FORTNA

THE DISTRIBUTION EXPERTS™
Fortna Thought Leadership Series

Push vs. Pull: How Pull-driven Order Fulfillment Enables Greater Flexibility in Distribution Operations

The demands of e-commerce and omni-channel fulfillment are increasing the need for innovative distribution operations that can adapt swiftly to changing demand patterns, including seasonal and daily volume peaks. Operations with a large variety of SKUs and/or high and fluctuating order volume (usually consisting of few lines requiring more each picking) are often subject to customer expectations for high service levels (short delivery times and free shipping). Additionally, maintaining high equipment utilization and high worker productivity represent dueling objectives.

To face these challenges, distribution center managers need solutions that can maximize productivity, speed and service levels. Pull-driven order fulfillment coupled with Warehouse Execution Systems (WES) offer a promising solution to these order fulfillment challenges with real-time continuous monitoring of order and resource (labor and equipment) status. With this level of visibility, algorithms that dynamically optimize decisions governing the movement and handling of orders are built into the WES. The WES provides the flexibility to maintain continuous flow in the DC despite peaks and valleys in order volumes and changing order profiles. Pull-driven order fulfillment can provide additional flexibility by meeting the same throughput requirements achieved by the push-driven wave-based order fulfillment, but with higher speed, higher service levels, less labor, more level resource utilization and smaller hardware investment cost. And pull-driven order fulfillment can also provide higher throughput when hardware and/or labor are held constant.

In this article, we will compare and demonstrate the measurable performance differences between pull vs. push control for a zone-picking operation over parallel zones followed by a manual sortation operation to separate and consolidate the orders before packing.

PUSH: THE PROBLEM WITH WAVES

Many distribution centers today utilize wave-based order processing where lots of orders are simultaneously released to the floor in a batch or wave. At each operation, all resources work on that wave until it is completed and passed to the next, downstream operation. Unexpected events routinely impact the efficiency of wave-based processing. From personnel working at a slower pace than anticipated, to equipment breakage and stockouts — every event has a negative impact on the efficiency of wave-based

processes. At the beginning and middle of waves, the operation is very efficient. However, once the tail of the wave is reached, there is a severe drop in productivity as one wave closes out the last, few remaining orders and the next wave starts to trickle in.

Wave-based fulfillment can be effective, but can also be problematic for DCs with multiple automated systems keeping all the zones in sync—and workloads balanced. Wave-based processing can lead to inefficiencies and throughput issues as worker and equipment utilization drops off at the tail of a wave. One of the challenges of push fulfillment is the rigidity of the wave management system that often dictates a fixed batch size that must be processed before the next wave of orders is released. As a result, operation managers usually push multiple waves, which leads to accumulation at bottleneck resources and poor synchronization at the sortation operation. As a result, large and expensive buffers (either over-sized sortation operations or wave banks installed before the sortation operation) are often added to wave-based designs to reduce idle times of pickers waiting for orders at the tail end of a wave.

“Wave-based processing can lead to inefficiencies and throughput issues as worker and equipment utilization drops off at the tail of a wave.”

PULL: DYNAMICALLY ADJUSTING TO REAL-TIME CONDITIONS

Pull-driven fulfillment optimizes resources (labor and equipment) by dynamically controlling tasks to handle unexpected events because it monitors the real-time conditions of the system. Rather than assigning work in a single wave, orders are dynamically released in real-time as a threshold is reached. Using a revolving batch instead of a wave batch allows pickers to work continuously regardless of unexpected events. Pull-driven order fulfillment allows work to be dynamically re-assigned based on availability of resources. If one zone falls behind, workers can be dynamically re-assigned to help clear the bottleneck. This dynamic re-assignment increases overall labor productivity by not having resources “idle” while waiting for upstream processes to be completed.

An order fulfillment engine using the “pull” paradigm enabled with WES capability to provide continuous visibility on the status of resources and units of flow can achieve the same or higher throughput requirements as the push-driven wave-based order fulfillment, but with higher speed, higher service levels, less labor, more level resource utilization, and lower capital expenditures on material handling equipment.

