneous participants and control flow are in vogue. Robot automation employed in Amazon's warehouses has drastically reduced procurement times from 60 minutes taken by human participants to around 15 minutes. Automation has increased the inventory capacity/square foot by 50% and reduced operating cost by 20% ($225 million/warehouse) [6].

Models for warehouse activities have been typically been focused on inventory management, load balancing and supply chain optimization [11]. While traditional warehouses only require orchestration of business processes [7], automated warehouses include intelligent robotic agents and IoT devices [4], requiring accurate workflow models. In industrial environments where software and (mobile) hardware components have tight interactions, such workflow specifications would involve intricate flow control and concurrency issues. Petri net models of factory workflows are presented in [17], using which properties such as liveness and deadlock freeness are synthesized. In [18], an integrated database is proposed to analyze performance indicators in warehousing activities.

Table 4: Structured Performance Improvements with Workflow Resource Patterns.

Scenario	Resource Pattern	Max. Throughput	Max. Concurrency	Bottleneck Latency	Cost per Transaction
Single Class, Constant Service Demand, low *p*	Default (1 human)	15 trans/hour	74	5 hours	50 units
	Scale out (2 humans)	+140%	+117%	−40%	−40%
	Scale up (1 robot)	+113%	+8%	−60%	−70%
	Hybrid Scale (1 human, 1 robot)	+148%	+134%	−40%	−40%
Single Class, Constant Service Demand, high *p*	Default (1 human)	55 trans/hour	140	2.3 hours	14 units
	Scale out (2 humans)	+150%	+130%	−37%	−38%
	Scale up (1 robot)	+122%	+25%	−57%	−78%
	Hybrid Scale (1 human, 1 robot)	+159%	+130%	−37%	−45%
Single Class, Load Dependent Service Demand, low *p*	Default (1 human)	20 trans/hour	60	3 hours	28 units
	Scale out (2 humans)	+500%	+170%	−44%	−47%
	Scale up (1 robot)	+100%	+16%	−67%	−82%
	Hybrid Scale (1 human, 1 robot)	+400%	+100%	−42%	−47%
Multi Class, Constant Service Demand, low *p*	Default (1 human)	7.5 trans/hour	120	16 hours	100 units
	Scale out (2 humans)	+130%	+98%	−35%	−35%
	Scale up (1 robot)	+127%	+21%	−53%	−70%
	Hybrid Scale (1 human, 1 robot)	+142%	+112%	−35%	−40%

A structured process of describing workflows and business processes [7] has brought in formalisms such as workflow patterns [9] and workflow nets [19]. To analyze performance of these models, Petri-net based approaches [19] and data flow approaches [20], have been proposed. Verification and timing analysis of such workflows in the Industry 4.0 context has also been studied in [21]. Hierarchical decomposition of workflow tasks to robotic agents has been analyzed in [15].

Performance analysis of software systems is a well studied area, with queuing network models typically employed [10]. Mean value analysis (MVA) has been proposed as a recursive technique to estimate performance with incremental increase of loads in closed queuing networks. In [13], open vs. closed queuing networks are studied and compared in various scenarios. Auto-scaling features, which are important in the context of elastic cloud deployments, are studied in [14].

In this work, we model high level automation workflows using workflow patterns. Improvements in performance are studied using queuing network models and MVA. We evaluate improvements provided by various scaling up/out patterns, that are typically employed in automation framework.

7 CONCLUSIONS

Automation resources are being increasingly employed in the manufacturing, retail and logistics industries to help improve end-to-end performance efficiency. In order to integrate these resources into traditional business processes, a framework for performance modeling and analysis of automation improvements is needed. In this work, we couple the modeling approaches of workflow patterns with queuing network analysis to measure automation performance. Through the use of Mean Value Analysis (MVA) algorithms, we analyze Industry 4.0 warehouse automation workflows for scenarios involving single-class, multi-class, batch, transactional and load dependent service demands. The analysis shows that superior improvements may be gained in each case through judicious selection of appropriate workflow resource patterns. The structured patterns for performance improvement also allows for runtime adaptation to satisfy varying demands.

