Distribution and Federation in Real-Time, Event-Driven Business Applications

White Paper by
Paul Butterworth
February 2020

VANTIQ

Table of Contents

Overview

Around 10% of enterprise-generated data is created and processed outside a traditional centralized data center or cloud. By 2025, Gartner predicts this figure will reach 75%.

Real-time, event-driven business applications are taking center stage as companies approach how to digitally transform their business. Next generation planning, operations and customer engagement applications that provide optimal, personalized experiences all depend on real-time sensing and near real-time decision making. Such applications must be built on a modern, event-driven application platform.

The Internet of Things (IoT) has caused a huge proliferation of sensory devices to be deployed far and wide. With VANTIQ, an event-driven business application is developed in a single cloud location and then automatically partitioned and deployed to nodes in the cloud or on the edge, or a combination thereof. When responsiveness is critical, logic can be located on edge nodes to avoid latency issues that arise when data has to be sent to the cloud. The provisioning and management of these deployments are made automatic and easy to manage by intelligent features built into the VANTIQ platform.

Application components can be dynamically changed anywhere in the distributed environment for one or tens of thousands of nodes while the system is running.

VANTIQ easily includes people, when needed, through direct support for collaboration between systems and humans via mobile devices, more traditional computing devices or emerging voice and video channels.

VANTIQ's design goal is to automate as much as possible of the design, development, testing, deployment, and management of real-time, event-driven business applications which allows the business to focus on the development of these applications without having to construct the underlying infrastructure.

VANTIQ's unique combination of advanced capabilities and seamless integration dramatically improves the speed and efficiency with which event-driven business applications can be constructed, deployed, and operated.

Background

REAL-TIME, EVENT-DRIVEN BUSINESS APPLICATIONS

In this paper, a fairly expansive definition of a real-time, event-driven business application is used to cover the wide array of modern event-driven applications. Such applications have the general flow of:

- **Input is ingested** from a number of sensors, perhaps over an extended period of time. Sensors may be physical sensors, images or video coming from cameras, or data streams produced by other enterprise systems.

- The sensor **data is analyzed** using Artificial Intelligence or machine learning algortithms as necessary to produce the events, consisting of information and context, on which automation, recommendation and collaboration decisions are made. Additional context may be extracted from other systems to augment the sensor data.

- The **events are evaluated** in real time to determine the complex actions that need to be taken. In general, discrete rules and/or machine learning strategies are used to perform the real-time evaluation.

- **Actions are transmitted** to the responsible systems for implementation or human-machine collaboration is initiated with responsible personnel to determine the most appropriate response to the current situation.

In real-time event-driven applications, processing is typically better performed local to the device, improving response time and reliability. For example, in an industrial setting, managing the position of a materials handling system requires near real-time responses within a few hundred milliseconds. Such response times cannot be guaranteed by a remote decision-making system that may be delayed by thousands of milliseconds if there is a network problem.

All processing is done in a secure environment that carefully manages access to situational data and the ability to initiate control actions.

Background *(Continued)*

SENSE — DATA ACQUISITION

The notion of a sensor can be very broad including:

- Mobile devices hosting a wealth of sensor data including location, acceleration, audio, video and behavioral patterns derived from the raw sensor data.
- Wearable devices such as watches, activity trackers, health monitors, audio and video headsets.
- Machines including industrial machines, land and airborne transportation, home appliances, and any mechanical or electronic equipment that can be sensed and/or controlled. For example, imagine a robot's manipulators instrumented with pressure sensors to vary the pressure applied to objects that may have different crush points.
- Stand-alone sensors deployed in great numbers. For example, moisture sensors distributed across the fields of a farm to minimize water consumption while maximizing growth rates for the crops.
- Video and audio feeds that produce high volumes of what can be considered sensor data. Object recognition software is used to determine what the video represents to translate the video into more discrete events on which automation decisions can depend. Once an object has been defined, it may be tracked as it moves, frame by frame, through the field of vision.
- Existing enterprise applications producing streams of transactions.
- Autonomous vehicles and the collection of sensory systems that make up the framework of the vehicle's intelligence.

The sensors can be connected directly to the internet with their own IP communications stack, or indirectly connected to the Internet via an edge node. In the latter case, the sensors themselves may communicate over more specialized protocols such as Modbus or ZigBee with the edge node providing protocol conversion so that the sensors appear as virtual nodes participating in the IoT.

