WHITEPAPER

Conga RLM – CPQ

Performance and large carts

Conga CPQ is built to price large carts with line items exceeding 50k

Table of contents

1 Executive summary

Conga's Revenue Lifecycle Management (RLM) platform, the Advantage Platform, is a multi-tenant cloud platform designed and optimized for Revenue Lifecycle Management processes. This white paper focuses on the performance and scalability of Conga's Configure, Price, Quote (CPQ) solution, particularly emphasizing its ability to handle complex CPQ computations efficiently.

Highlights of Conga CPQ:

1. Ability to process large carts with line items exceeding 50,000.
2. Designed with performance and scalability in mind, leveraging the platform built on top of Kubernetes.
3. Built on a highly concurrent architecture using the Actor model, enabling parallel async processing of complex pricing computations and highly scalable computations of complex bundle configurations.
4. Conga CPQ UI leverages GraphQL for efficient, real-time data fetching, enabling faster and more responsive user interfaces.
5. API-first approach, allowing every CPQ action to be performed instantaneously through REST APIs, enabling extensibility.
6. Continuous benchmarking by the performance engineering team using API-based persona tests, stress tests, long-running tests, volume tests, and UI-based persona tests enabling a well-tested, scalable, and performant product.
7. Telemetry everywhere enables the RLM platform to provide end-to-end tracing capabilities across multiple microservices using the concept of distributed tracing and open telemetry standards. Platform observability dashboards provide a single pane of glass to monitor the platform and application services continuously.

In conclusion, Conga's Revenue Lifecycle Management platform, with its powerful CPQ solution emphasizing performance-supporting large carts, empowers businesses to manage complex revenue processes efficiently, drive growth, and optimize their revenue management processes. The platform's robust architecture, focuses on performance and scalability, and continuous enhancements make it a compelling choice for organizations seeking to streamline their revenue lifecycle and gain a revenue advantage.

2 Conga Revenue Lifecycle Management platform

Conga's Revenue Lifecycle Management (RLM) platform is a comprehensive platform designed to streamline and optimize Revenue Lifecycle Management processes. This white paper aims to provide an in-depth analysis of the RLM platform's Configure, Price, Quote (CPQ) performance and scalability, focusing on large cart performance. The Conga RLM Platform has been architected with a unified data model and key focus on performance and scale to enable businesses to leverage Conga RLM to drive growth and improve their revenue management processes.

2.1 What is Revenue Lifecycle Management?

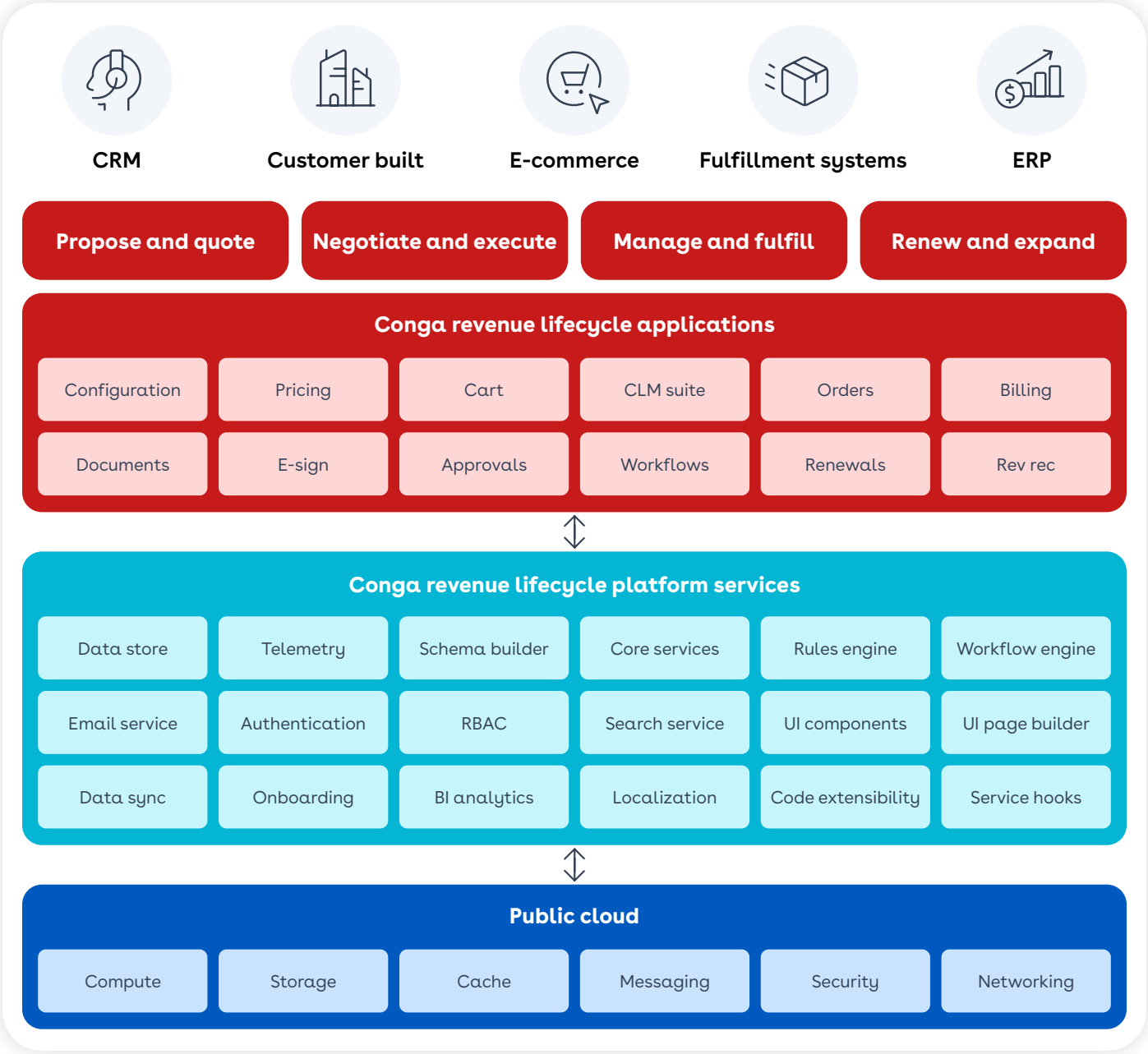
Conga Revenue Lifecycle Management is a multi-tenant cloud platform that supports the complexities of Revenue Lifecycle Management, leading to improved efficiency and certainty for businesses to drive configure, price, quote, contract, negotiate, manage, collect, and renew revenue. It helps unify and automate all revenue-generating processes to help customers increase lifetime value.