COMPARISON STUDY: PUSH VS. PULL

We compared push vs. pull-driven order fulfillment solutions for a zone-picking operation over parallel zones followed by a manual sortation operation (i.e., a put wall) to separate and consolidate the orders before packing and/or value-added services. The pull-driven solution showed a significant increase in fulfillment speed and smoother flow (leading to steadier utilization of resources).

There are two main processes in this solution, conceptually illustrated in Figure 1:

1. **Zone-picking system over parallel zones:** Multiple orders are chosen for a wave. The work content (i.e., lines) for those orders are split over multiple picking zones, forming parallel batches. Each batch is picked in its zone and a tote containing multiple lines is delivered to a put wall downstream where it is combined with totes from the other picking zones. Considerable productivity gains are achieved by batch picking because a pick location is visited once during a wave—reducing the amount of travel by pickers. However, order integrity is not preserved during picking and requires a second process for sorting container contents to consolidate order units.
2. **Put wall sortation system:** A put wall is a set of openings or compartments (referred to as cubbies). One side of the put wall is staffed by an operator who puts units that belong to a specific order into an assigned cubby. The other side of the put wall is staffed by operator(s) who packs a complete order or sends to a downstream value-added services operation. The number and size of cubbies depends on order cubic size and the number of orders that needs to be processed simultaneously. Most often, there are multiple put walls in a DC. Figure 2 shows a picture of a put wall used for a direct-to-consumer order consolidation process.