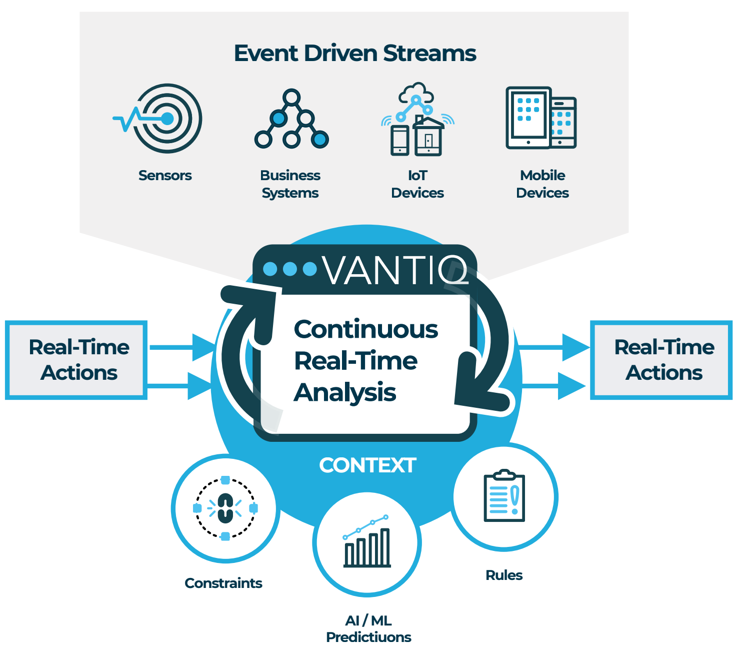
Background *(Continued)*

ANALYZE — SITUATIONAL ANALYSIS

Once the data has been acquired, a real-time, event-driven business application is responsible for analyzing the data and producing events or situations that represent business or technical conditions that require a response. The application then initiates an automatic response to the current state of the machine or customer, and/or a collaboration between the appropriate operations personnel and the system, to produce the optimal response.

Events and situations are detected by analyzing the data streams and their context using rules, statistical methods and machine learning. Examples of events or situations that might be detected during analysis include:

- Equipment that is not performing to expectations with conditions such as high temperature or low speed.
- Customers that have arrived at an interesting location in a store or facility. For example, they are standing at a checkout kiosk or a specific merchandise display.
- User is in an unsafe area of a factory and needs help.
- The distribution of orders has changed requiring the attention of product management.
- A person lying in a prone position in the waiting room of a medical facility.



Background *(Continued)*

ACT – AUTOMATION AND HUMAN-TO-MACHINE COLLABORATION

EDA drives Digital Transformation by initiating complex actions as the result of analyzing incoming events in real-time. Responses may be initiated autonomously by the system or responses can initiate collaborations among the automation system and responsible individuals. **Actions** could include:

- Providing relevant responses to consumers based on their current situation (items on sale, facility map, emergency response recommendations).
- Respond intelligently to exceptional conditions (close a valve, turn on sprinklers, stop a malfunctioning robot).
- Proactively alert personnel to opportunities/problems based on the current situation (extra delivery trucks available, shortage in part of the supply chain).
- Optimize the user or business resources to improve productivity and/or customer satisfaction (speed up an assembly line, advise sports attendees on the shortest path to their car).

Background *(Continued)*

AUTOMATED RESPONSES

In response to a situation, an automated response may be taken directly by the real-time, event-driven application or forwarded to a more specialized system for implementation. For example, an action to shut down a machine may be forwarded to the control system that directly manages the machine rather than having the application directly send a shutdown command to it.

COLLABORATIVE RESPONSES

For situations where the optimal response may be somewhat ambiguous or where determining the optimal response is beyond the capabilities of the system, a collaboration activity involving the system and the responsible individuals develops the optimal response.

For example, the sensor readings may indicate there is a potential problem that requires human intervention to resolve. Sensors in a smart building may detect a water leak that necessitates a member of the building maintenance team to investigate and fix.

Some cases in which collaboration can produce optimal outcomes:

- Exception situations for which the data streams are inadequate to uniquely define the root cause and determine the best course of action
- Situations in which the operations team is privy to additional information not available to the system
- Situations in which a manual action must be taken on the part of a system that is not controllable online
- Situations in which policies or regulations demand more in-depth analysis of the situation before an action can be taken

Another important class of collaborations notifies interested parties of actions taken and the resulting new state of the system. Notifications can be delivered to other automated systems so that they can independently respond to the situation, or delivered to responsible staff via desktop PCs, mobile devices, and/or wearable devices. Notifications can also include recommended actions and situational awareness of pending problems.