Automate: Prepare proposals and quotes for prospects automatically—and negotiate and execute various contracts to close your next deal.

Integrate: Manage billing, invoicing, and fulfilling obligations all in one place—no matter your CRM, ERP, or data source.

Grow: Renew and expand your accounts while coordinating your business function within a single source of truth to achieve revenue results.

2.2 Platform architecture



The Conga Advantage platform consists of three main layers,

a. The Cloud layer consists of various cloud services from major cloud providers such as AWS, Azure, etc.,

b. The Platform services layer offers foundational services to revenue lifecycle applications, including data storage, telemetry, security, caching, search, and other core capabilities required for any application to be hosted on the platform.

c. The Application layer consists of various revenue lifecycle applications such as Configure, Price, Quote (CPQ), Manage & Fulfil, and Renew & Expand which interact with platform services APIs without having to worry about specific cloud providers or infrastructure or the underlying technology used.

Conga's platform services make use of Cloud PaaS (Platform as a Service) getting significant benefits of cloud-native efficiency, auto-scale, ease of management, and upgrades. High-security infrastructure is built on the public cloud making use of compute, storage, cache, messaging, and networking components of the cloud providers. The cloud layer is integrated with a Web application firewall (WAF) to provide secured access to our customers eliminating DDoS attacks, cross-site scripting (XSS), and other security risks.

The platform services layer consists of services architected from the ground up to achieve high scale and performance. These services are designed to be cloud-agnostic and capable of integrating with any cloud provider. All the complexities of different cloud vendors are automatically handled at this layer making building applications on the platform super easy.

Some of the major platform services are as follows.

a. **Data storage services:** These services take care of all data storage management aspects including managing schema and data reads/writes operations.
b. **Telemetry services:** The Platform ingests billions of telemetry signals on a daily basis in a scalable way to give end-to-end observability required to maintain and troubleshoot revenue lifecycle applications on the platform.
c. **Authentication and role-based access control (RBAC):** The platform supports various authentication mechanisms along with sophisticated Role-based access control to provide the right access and support various security controls to safeguard customers and data.
d. **Workflow and rules engine:** These services play a key role in bringing extensibility to the platform. Customers can design custom workflows and add rules to them to reflect their specific revenue processes that are crucial for meeting their use cases and business needs.
e. **Localization services:** Localization and Internalization enable customers to expand their operations to multiple countries and regions. This includes handling language support, currency, data time formats, number formats, and compliance with international regulations and standards.
f. **Customer onboarding and management services:** These services take care of setting up a customer in the platform by registering a customer organization, managing licenses, creating users and master data, configuring auth, and setting up the configuration and necessary infrastructure.
g. **Services hooks and code extensibility:** Service hooks let the customers configure the system to take action when certain events are triggered within the platform. The code extensibility piece provides a mechanism for customers to author custom code that can be integrated with service hooks, workflow, or API integrations for the revenue life cycle applications.
h. **UI components and page builder:** The platform provides UI extensibility in the form of Conga component library and UI page builder services. These tools help customers extend the existing application UI or add new UI pages, widgets, and components to the existing UI resources to satisfy business requirements.
i. **Search services:** Search is a critical component of the platform. It provides customers with the ability to quickly and efficiently find relevant information from data, such as documents and records. The platform supports various search types, such as simple, look ahead, and advanced search.
j. **Data sync services:** Data sync services bridge the gap between any CRM system with the revenue lifecycle cloud. These services provide bidirectional data sync between the platform and various external systems such as CRMs and others. BI Analytics and Reporting Services: BI and Reporting services provide customers with tools and capabilities to analyze operational and usage data to generate meaningful reports and gain insights into revenue lifecycle-related processes and performance.

2.3 Key attributes of the platform

Conga Revenue Lifecycle Management is a multi-tenant cloud platform that supports the complexities of Revenue Lifecycle Management, leading to improved efficiency and certainty for businesses to drive configure, price, quote, contract, negotiate, manage, collect, and renew revenue. It helps unify and automate all revenue-generating processes to help customers increase lifetime value.