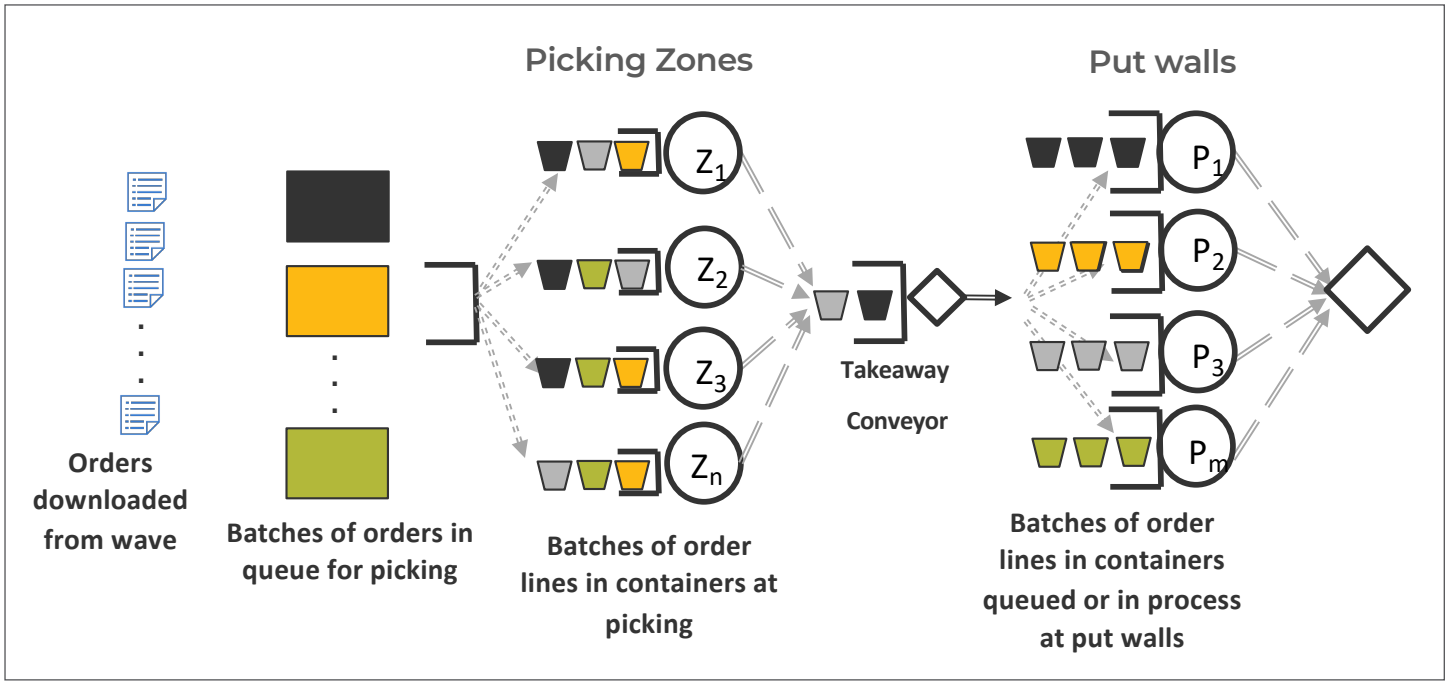

Fig. 1. Zone Picking to Put Walls Processes. Orders in a wave are batched to improve picking efficiency. Batches are released to pick zones for picking. Containers carrying the items belonging to the same batch are sent to a specific put wall. Items belonging to the same order are put into a specific put wall cubby. When the order items have been consolidated, the order is packed out and the cubby is now available for another order.

Fig. 2. A put wall. Each cubby is dedicated to an order. Totes arrive carrying mixed SKUs. When an operator scans an item, the light under the cubby lights up where the item needs to be put.

PULL FRAMEWORK

The pull framework is illustrated in Figure 3, where we have a virtual queue of orders in a wave. When a wave is released, an algorithm is used to group the orders into virtual batches in each zone. Some of these batches are available to pickers and some are released when the pull signal is triggered. An available picker is assigned the next batch(es) to pick. A batch now is assigned to a container and the picker executes the pick. Upon completion, the container is sent to the put walls. Units from the containers are put into the order cubby. When all the lines of an order are assembled at the put wall, the packer empties the cubby, and the order exits the area.

Pull parameters: The pull-driven flow algorithm determines the values of three parameters every time a wave is dropped into the system. These parameters are:

- X: is the number of orders in a batch; orders in a buffer are released to the same put wall, and therefore, picked into the same container. Tradeoff: As the number of orders in a batch increases, picking and putting productivity increase, but the cycle time of picking and putting also increase. The cycle time increase will increase variability, which will requires additional accumulation to mitigate.
- Y: Number of put wall batches to be picked simultaneously (assuming a picker can accommodate more than one container on the cart). Tradeoff: As the number of put wall batches increases, picking productivity increases, but picking cycle time increases.
- W: is the number of batches to maintain in the system (the released pool of batches, batches in-picking, batches in transfer to put walls, and at put walls). Tradeoff: As the number of batches increases, throughput increases, but cycle time also increases. As the number of batches increases, the system starts to approach a push system. As the number of batches decreases, throughput is degraded.

Although we are focusing on the pull algorithm here, clearly the underlying design has the greatest impact on the performance of the operation and must be optimized appropriately.

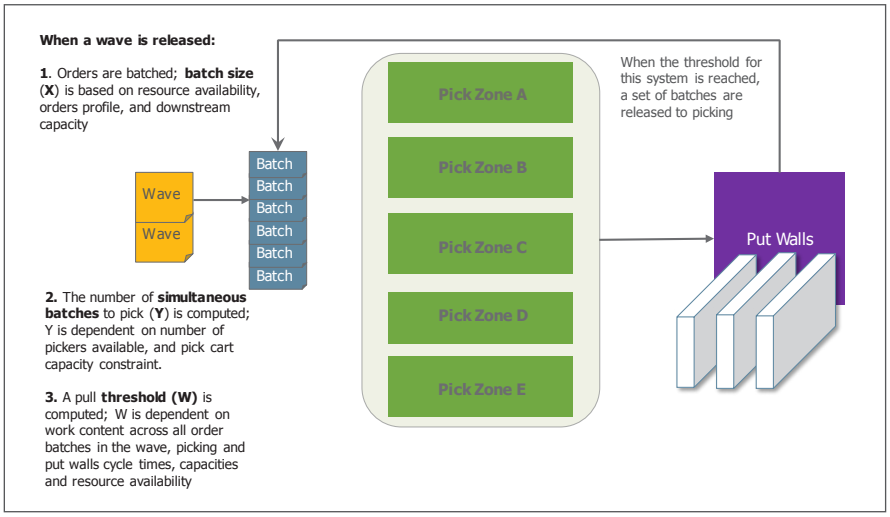

Fig. 3. Pull Framework for Batch Picking to Put Walls

Algorithm: The exact formulas are proprietary, but the general steps for dynamically setting these three parameters are outlined below. But the algorithm alone is not enough. It is the use, timing and complex variations of these sophisticated algorithms used in combination of the right design that allows pull-driven to deliver optimal results.