Background *(Continued)*

DISTRIBUTED BUSINESS APPLICATIONS

Many real-time, event-driven business applications are naturally distributed. In manufacturing environments, Programmable Logic Controllers (PLCs) communicate with area controllers and edge nodes that forward the data to more centralized IT systems. A Smart City initiative may include a Smart Parking application. Edge nodes may be deployed in parking garages to monitor, analyze and control sensor devices in each parking garage to identify where parking spaces are available and indicate how many spaces are available on each level. Additional edge nodes may monitor connected parking meters to track not only which spots are available but also when an occupied spot's meter will expire and should then become available. The edge nodes can aggregate the streams of data being received from individual sensors, sending summary updates at a scheduled interval. From there, a consumer can use an application that talks to the cloud to query as to where parking spaces are available.

The simplest architectures are sensors reporting to a central site. Many examples of such systems exist today. A system collecting sensor data from a mobile phone and reporting that data to a cloud service represents a common example of a centralized architecture. These simple star architectures represent the bulk of the existing event-driven business applications as they are the easiest to understand and build.

More sophisticated architectures contain additional levels of processing and connectivity. Hierarchical systems are more complex and mimic many existing physical and organizational structures. For example, an industrial IoT system that consists of sensors reporting to local controllers that report to plant-wide controllers that report to divisional headquarters that report to corporate headquarters represents a tree topology. These systems provide both centralized and decentralized monitoring and control. Such systems are more responsive in real-time or near real-time situations. For example, it would be impractical to control factory equipment in real time by collecting the data, shipping it to corporate HQ and having corporate HQ systems determine the next action for the machine. It is far more effective to do such an analysis on the local controller and simply report the situation and the action taken to the plant-wide controllers and, subsequently, to regional and

Background *(Continued)*

DISTRIBUTED BUSINESS APPLICATIONS
Continued

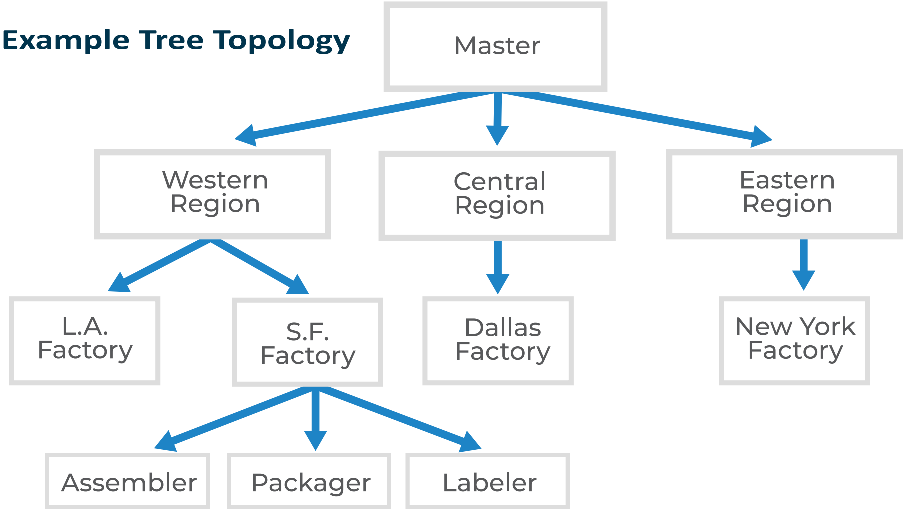
corporate HQ. Faster response times, improved availability, and local control make the distribution of the situational evaluation, collaborative decision making, and response processing across the hierarchical topology more efficient than moving everything to HQ and making all decisions in a centralized fashion.

Another classic example of hierarchical real-time, event-driven applications is the use of edge nodes to act as local processors for a collection of sensors and control points with the edge nodes, then interacting with more centralized systems.

The most sophisticated distributed real-time, event-driven business applications are peer-to-peer systems where peers are managed by separate organizations.

For example, in an electrical demand-response system, the overall system consists of sensors managed by power utilities and sensors managed by utility customers while control of the system is distributed across the utility and its customers. To provide real-time demand-response, the utility system and the customer systems must collaborate. This is accomplished by each system making local decisions and transmitting both the local situation and the local decisions to the other party and then agreeing to modify their real-time behavior based on feedback from each other.