- **API first:** The platform is built using the API-first design principle where well-defined business APIs become first-class citizens and are published to customers to consume and build applications or utilize the application provided by Conga powered by the same business APIs. This enables organizations to extend the functionality of the platform, integrate with third-party services, and build custom solutions tailored to their specific needs.
- **Scalability:** The platform is designed to handle a high volume of users, transactions, and data, providing seamless scalability as users, transactions, and data load grow. The platform is built using highly scalable patented data storage technology that provides low-latency multi-tenant data access in a scalable way, minimizing noisy neighborhood problems that exist in most commercial database management systems implementing multi-tenant data architecture. The data platform makes use of modern NoSQL data modeling techniques along with the support of an extensible data model for customers to extend the data schema to suit their workflows and business needs. The platform uses a sophisticated and elegant approach to scale infra elastically using cloud-native technologies such as auto-scaling, load balancing, and elastic resources to handle varying workloads, effectively serving thousands of conga customers.
- **Performance:** The platform cloud architecture focuses on performance at scale to ensure fast response times and high throughput for the transactions and workloads. The data storage architecture is tailored and built to provide low-latency data access for revenue lifecycle applications. It employs NoSQL data modeling architecture, caching, and indexing to enhance application performance, resulting in fast processing and response times.
- **Security and data isolation:** The platform architecture emphasizes security first design methodology and prioritizes security to support various security controls and measures to safeguard customers and data. It employs robust access controls, encryption, and other security measures to protect sensitive information. The platform uses WAF(Web Application Firewall) at the entry point to prevent all types of attacks including cross-site forgery, cross-site scripting (XSS), DDoS, etc. The platform supports various types of Authentication mechanisms via OAuth, OIDC, and SAML protocols to support different customer scenarios. Customers can choose to use either Conga Identify Provider (IDP) or use other external Identity providers such as Salesforce, Microsoft, and Okta to perform the single sign. The data stored in the platform is always encrypted at REST and in transit providing the highest level of data security in the platform. The Role Based Access Control (RBAC) plays an important role in allowing users to see and act on data the user is authorized for. RBAC implements row-level security called data scopes that lets only authorized users see and act on data. RBAC action permissions let users perform certain operations based on the role assumed.
- **Cloud agnostic:** The platform is designed to be cloud agnostic, meaning it can be deployed on various cloud platforms such as Amazon Web Services (AWS) or Microsoft Azure. This flexibility allows organizations to choose from multiple cloud providers adopting a multi-cloud strategy based on their specific requirements and preferences.

- **CRM agnostic:** The platform is also CRM agnostic, meaning it can seamlessly integrate with different Customer Relationship Management (CRM) systems such as Salesforce, Microsoft Dynamics, or other CRM Systems. This flexibility allows organizations to leverage their existing CRM investments without being tied to a specific CRM vendor.
- **Reliability and availability:** The platform is architected for high reliability and availability of services. This is achieved by building enough redundancy in the system to take care of failover mechanisms and data replication across multiple locations to minimize downtime and ensure business continuity. Service Level Agreements (SLAs) guarantee a certain level of uptime and availability.
- **Telemetry and observability:** The platform is architected to ingest billions of telemetry signals on a daily basis. Highly scalable telemetry architecture provides end-to-end tracing capabilities across multiple microservices using the concept of distributed tracing and open telemetry standards. Platform Observability dashboards provide a single pane of glass to continuously monitor the platform and help understand how the system is behaving with respect to infrastructure and services deployed in the cloud. With the data collected, one can analyze performance bottlenecks, check various resource utilization patterns and usage analytics, detect errors and anomalies in the system proactively, and take corrective actions.
- **Platform extensibility:** The platform provides features that extend baseline capabilities beyond its out-of-the-box functionality allowing customers to tailor the applications to their unique workflows, processes, and business logic. Schema builder lets customers extend the data model. The platform supports a workflow engine and rules engine that helps build custom workflows and custom business logic for use cases that can be completely customized to solve unique business problems for customers. Service hooks let the customers configure the system to take actions when certain events are triggered. The platform provides UI extensibility in the form of a Conga component library and UI page builder services. These tools help customers extend the existing application UI or add new UI pages, widgets and components to the existing UI resources to satisfy business requirements.
- **Microservices-based architecture:** The platform architecture adopts a microservices-based approach, where different functional computation application services are decoupled and developed as independent services. This architecture enables better maintainability, scalability, and fault isolation. It also allows for continuous deployment, as updates or new features can be rolled out to specific microservices without impacting the entire system.
- **Multi-tenant architecture:** Conga's Revenue Lifecycle Cloud is built on a multi-tenant architecture, where a single set of microservices the application serves multiple organizations or tenants. This architecture allows efficient resource utilization and cost-effectiveness by sharing infrastructure, data stores, and services among tenants while ensuring data isolation and security.
- **Unified data model:** The platform uses a single, standardized view of data across different systems and applications within an enterprise. This helps eliminate data silos, reduce redundancies, and enable seamless data sharing and integration across the organization. This allows for better collaboration among cross-functional teams involved in the revenue lifecycle and eliminates silos in decision-making with a single source of truth.

3 CPQ and complexity

Configure, Price, and Quote software is a system that allows a company to generate quotes using real-time data to determine how much a product or service will cost a customer while maximizing margin. Also known as CPQ, it can calculate complex product configurations and pricing, instantly using a range of preset variables.

One of the big challenges with CPQ solutions is the ability of the product to handle complexity, and scale without compromising its performance. Customers have very different needs regarding the complexity of their products, prices, and quotes. For example, if a user wants to configure a product that has several thousands of part numbers and the complexity of the product is high, they need a powerful configuration engine. Some customers have a high complexity regarding their product pricing; for example, every quote has several hundred different products, and customers would like to see real-time pricing.

3.1 Why Conga RLM – CPQ?

Conga CPQ is built on the Conga RLM platform (described in Sec-3 above) and is composed of multiple microservices. Conga CPQ is designed and architected to solve some of the major challenges associated with the business complexities associated with complex configurations & rules, different cart sizes, and pricing calculations.

- **API driven:** Leveraging API first approach of Conga Platform, all interaction with Conga CPQ is through a set of REST APIs designed specifically for admin and runtime operations with clear segregation between the two. This allows the runtime APIs to be independently scaled depending on the customer and load. This design provides the ability to perform every CPQ action on Cart, Config, Pricing, and Assets instantaneously leveraging REST APIs, also enabling extensibility. All the runtime and admin capabilities of CPQ can be exploited through REST APIs. This is best suited for system-to-system integrations.
- **Highly performant:** Conga CPQ uses some of the cutting-edge technologies under the hood. Designed with performance in mind, it's built on a well-established, highly concurrent architecture using the Actor model and is deployed under the industry-accepted Kubernetes hosting platform. The actor model enables the parallel async process of complex pricing computations and highly scalable computations of complex bundle configurations.
- **Highly scalable:** Conga CPQ is designed with scale and performance considerations from the beginning, making the most of Conga Platform design & architecture. Conga CPQ allows scalability at multiple levels, complex carts as well as thousands of current users across multiple tenants. Conga CPQ is powered with GraphQL, enable to optimize application performance and speed. With this enablement, payload size is reduced, resulting in less data needing to be processed and rendered on the client side, leading to faster load times and a better user experience. Additionally, the ability to handle complex queries in a single request minimizes the overhead associated with multiple network calls.
- **Large carts:** Conga CPQ is designed to handle large carts efficiently, overcoming the limitations imposed by Salesforce Governor Limits. With Conga CPQ, businesses can create and process carts with line items exceeding 50,000, enabling them to manage complex deals and high-volume transactions seamlessly. The platform's robust architecture, optimized for performance and scalability, ensures that users can work with large carts without experiencing significant performance degradation or system errors. This capability empowers sales teams to focus on closing deals and delivering value to customers, rather than worrying about technical limitations hindering their productivity.