For a given setting characterized by:

- Pool of orders in the wave and their associated parameters including number of lines, number of units
- Available number of pickers
- Number of pick zones
- Picking process work content characterized by layout, process delays
- Available put walls
- Cubbies per put wall
- Putting process work content characterized by put wall size, process delays

1. Determine minimum X and Y that meet throughput requirement within capacity constraints (i.e., the current number of workers available).
2. Determine W that maximizes system throughput (i.e., number of orders for a given period of time). In an unconstrained sense, this parameter is influenced by the mean and variance of picking cycle time, putting cycle time, and transportation times from picking zones to individual put walls. Constraints like shipping windows and available dock doors as well as order priorities also have an impact on W.

The algorithm is run across every wave to ensure that real-time information is incorporated to optimize performance over the course of the day. The algorithm, although simply stated, is controlling a dynamic system with many interacting complexities based on the timing over several areas of the DC and has impacts on labor, equipment utilization, and service level. The algorithm relies on feedback from within the system to adjust the parameters. In this way the algorithm has been extensively tested and measured against actual performance.

BATCH PICKING TO PUT WALL PROCESS

When orders are released to picking, batches are formed by SKU and units are picked from multiple zones in the forward pick area into multiple containers. Containers from multiple zones are delivered to the put wall. The put wall operator starts removing units from the containers, scanning them, and putting them into the destination cubby until the container is empty. SKUs shared by multiple orders assigned to the same put wall within a wave are picked together into the same container then separated at the put wall. A Warehouse Execution System (WES) directs the operator to the correct cubby using put-to-light (typically). Once all the required units for an order have been put into the designated cubby, the order is ready to be packed by the operator on the other side of the put wall.

Put Wall Advantages: Order and SKU profiles and the picking methodology dictate whether there is a business case for installing put walls in a DC. Generally, put walls are beneficial for environments dominated with e-commerce multi-line orders and order volumes that are higher than what is feasible for a discrete or cluster picking operation. Put walls require lower capital investment compared to conveyor-based unit sorters, which are used for higher order volume and/or high SKU commonality over orders (more typical in wholesale and retail replenishment). When coupled with an intelligent WES, putting productivity and order sortation accuracy can be enhanced substantially.

Put Wall Drawbacks: The main drawbacks that we have observed in systems with a business case for put walls are operational challenges that manifest themselves in long and highly variable dwell times of orders in cubbies, erratic resource utilization patterns, and long queues of containers at the put walls. The underlying causes for these effects are the result of difficulty in synchronizing the arrival times of units for the same order. When the lines of an order are picked from different zones, they arrive to the put wall in longer dwell time of the order in its cubby. The long dwell times keep the cubbies from being turned and reused for other orders, which might create a queue of containers in front of the put wall, or a delay in the release of the next wave's orders, which leads to idle resources when there is work to be done.

Oversizing the Put Walls: System designers typically address this drawback by oversizing the put walls to create a buffer. Larger put walls result in more travel from container to cubby and so forth, which reduces the

putting productivity, especially when the WES or WCS lack the intelligence to assign cubbies to orders in a way that minimizes operator travel. The larger put walls might solve the container accumulation problem, but does not address the long order cycle times and service level implications.

Installing a Wave Bank: Another design solution to address the synchronization problem is adding a wave bank, which is essentially a central buffer to which containers are pushed after being picked. Release of containers from the wave bank improves the speed of order consolidation. The downside is the additional investment in the hardware and space required for the wave bank. Figure 4 shows a picture of a wave bank.

"The right design coupled with the right systems support can retain the advantages while mitigating or minimizing the drawbacks."

The right design coupled with the right systems support can retain the advantages while mitigating or minimizing the drawbacks.

Fig. 4. Wave Bank. Totes are waiting at the wave bank until ready to be released to the put walls for sortation.