Example Tree Topology



The Growing Need for Edge Computing

The proliferation of IoT devices, sensors, cameras, and other data collecting technologies is creating a greater need to move computing power and data analysis closer to these data sources. The following factors highlight scenarios that drive the need for edge computing.

Gartner states: "More than 90% of enterprises start with a single unique use case for edge computing; over time, a typical enterprise will have many."

HIGH LATENCY

Sensors can be located almost anywhere and depending upon the device, can generate a significant number of events with a large amount of associated data. Transmitting these streams of data to the cloud can significantly slow an applications ability to respond in real-time to the events being received. Locating compute power via edge nodes close to the sensors eliminates latency from the equation. Analysis can occur locally and any automated responses can be initiated almost immediately. The second issue resolved by the placement of an edge node is that communication across a wide-area network is eliminated and only much more robust local connectivity is required.

Also, the introduction of 5G networks will allow IoT devices to transmit even greater amounts of data to be captured and analyzed. 5G may allow for edge nodes to be placed in locations where they can be physically further from sensor devices and located to manage a cluster of devices without a reduction in latency.

BANDWIDTH

Camera feeds can also generate repetitive and uninteresting data for much of the time but in much greater amounts. Imagine a camera monitoring a loading dock to detect when trucks arrive for pickups or deliveries. For long periods of time, the loading dock may be empty. A camera feeding images to an object recognition algorithm could be sending very similar uninteresting images over and over for long periods of time. Placing an edge node local to the camera can allow processing of the stream of images that can filter out uninteresting images and reduce the amount of data that will be needed to be

The Growing Need for Edge Computing

BANDWIDTH

Continued

transferred to the cloud for either deeper processing or large term storage.

Now image a large number of cameras being put in place without local edge nodes deployed to manage these devices. The amount of data being pushed from the devices to the cloud could simply overwhelm the bandwidth available to transmit the data to the cloud.

TRANSIENT NATURE OF THE DATA

IoT data can be repetitive and much of the time, uninteresting. Take a temperature sensor as an example. If a temperature sensor is being monitored to ensure a space is heated or cooled to a specific temperature range, the sensor could be reporting temperature readings every 10 seconds that vary by one degree for hours, days or even weeks on end. Knowing that the temperature remains within range is all that is important and the need to transmit each reading back to a centralized cloud from every sensor every 10 seconds is not necessary. But, if a temperature suddenly spikes, there's not only a need to send an alert that the temperature is out of bounds but that there may be a need to take immediate action, such as shutting down a machine or turning on an A/C unit. Responding in real-time requires that the analysis is done close to the sensors being monitored and also allow for immediate automated responses to any anomalies that are observed.

PRIVACY

IoT devices may generate data that can identify individuals and violate privacy laws and regulations. In these cases, a Digital Twin may be deployed to obscure individuals but still represent the placement of someone in a location. Imagine a use case where pedestrian traffic is being monitored in a public location such as a city street, an airport terminal, or a train station. Data identifying an individual may be problematic but if object recognition technology is applied to the stream of data coming from cameras, individuals can be identified and placed within a Digital Twin of the associated space and any monitoring of the space can occur within the Digital Twin. It's critical here that the processing of the original camera images happens locally on the edge node so that the original camera feed can then be discarded and concerns with privacy alleviated.