- **Overcoming the limits of Salesforce:** Salesforce, being a multi-tenant platform, imposes various limits known as "Governor Limits" to prevent resource abuse by a single tenant. These limits significantly impact the creation and processing of large carts in CPQ applications. Key limitations are shown below. These constraints hinder the ability to efficiently create and manage large carts with complex configurations and pricing rules, leading to performance issues and a suboptimal user experience. As a result, businesses using Salesforce CPQ often struggle to handle large carts (1,000-line items or more) effectively, necessitating the adoption of alternative solutions like Conga CPQ.
- **Extensibility framework:** Conga CPQ capability is exposed through various hook points along the processing pipeline, allowing customers and implementation teams to add additional customization and extensibility logic to enhance the engine's capability. Extensibility is achieved through a set of pre-defined hook points, which let a professional services or partner developer write scripts to augment the existing functionality.

4 Performance measurement strategy

Conga CPQ performance is continuously benchmarked in two different ways by the performance engineering practice at Conga.

- API-based persona tests, stress tests, long-running tests, and volume tests.
- UI based persona tests that capture the end-to-end performance emulating user behavior during peak load.

Performance gate

Performance gate is a quality gate that is included as part of the Conga release process. Every application, microservice, and platform service is benchmarked so as to meet the SLAs established by the business. The performance team runs a series of API and UI persona-based end-end tests as part of every release ensuring SLAs are met. As part of our release criteria, Performance Gate has helped us to monitor the scale & performance of the applications continuously and helped us to identify any performance bottlenecks, way early in the product development cycle.

Performance Lab

Performance Lab @ Conga has been established to emulate Multi-product, Multi-tenant, and Multi-persona based loads to mimic the actual production-like load in our internal testing. The primary objective of this exercise is to confirm the performance and scale aspect of our products in real load conditions. The lab is to be running continuously, updated with the latest builds showcasing Conga Platform & Products resilience and scalability with Grafana dashboards (that can be shared with customers) which can be used to slice/dice performance by day, week, month or compare. The customer benchmarks shared below were run in the performance lab with actual customer-like data sets and complexity.

On the next page is a summary of large cart performance, API, UI and customer performance tests.

4.1 Large cart performance

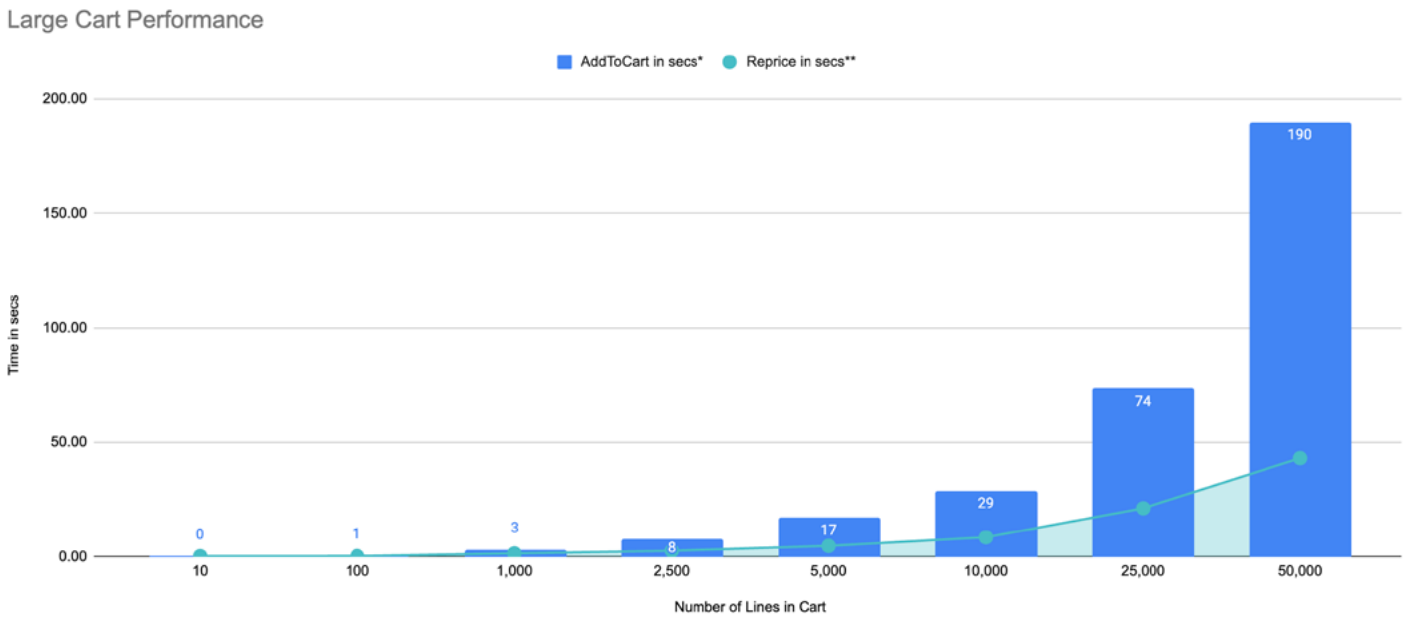
Large carts are carts with lines greater than 1,000 lines. The table below illustrates large cart performance with standalone products, using average response time as a key metric.