DESIGN CASE STUDY

We were asked to design a distribution center for a major US retailer and determined that a put wall operation was the correct design for fulfilling multi-unit orders for their e-commerce orders. We evaluated the operation under a push- and pull-driven environment to illustrate the value provided by WES. Cyber Monday is their peak day when they see about 40K multi-unit orders with an average of 4.3 lines/order and 1.2 units/line. SKUs are organized in four pick modules with three levels each, which results in 12 pick zones. The default push approach was to use a batch size of 108 orders per wave and have 50 put walls with about 288 openings (cubbies) per put wall (i.e., the assumed approach for accommodating the variation was to oversize the put walls by 167%).

We applied the algorithm to the operating parameters described above with its assumed setting for headcount and productivity rates in a static environment. The output of the pull heuristic is a batch size of 60 orders on peak day (compared to 108 used for the baseline push system) and put walls with 100 cubbies per wall (compared to 288 in the baseline system). The number of batches to maintain in the system (pull threshold) is 75 batches.

Note that from the design side the pull system has several advantages. Smaller put walls (65% smaller), which not only reduce the capital investment tremendously, will have higher productivity as put wall operators travel shorter distances along the wall. However, the smaller batch size will negatively impact picking productivity. Table I provides a number of comparison points and indicates that the productivity pickup at the put walls (43% improvement) outweighs the negative impact on picking productivity (4% reduction). Overall, even ignoring the impact of wave tails (which is greater in the push scenario), there is over a 5% reduction in workers (14 workers over two shifts) using the pull system.

TABLE I: PUSH VS. PULL PEAK DAY SETTINGS

	PUSH	PULL
Total throughput required	2,000 orders/hour; 8,600 lines/hour; 10,062 units/hour	
Number of pick zones	12 zones: FOUR pick modules with THREE levels each	
Orders/Batch (X)	108	60
Batches/Pick Cycle (Y)	10	6
Picker productivity	100 Lines/Hour	96 Lines/Hour
Pickers	108 (9 pickers/module level)	
Number of Put Walls	50	
Size of the put walls	288 cubbies/PW	100 cubbies/PW
Putting productivity	420 units/hour	600 units/hour
Release Rule (W)	Release 1000 orders every 30 minutes	Release a batch of orders when the number of batches in the system goes below 75

Note that we chose to hold the total throughput required constant for the two systems and to measure the impact of changes to the time to pick and sort all orders and order cycle time and variation in addition to the investment in the put walls and worker productivity. Other examples can be constructed where the number of workers is held constant and potential differences in throughput-are measured. The move to a pull environment provides flexibility over the push environment. Table II summarizes the performance output from the simulation model used to compare the two approaches.

TABLE II: PUSH VS. PULL PEAK DAY PERFORMANCE

	PUSH	PULL
Hours to pick and sort all orders	24.4	21.7
Average Order Cycle Time (Minutes)	267	103
Order Cycle Time Standard Deviation (Minutes)	19.8	11.2

Figure 5 contrasts the order-by-order cycle time for push vs. pull, and Figure 6 shows the number of lines in process at the put walls throughout the simulated day.

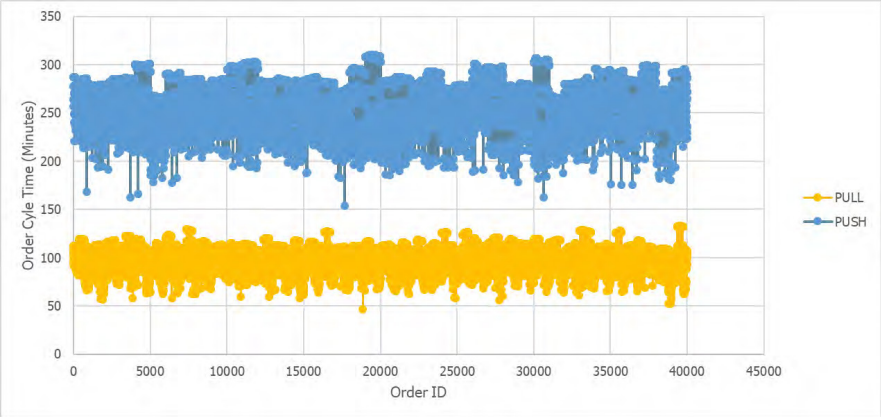

Fig. 5. Order-by-order cycle time in push vs. pull.