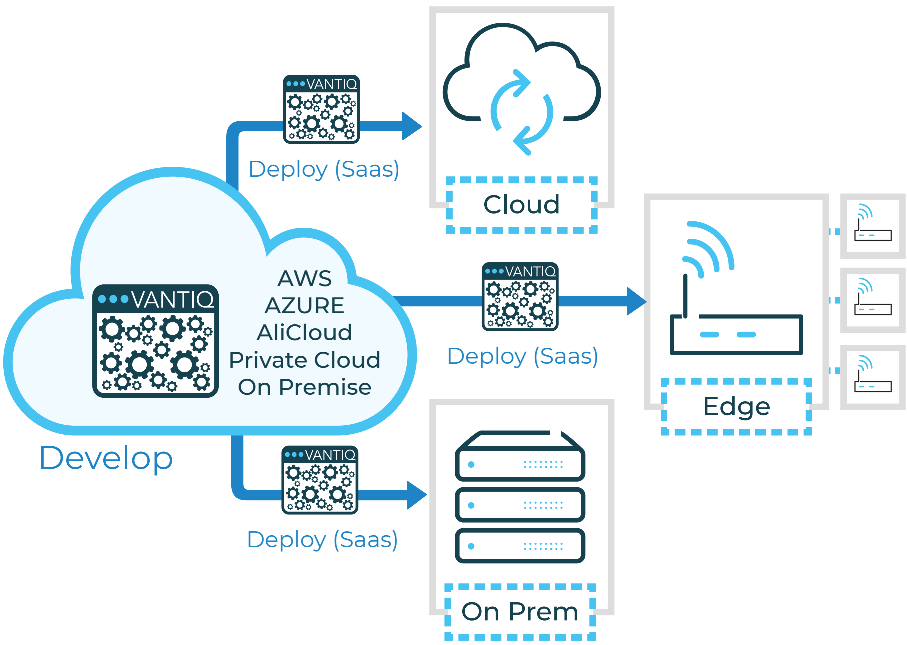
Distributed Model

TOPOLOGIES

VANTIQ supports a general model of distributed and federated topologies. A distributed VANTIQ application consists of two or more nodes with each node representing a VANTIQ installation. A VANTIQ installation can contain a single service instance or a cluster of service instances. The VANTIQ installations are assembled into a distributed topology when an installation declares at least one "peer" node with which it desires to exchange messages.

VANTIQ installations, by default, are considered independently managed. A node, X, declaring another node, Y, as a peer MUST have credentials to access node Y. Thus, the system is naturally federated since a node may only exchange messages with another node if it has been granted sufficient rights to perform the desired operation on the peer node. Peering is symmetric. If a node is defined on system X describing access to system Y this does not mean that system Y knows about system X. If you want the relationship to be bi-directional, then system Y must contain a node of its own describing system X

Since the peering relationships can be defined between any two nodes, VANTIQ can support any distributed topology. Also, the topologies are implicitly federated since



Distributed Model *(Continued)*

TOPOLOGIES

Continued

authentication and authorization are independently managed at each node.

VANTIQ anticipates that initial usage patterns will favor topologies in which all nodes in the distributed system are managed by a single authority. Such systems are typically organized into star and tree topologies:

- Star – consists of a single parent node with an arbitrary number of child nodes.
- Tree – consists of a root node with an arbitrary number of child nodes where each child node may act as a parent for an arbitrary number of child nodes.

As the deployed system becomes more collaborative, more general federated peer-to-peer networks will be constructed. In such a network topology, any node may peer with any other node leading to a general graph structure representing the connections among the nodes. The network model tends to be the most complex since cycles in the graph are possible and the cycles must be handled by any functions that operate on more than one node in the graph. Also, because each VANTIQ node represents an independent system that may require separate credentials the systems naturally generalize to federations among collaborating organizations.

FUNCTIONAL CAPABILITIES

Each node in a VANTIQ application, whether clustered in a cloud or deployed on the edge, implements a full set of VANTIQ services. All

VANTIQ nodes support the key functions of:

- Managing the Edge

Managing the Edge

SECURITY ON THE EDGE

VANTIQ includes an extensive set of security features for protecting applications developed in its environment:

- Access to the system via the REST interface requires authenticated credentials. The generated access tokens are valid for a limited time to reduce the opportunities to steal and re-use the tokens.
- Any actions requested must be authorized by the authorization system. Access controls can be coarse or fine-grained depending on the requirements of the automation system. Fine-grained authorization may specify access controls down to a single property of a single object managed by VANTIQ.
- Access to sources operates under credentials supplied by the user. Therefore, no external system can be accessed without the initiating user being properly authorized. All credentials are encrypted for increased security.
- Encrypted communications with external sources are supported where the external source supports encrypted connections.
- Properties managed by VANTIQ may be encrypted before being moved to permanent storage.
- Communications with client applications are encrypted via TLS.
- Communications among VANTIQ services are encrypted via TLS.
- All security events are logged for subsequent auditing.
- Rules can be applied to the audit logs to automatically detect suspicious activities.

Managing the Edge *(Continued)*

PROVISIONING TO THE EDGE

Key to high productivity is an efficient provisioning and management architecture. Distributed systems managed by a single authority generally operate best with centralized provisioning and management. This applies to most instances of star and tree topologies. Network topologies tend to be managed by multiple authorities obviating the need for centralized management. However, to make the federation effective, collaborative models are required for provisioning and management.