Lines in cart	Add to cart in seconds*	Reprice in seconds**
10	0.364	0.347
100	0.668	0.412
1,000	3.22	1.548
2,500	7.849	2.692
5,000	17.246	4.82
10,000	29.067	8.587
25,000	73.656	21.292
50,000	189.907	43.172

Table below illustrates the large cart performance with mix of standalone and Bundle products.

Lines in cart	Add to cart in seconds*	Reprice in seconds**
5,000	16.363	5.097

**Reprice of 200 lines, *Average



5k API end-end large cart test details

Below is an end-end API-based large cart performance test that illustrates running continuously for 4 hours without errors while the 95% and 99% percentile times remain steady.

Label	# Sample	Average	Median	90% line	95% line	99% line	Error %
CreateQuote	27	208	200	234	325	414	0.00%
LaunchCart	27	887	872	1,025	1,074	1,194	0.00%
LoadCatalog	27	387	374	458	541	665	0.00%
AddLineItems	27	2,087	2,068	2,191	2,228	2,253	0.00%
AddLineItem_Status	285	2,397	451	9,091	9,190	9,255	0.00%
AddToCart(ElapsedTime)	27	25,394	25,429	25,739	25,826	25,856	0.00%
AddLineItem_Complete_Status	27	3,398	3,401	3,443	3,459	3,460	0.00%
Reprice_200Lines	27	232	232	252	255	294	0.00%
Reprice_Status	27	2,657	2,627	3,034	3,043	3,083	0.00%
Reprice(ElapsedTime)	27	7,829	7,823	7,955	7,978	8,146	0.00%
Reprice_200Lines_Complete_Status	27	3,232	3,237	3,273	3,288	3,300	0.00%
Finalize	27	183	181	233	238	293	0.00%
Finalize_Status	55	1,289	129	2,576	2,581	2,583	0.00%
Finalize(ElapsedTime)	27	3,840	3,812	3,878	3,967	4,435	0.00%
CheckOut	27	48,725	48,659	50,196	51,308	51,421	0.00%
DeleteCart	27	9,291	8,398	9,866	11,294	30,763	0.00%

4.2 API-based stress test

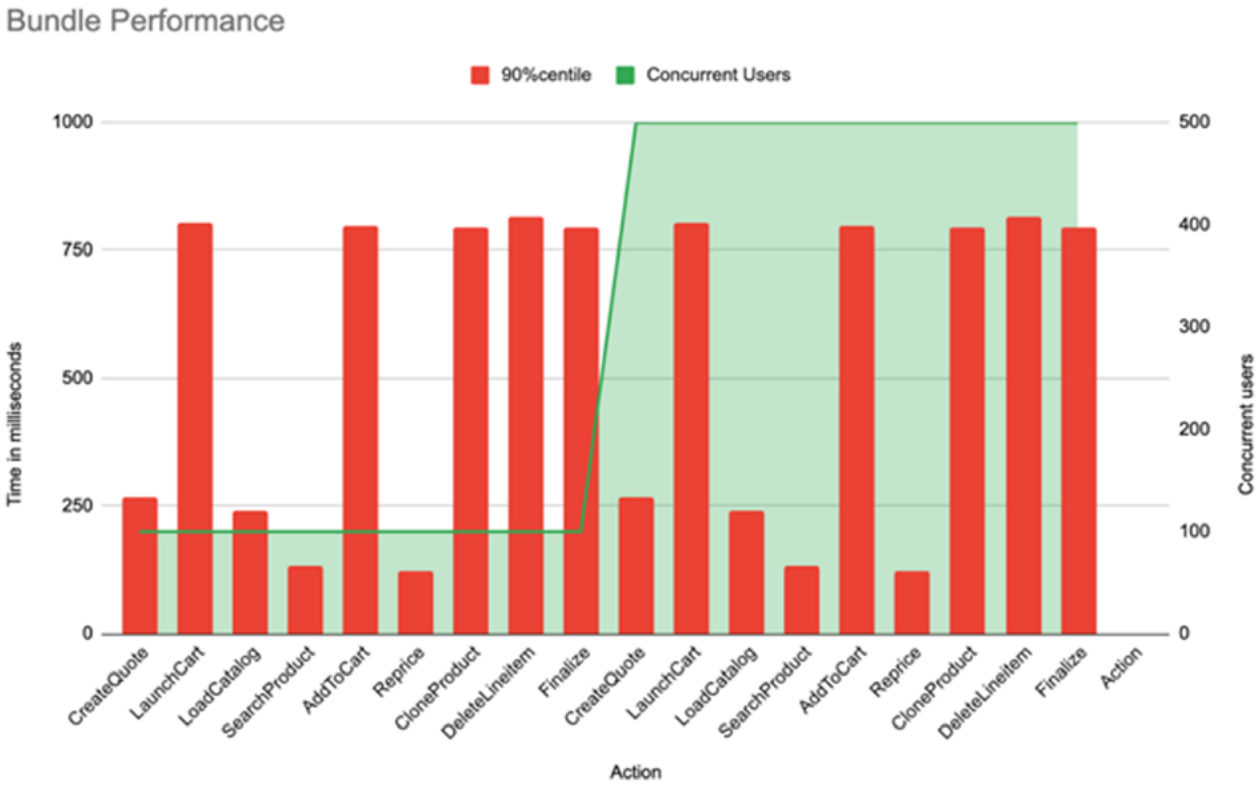
Stress testing is a type of performance test that checks the upper limits of the complete system by testing it under extreme loads. Stress tests examine how the system behaves under intense loads and how it recovers when going back to normal usage. Conga products go through varying levels of stress tests, API-based stress test helps to determine all the potential bottlenecks in the processing pipeline when the system is subjected to stress through API. Observability and tracing built into the system help narrow down the cause that can be taken up for further analysis and fixing.