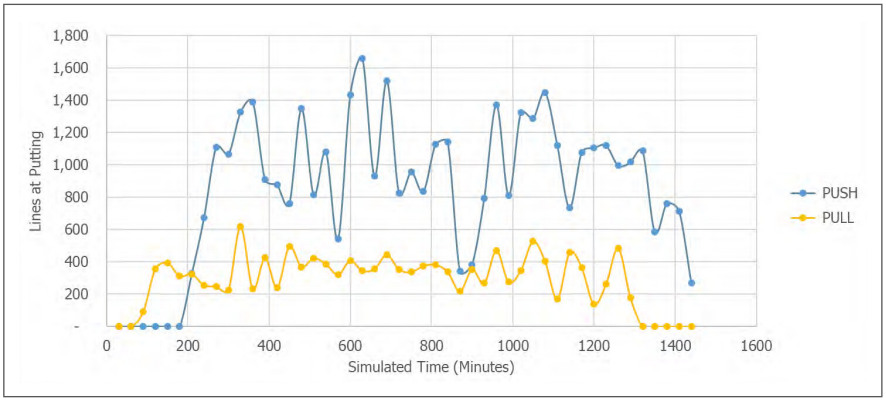

Fig. 6. Variation in number of lines being processed at the put walls over the simulated time. We see a smoother flow in pull vs. push.

Operationally, not only is the average order cycle time in the pull system 61% lower than in the push system, but there is much higher variation in the push system cycle time values, which correlates with lower service levels. Additionally, in the pull system, the orders were completed 2.7 hours ahead of push.

Under the push framework, operators at the put walls would be idle for extended time in the morning (see figure 6) and when the batches in totes started arriving to the put walls, they would be overwhelmed with work; which would result in an accumulation of totes at the put walls. Operators would then be forced to down-stack the totes on the floor to open space for other totes and unblock the conveyors. This additional work consumes work capacity and reduces productivity.

THE BENEFITS OF PULL

Some of the key benefits of pull-driven vs. push fulfillment that can be realized include:

1. **Higher Throughput Capacity:** Reclaim the wasted capacity between waves by eliminating the low-productivity transition periods and capacity losses due to queueing and accumulation.
2. **Lower Initial Investment:** Pull-based control allows smaller batches and reduces the need for large buffers when sizing put walls, which results in dollar and space savings. The downside of smaller batches is the additional number of totes flowing on the conveyor, which could lead to congestion and recirculation. But distribution centers designed for pull operations generally require lower initial investment than those designed to operate with waves because:
   - Without the low-productivity wave transitions, facility utilization is higher. The same throughput can be achieved with smaller facilities and less equipment.
   - The need for buffers required by wave-based processes is eliminated or greatly reduced.
   - In wave-based unit sortation-based operations, most orders seize chutes at the beginning of a wave, but orders do not complete until the tail of the wave. The number of chutes required for incomplete orders peaks mid-wave. Pull processing levels the requirements for chutes by holding incomplete orders in the queue, allowing for designs with fewer chutes.
3. **Higher Productivity:** In wave-based processes, low productivity periods appear at wave tails and potentially bring operations to a full stop. In pull processes, stockouts and other unexpected events affect only the orders they belong to and all other resources can continue working without any delay. Picking productivity can be higher with pull processing – even with smaller picking batches – because it eliminates work starvation periods for the pickers created by wave transitions. And pull processing reduces travel time at the put walls, which increases the productivity of that operation substantially.
4. **Better Handling of Rush Orders:** In wave-based processes, emergency orders are often held to be assigned to an upcoming wave where they will have a minimal impact on productivity. With pull-driven processing, the emergency order can be inserted as the next released order (or as the highest priority order to process) without any impact on the productivity of the operation.
5. **Enhanced Customer Service:** Typically, we see between 20% to 60% reduction in average order cycle times. The real-time nature of pull-driven processing allows the distribution center to better manage shipping deadlines. If a distribution center is processing 50 orders and realizes that the next 30 orders in line are at risk of missing their deadline, a hold can be placed on the other orders to speed up the processing of the currently at-risk orders. This hold can be cancelled when the situation is rectified. Such an approach is very difficult to process in a wave-based system.