The main provisioning and management functions supported in the distributed environment:

- Deployment of core VANTIQ runtime services (installation) to remote nodes.
- Configuring the distributed topology.
- Provisioning of VANTIQ application artifacts to remote nodes.
- Several features of the provisioning system are optimized for use in distributed topologies:
- VANTIQ application artifacts may be defined on one node and automatically provisioned onto other nodes within the distributed environment either automatically by the VANTIQ system or under user control.
- VANTIQ artifacts may be changed and dynamically re-provisioned at any time.
- Capabilities are available to deploy definitions without activating them in preparation for activating the entire set of definitions once they have been fully deployed and checked.
- A versioning system is employed so that multiple versions of a definition may be created, selectively deployed, and selectively activated in a context-dependent fashion.

Managing the Edge *(Continued)*

PROVISIONING TO THE EDGE

Continued

- Declaration statements for VANTIQ artifacts can be issued from a rule. This supports sophisticated installations deploying and modifying rules automatically based on potentially complex internal situations. As a simple example, when a new node is added to the distributed environment, rules can be established to automatically deploy the relevant definitions to that node. If the network bandwidth between nodes changes, the allocation of situation detection processing between the nodes can be automatically readjusted to reflect the increase (or decrease) in bandwidth.

VANTIQ supports the provisioning of all nodes participating in a distributed topology whether the topology is a:

- Star
- Tree
- Network

As mentioned previously, provisioning may be centralized at one master node or distributed, partially or fully, across the distributed topology. Key to the VANTIQ provisioning model is that all nodes can execute independent provisioning actions. For example, if node A provisions an artifact into node B, node B may subsequently provision that artifact into Node C, if C is one of its peer nodes. Provisioning can continue until the artifact has been propagated to all nodes in a star or tree. In a network containing cycles, the system detects cycles to stop such provisioning if an attempt is made to provision an artifact onto a node on which the artifact is already provisioned.

A major consideration in remote provisioning is how to maintain consistency between the artifact definitions on the local node and the corresponding artifact definitions on the peer nodes. VANTIQ supports a range of strategies for maintaining the consistent provisioning throughout the distributed topology and for recovery from failures occurring during the provisioning activities.

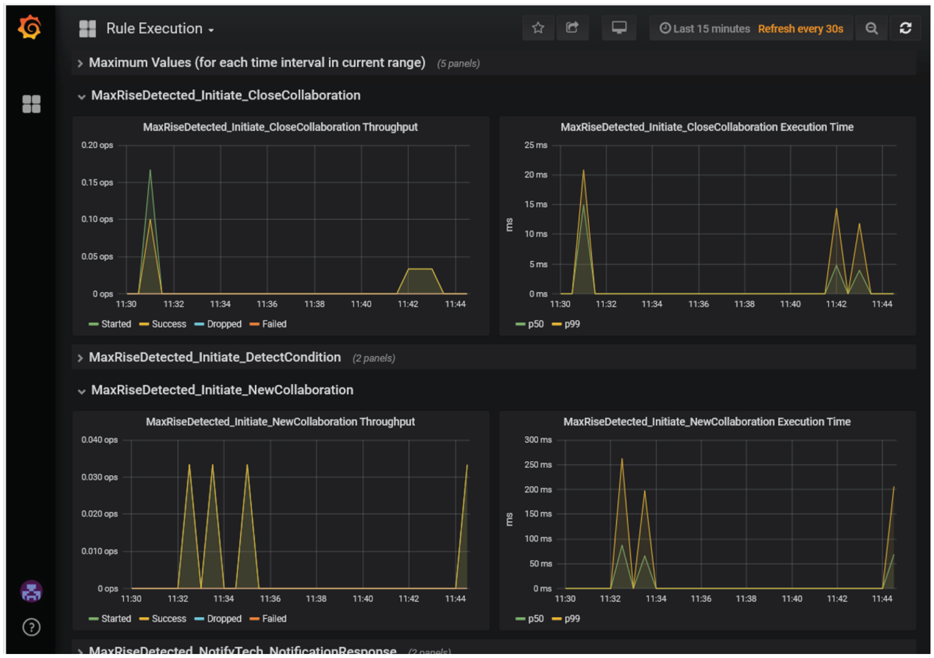
Managing the Edge *(Continued)*

MANAGING THE EDGE

The management system supports the operation of a distributed VANTIQ event-driven application:

- Reports the current health of each node in the VANTIQ system using standard metrics captured by the VANTIQ infrastructure.
- Reports performance metrics focusing on response times, throughput and capacity throughout the VANTIQ system.
- Maintains security logs for auditing security sensitive actions.