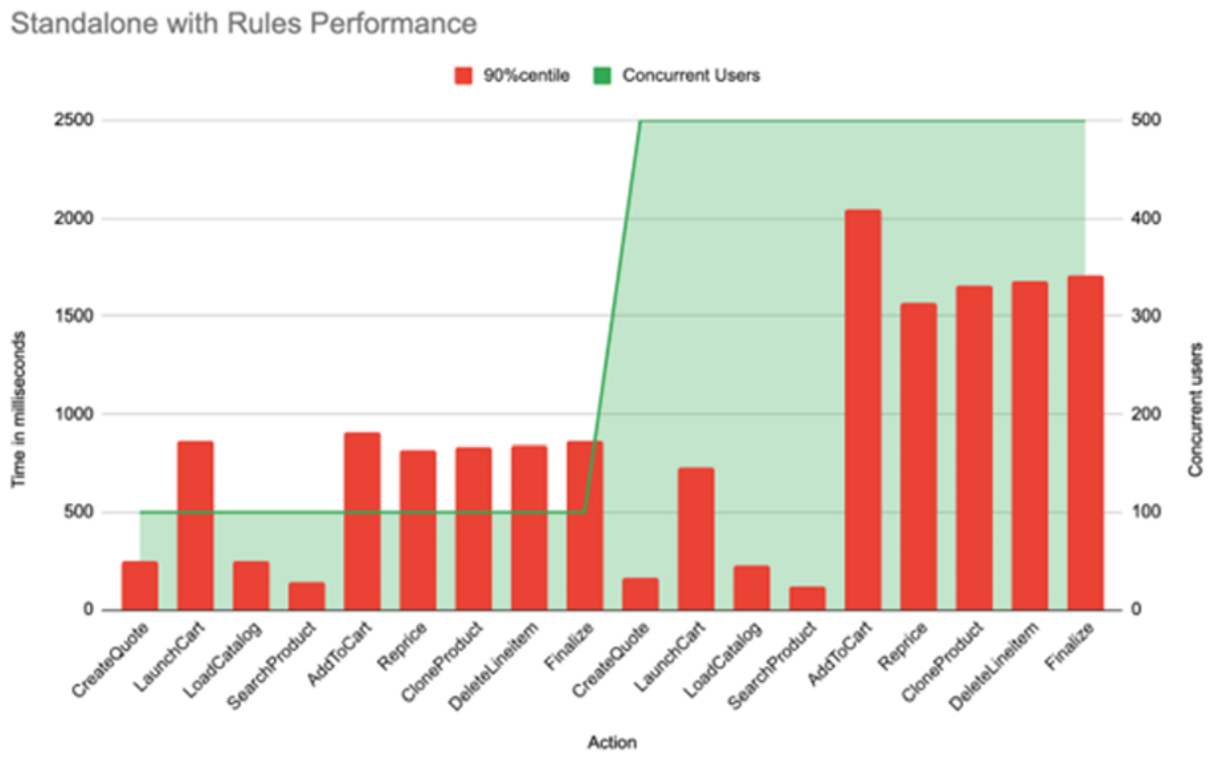
Below is a short summary of the results seen from one of the stress tests containing 100 and 500 concurrent users API tests that use a mixture of 50 Standalone, 100 Standalones with Rules, Bundle that adds 13 options; in total about 163 line items are added to cart, repriced, cloned, deleted and finalizing the cart. The table also shows creating the quote, launching the cart, loading the catalog, searching for a product and then adding to the cart.

Action	Users	Line items	90 percentile in milliseconds	Throughout
Bundle_01_CreateQuote	100	13	268	125
Bundle_02_LaunchCart			805	91
Bundle_03_LoadCatalog			239	91
Bundle_04_SearchProduct_1			132	91
Bundle_AddToCart(ElapsedTime)			798	91
Bundle_Reprice(ElapsedTime)			122	91
Bundle_CloneProduct(ElapsedTime)			794	91
Bundle_DeleteLineitem(ElapsedTime)			815	91
Bundle_Finalize(ElapsedTime)			795	91
Standalone_01_CreateQuote		50	247	91
Standalone_02_LaunchCart			866	91
Standalone_03_LoadCatalog			248	91
Standalone_04_SearchProduct_1			143	91
Standalone_AddToCart(ElapsedTime)			911	91
Standalone_Reprice(ElapsedTime)			821	93
Standalone_CloneProduct(ElapsedTime)			829	69
Standalone_DeleteLineitem(ElapsedTime)			842	69
Standalone_Finalize(ElapsedTime)			866	69
Standalone_wcr_01_CreateQuote		100	194	69
Standalone_wcr_02_LaunchCart			843	139
Standalone_wcr_03_LoadCatalog			241	69
Standalone_wcr_04_SearchProduct_1			139	69
Standalone_wcr_AddToCart(ElapsedTime)			2,083	69
Standalone_wcr_Reprice(ElapsedTime)			1,556	69
Standalone_wcr_CloneProduct(ElapsedTime)			1,664	69
Standalone_wcr_DeleteLineitem(ElapsedTime)			1,684	69
Standalone_wcr_Finalize(ElapsedTime)			1,738	69