In future, we would like to deploy these models in domains such as factory automation and logistics to predict performance and provide accurate reconfiguration patterns.

REFERENCES

[1] P. Leita, A. W. Colombo & S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges", *Computers in Industry*, vol. 81, pp. 11–25, 2016.
[2] S. Greengard, "The Internet of Things", *MIT*, 2015.
[3] M. Hermann, T. Pentek & B. Otto, "Design Principles for Industrie 4.0 Scenarios", *49th Hawaii Intl. Conf. on System Sciences*, 2016.
[4] S. Russell & P. Norvig, "Artificial Intelligence: A Modern Approach ", *Pearson*, 3rd Ed., 2009.
[5] J. Bartholdi & S. Hackman, "Warehouse and Distribution Science", *The Supply Chain and Logistics Institute*, 2016.
[6] P. Wurman, R. D'Andrea & M. Mountz, "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses", *AAAI Artificial Intelligence Mag.*, vol. 29, no. 1, pp. 9–19, 2008.
[7] M, Weske, "Business Process Management: Concepts, Languages, Architectures", *Springer-Verlag Berlin Heidelberg*, 2nd ed., 2012.
[8] Hao Zhang et al., "DoraPicker: An autonomous picking system for general objects", *IEEE Intl. Conf. on Automation Science and Engineering (CASE)*, pp. 721–726, 2016.
[9] N. Russell, A.H.M. ter Hofstede, D. Edmond & W.M.P. van der Aalst, "Workflow Resource Patterns", *BETA Working Paper Series – Eindhoven University of Technology*, WP 127, 2004.
[10] E. Lazowska, J. Zahorjan, S. Graham & K. Sevcik, "Quantitative System Performance: Computer System Analysis Using Queuing Network Models", *Prentice-Hall, Inc.*, 1984.
[11] R. Ganeshan, "Managing supply chain inventories: A multiple retailer, one warehouse, multiple supplier model", *Int. J. Production Economics*, vol. 59, pp. 341–354, 1999.
[12] M. K. Govil & M. C. Fu, "Queuing theory in manufacturing: A survey", *J. of Manufacturing Sys.*, vol. 18, no. 3, pp. 214–240, 1999.
[13] B. Schroeder, A. Wierman & M. Harchol-Balter, "Open Versus Closed: A Cautionary Tale", *USENIX NSDI Tech. Paper*, 2006.
[14] T. Lorido-Botran, J. Miguel-Alonso & J. Lozano, "A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments", vol. 12, no. 4, pp. 559–592, 2014.
[15] A. Kattepur, S. Dey & P. Balamuralidhar, "Knowledge Based Hierarchical Decomposition of Industry 4.0 Robotic Automation Tasks", *IEEE Intl. Conf. on Industrial Electronics*, 2018.
[16] A. Kattepur & M. Nambiar, "Performance Modeling of Multi-tiered Web Applications with Varying Service Demands", *IPDPS Workshops*, 2015.
[17] F. Basile, P. Chiacchio & J. Coppola, "A hybrid model of complex automated warehouse systems – Part I: Modeling and simulation", *IEEE Trans. on Automation Science and Engineering*, vol. 9, no. 4, 2012.
[18] J. C. Hernandez-Matias, A. Vizan, J. Perez-Garcia & J. Rios, "An integrated modelling framework to support manufacturing system diagnosis for continuous improvement", *Robotics and Computer-Integrated Manufacturing*, vol. 24, pp. 187–199, 2008.
[19] W.M.P. van der Aalst, "Verification of Workflow Nets", *Intl. Conf. on Application and Theory of Petri Nets*, 1997.
[20] M. Kovacs and L. Gonczy, "Simulation and Formal Analysis of Workflow Models", *Electronic Notes in Theoretical Computer Science*, vol. 211, pp. 221–230, 2008.
[21] A. Kattepur, A. Mukherjee & P. Balamuralidhar, "Verification and Timing Analysis in Industry 4.0 Warehouse Automation Worklows", *IEEE Intl. Conf. on Emerging Technologies and Factory Automation*, 2018.