The Role of WES in Pull

Pull-driven systems allow us to address a critical business need – reducing the dependence on labor and our ability to optimize labor deployed in the facility. WES is critical to a pull-driven system, preventing one area from getting too far ahead or too far behind the others. The constant flow of orders depends on the WES, which has visibility to machine controls to assess real-time conditions to dynamically reprioritize the work. Some Tier-1 WMS solutions offer waveless concepts, but they are often not effective in an e-commerce environment where the work needs to be constantly re-optimized as orders arrive throughout the day – the WMS (with no visibility into real-time conditions) cannot reprioritize work to mitigate equipment bottlenecks and balance flow and so efficiency is degraded. And although WCS solutions offer real-time access to the conditions on the floor, they lack the business intelligence layer needed to dynamically adjust priorities in response to service expectations and utilization. Only a WES has real-time visibility across all systems and equipment with a business intelligence layer to adjust sequencing and reprioritize work for the highest levels of efficiency.

CONSIDERATIONS

A pull-based flow for fulfilling orders can vastly improve the operational performance including reducing the cycle time, meeting throughput requirements, and leveling resource utilization. But first, the interdependency between batch sizes, resource availability, work content, and productivity need to be modeled and understood well prior to setting the pull parameters. There are unintended consequences for smaller batches, such as the need for more people or more containers and more carts to keep the work flowing and avoid resource starvation. Therefore, a flexible workforce that can react quickly to shifting resource allocation needs is necessary, and a thorough system-wide tradeoff analysis needs to be conducted prior to an architectural software design and post implementation to calibrate and revise parameters. While there is wide-spread belief that larger batches and continuously pushing the work is the more efficient method of operation, there are nonintuitive effects of variability and lost capacity.

“Pull-based control allows smaller batches and reduces the need for large buffers when sizing put walls, which results in dollar and space savings.”

> **"A pull-based flow for fulfilling orders can vastly improve the operational performance including reducing the cycle time, meeting throughput requirements, and leveling resource utilization."**

SUMMARY

E-commerce and omni-channel fulfillment are driving new requirements for the business. Pull-driven order fulfillment coupled with Warehouse Execution Systems (WES) offers a solution that delivers not only operational efficiency, but increased service levels and potential cost savings. The ability to dynamically orchestrate orders and maintain continuous flow in the DC despite the peaks and valleys in order volumes and the changing order profiles provides unprecedented flexibility for the business. In this article, we've shown that there are significant performance differences between pull vs. push control in a zone-based, parallel batch-picking operation with a downstream manual sortation operation. Pull-driven order fulfillment can meet the same throughput requirements, but with higher speed, higher service levels, more level resource utilization and smaller hardware investment cost or it can increase the throughput capacity of a system.

By using advanced analytical and modeling capabilities, Fortna is able to design and implement best-in-class distribution centers for our Clients that meet the needs of today and tomorrow.

FORTNA CAN HELP

Are you trying to decide how pull-driven flow might be a fit for your distribution operations? Fortna helps companies assess their operations, evaluate the suitability of different technologies and processes, and build a business case for investment.

For more information, contact The Distribution Experts at info@fortna.com.

THE DISTRIBUTION EXPERTS™

Fortna partners with the world's leading brands to transform their distribution operations to keep pace with digital disruption and growth objectives. Known world-wide as the Distribution Experts, we design and deliver intelligent solutions, powered by FortnaWES™ software, to optimize fast, accurate and cost-effective order fulfillment. Our people, innovative approach and proprietary algorithms and tools, ensure optimal operations design and material and information flow. We deliver exceptional value every day to our clients with comprehensive services including network strategy, distribution center operations, material handling automation, supply chain systems and warehouse software design and implementation.

CONNECT WITH US

Visit fortna.com

Contact info@fortna.com