Action	Users	Line items	90 percentile in milliseconds	Throughout
Bundle_01_CreateQuote	500	13	206	346
Bundle_02_LaunchCart			754	389
Bundle_03_LoadCatalog			224	503
Bundle_04_SearchProduct_1			129	488
Bundle_AddToCart(ElapsedTime)			755	488
Bundle_Reprice(ElapsedTime)			726	488
Bundle_CloneProduct(ElapsedTime)			762	455
Bundle_DeleteLineitem(ElapsedTime)			728	455
Bundle_Finalize(ElapsedTime)			745	455
Standalone_01_CreateQuote		50	179	455
Standalone_02_LaunchCart			760	455
Standalone_03_LoadCatalog			231	455
Standalone_04_SearchProduct_1			131	455
Standalone_AddToCart(ElapsedTime)			848	455
Standalone_Reprice(ElapsedTime)			793	344
Standalone_CloneProduct(ElapsedTime)			784	353
Standalone_DeleteLineitem(ElapsedTime)			781	393
Standalone_Finalize(ElapsedTime)			824	370
Standalone_wcr_01_CreateQuote		100	164	370
Standalone_wcr_02_LaunchCart			726	594
Standalone_wcr_03_LoadCatalog			222	345
Standalone_wcr_04_SearchProduct_1			124	345
Standalone_wcr_AddToCart(ElapsedTime)			2,049	345
Standalone_wcr_Reprice(ElapsedTime)			1,564	345
Standalone_wcr_CloneProduct(ElapsedTime)			1,658	345
Standalone_wcr_DeleteLineitem(ElapsedTime)			1,680	345
Standalone_wcr_Finalize(ElapsedTime)			1,713	345

Benchmark with a mix of products (Standalones, Standalones with rules, Bundles with default and configured options) and 100/500 concurrent users.





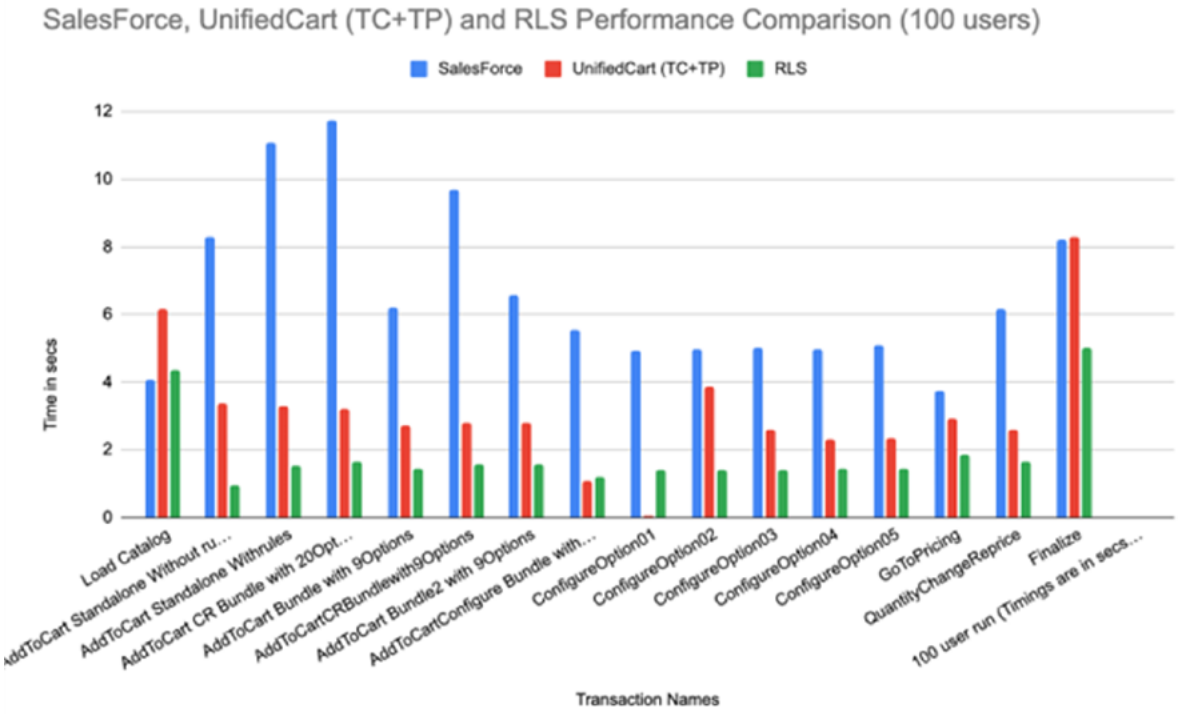
4.3 Persona-based UI load test

A persona-based load test measures the concurrent end user experience under peak load. These maps are close to what the end user will experience when using the system. This considers the number of concurrent users in the system, active users pricing a cart, activity distribution based on the business process, etc., with adequate think time to reflect the actual user behavior.

Transaction names	Salesforce	UnifiedCart (TC+TP)	Conga RLM CPQ
Load Catalog	4.08	6.18	4.38
AddToCart Standalone Without rules	8.28	3.37	0.97
AddToCart Standalone Withrules	11.09	3.30	1.54
AddToCart CR Bundle with 20Options	11.75	3.21	1.67
AddToCart Bundle with 9Options	6.21	2.73	1.44
AddToCartCRBundlewith9Options	9.70	2.79	1.58
AddToCart Bundle2 with 9Options	6.58	2.79	1.55
AddToCartConfigure Bundle with 9Options	5.56	1.08	1.22
ConfigureOption01	4.93	0.06	1.42

Transaction names	Salesforce	UnifiedCart (TC+TP)	Conga RLM CPQ
ConfigureOption02	4.97	3.86	1.42
ConfigureOption03	5.00	2.59	1.42
ConfigureOption04	4.95	2.31	1.44
ConfigureOption05	5.11	2.34	1.44
GoToPricing	3.72	2.91	1.86
QuantityChangeReprice	6.17	2.59	1.63
Finalize	8.23	8.28	5.02
100 user run (Timings are in seconds)(run with LoadRunner)			

UI Persona (Salesforce, Unified Cart (TP+TC), Conga) benchmark with 100 concurrent users



4.4 Early customer implementation performance

C1

	SFDC	RLM
Transaction Name	**90 percentile**	**90 percentile**
General Functionalities Persona		
Activate cart	9.09	1.51
Update qty	13.63	2.80
Delete product	10.03	1.54
Add products to cart	8.41	1.07
Update configuration	8.62	3.10
Export Cart	0.8	0.43
Create RFQ	6.2	5.38
Place Order	15.4	4.13
Total time to complete order in seconds	**72.18**	**19.96**

* As measured in the performance lab before going live

** Performance gains limited by customer custom code

*** Only a subset of actions shown

A discrete manufacturing customer uses Conga's Digital Commerce solution that serves as a comprehensive B2B and B2C platform for a global manufacturing organization, supporting multiple product offerings including standalone and configurable items. It caters to diverse user personas, both internal and external, across various Business Units worldwide. The system integrates complex pricing, configuration, and order processes, enabling direct orders and quote requests for all product lines.