Automation rules can be applied to the security logs to automatically identify security incidents and respond by sending notifications or temporarily reducing authorized actions available to the suspect users or accounts. Similarly, actions can be specified based on performance metrics generated by the system to optimize system performance and resource usage.



Managing the Edge *(Continued)*

PERFORMANCE AND AVAILABILITY CONSIDERATIONS

Continued

In most cases, performance and availability can be improved by performing computations close to the data and relying on fewer nodes participating in the computation of a single answer. For example, if the system consists of edge nodes communicating with sensors and controllers on one side and a set of cloud services on the other, it is best if the bulk of the data is initially processed. Otherwise, latency will cause the application to incur network delays communicating with the cloud servers in the best case and unbounded downtime will be incurred in the worst case if Internet connectivity is lost or the cloud services are down.

If more than one node participates in computing a value or answering a question, the system will only work if both nodes are operational. This can be mitigated by connecting the two nodes in an asynchronous fashion via reliable messaging allowing one to be down while the other processes. Of course, this can result in uncertain delay times during periods in which one node is down.

VANTIQ facilitates optimization of data movement and decision making throughout the distributed environment by transparently moving artifacts to the optimal nodes either automatically or under user control. This facility alone provides a large increase in productivity because the functional allocation decisions can be made and changed dynamically. This allows for rapid experimentation to determine the best performing allocations and simple reconfiguration as the load or topology of the system changes.

With processing allocated to the optimal nodes in the distributed environment, there is still a need to move potentially large amounts of data among the nodes. For example, sensor readings may be read, smoothed, and summarized by an edge node but the summarized data still needs to be moved to cloud servers for use in more global optimization decisions and for forwarding to predictive analytics systems that will subsequently analyze the data looking for new business insights. With VANTIQ's support for a wide range of

Managing the Edge *(Continued)*

PERFORMANCE AND AVAILABILITY CONSIDERATIONS

Continued

high-performance messaging systems that can be configured as the links between nodes in a distributed environment, optimal data transmission architectures can be constructed. As with everything else in VANTIQ, the link definitions can be configured dynamically and changed at any time. Both reliable and unreliable, synchronous and asynchronous protocols are supported; as well as IoT optimized messaging such as MQTT and AMQP and standard REST protocols among others. The user can choose the best protocol(s) for each pair of peer nodes in the distributed topology.

With VANTIQ, the system architect has complete control over the style of interaction between the VANTIQ nodes. Also, because of the dynamic nature of VANTIQ, the messaging bindings can easily be changed dynamically.

Summary

VANTIQ provides higher levels of abstraction whereby the underlying complexities (distributed environments, IoT, mobile, collaboration, etc.) are hidden. The development of event-driven applications can focus on the business logic rather than infrastructure and the challenges of new technology.

Real-time, event-driven business applications are naturally distributed systems so they can achieve the necessary scale, resiliency and security. VANTIQ automates the partitioning of the logic and provisioning to where it is the most effective.

To keep up with the pace of business, real-time, event-driven business applications will continue to evolve rapidly over time. To satisfy this requirement, VANTIQ offers comprehensive facilities to provision then monitor and adapt the deployed systems.

With the VANTIQ platform, all enterprises can leverage their existing skill sets to augment legacy systems of record and quickly build custom, real-time, event-driven applications to power their digital business.

For more information, contact Info@VANTIQ.com.

ABOUT VANTIQ

Customers around the globe rely on VANTIQ to quickly and easily create the next generation of transformative digital applications to serve the Internet of Things (IOT), smart cities/buildings, oil and gas, healthcare/life sciences, and telecommunications, among other industries. VANTIQ powers these mission-critical real-time business operations with our low code event-driven architecture (EDA) application development platform. Founded in 2015 by renowned business and technology leaders Marty Sprinzen and Paul Butterworth, VANTIQ dramatically reduces time-to-market, significantly lowers development and maintenance costs, and provides maximum agility in response to constantly changing operational requirements. Learn more at www.vantiq.com.