Throughout the project lifecycle, from initiation to deployment, performance was continuously benchmarked in the Performance Lab. This rigorous testing process simulated real-world scenarios with large user volumes, ensuring optimal performance and eliminating potential surprises when the solution was launched in production.

C2

	SFDC	RLM
Transaction Name	**Wall clock time seconds**	**90 percentile in seconds**
FLOW1_08_SaveProposal		9.284
FLOW1_09_ConfigureProducts	27.2	10.994
FLOW1_10_SelectCategory	17.59	1.246
FLOW1_11_RefineByFilter	5.34	0.134
FLOW1_12_AddToCart_25_of_50	0.76**	3.295
FLOW1_13_GoToNextPage	0.73	2.232
FLOW1_14_AddToCart_50_of_50	0.6**	4.122
FLOW1_15_GoToPricing_50_of_50	104.96	12.867
FLOW1_16_LoadApplyCommitLevel_50_of_50	3.79	7.379
FLOW1_19_SaveApplyCommitLevel_50_of_50	133.1	57.903
Total time to complete order in seconds	**294.07**	**109.456**

* As measured in the performance lab before going live

** Performance gains limited by customer custom code

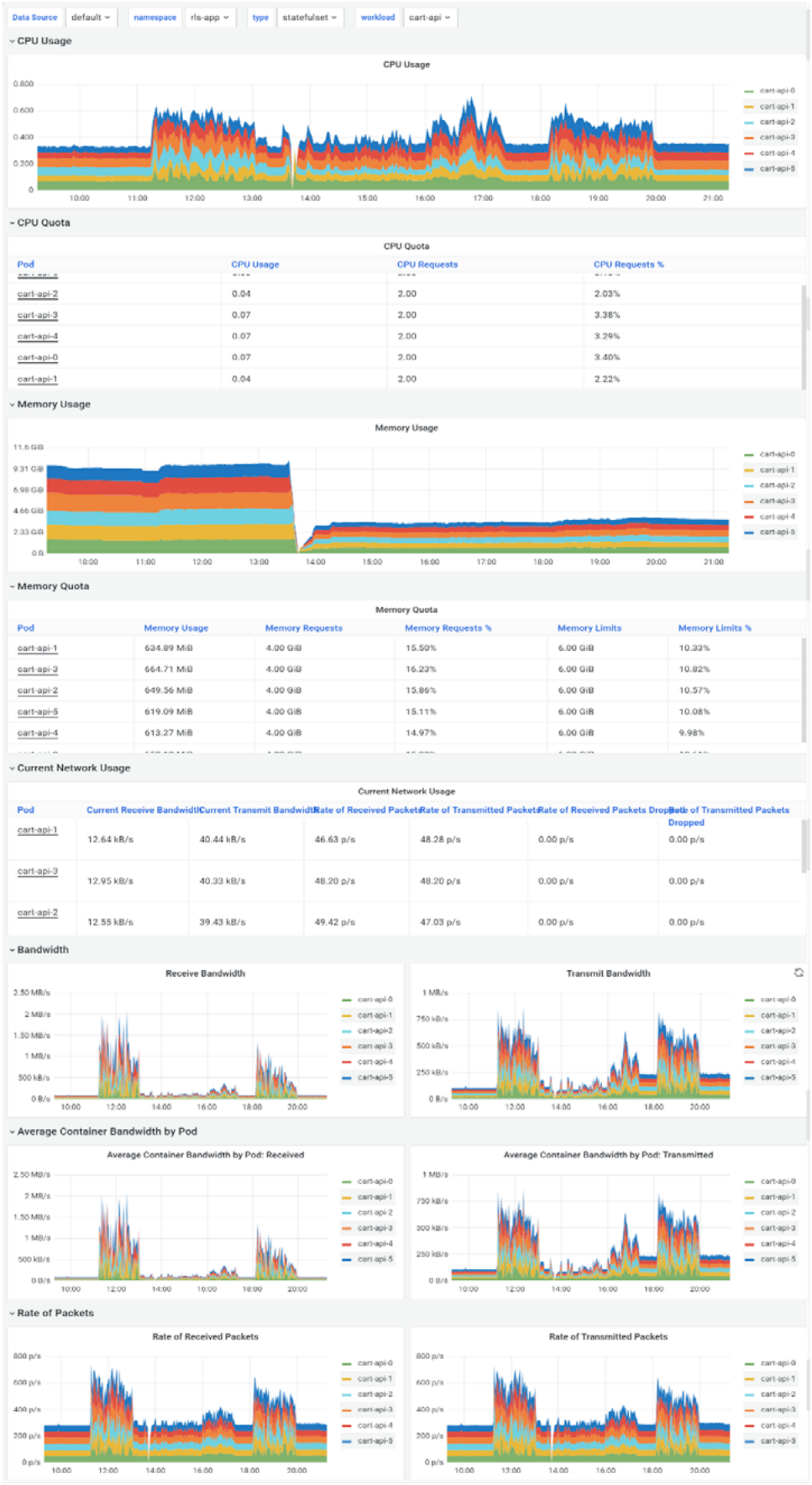
*** Only a subset of actions shown

Another SaaS Software customer's solution was rigorously tested with complex scenarios, including intricate bundle products and custom discount logic, for quotations ranging from 50 to 250 lines. Performance was continuously benchmarked throughout the project lifecycle in the Performance Lab, ensuring smooth operation at scale upon deployment and eliminating performance surprises in production.

5 Telemetry everywhere

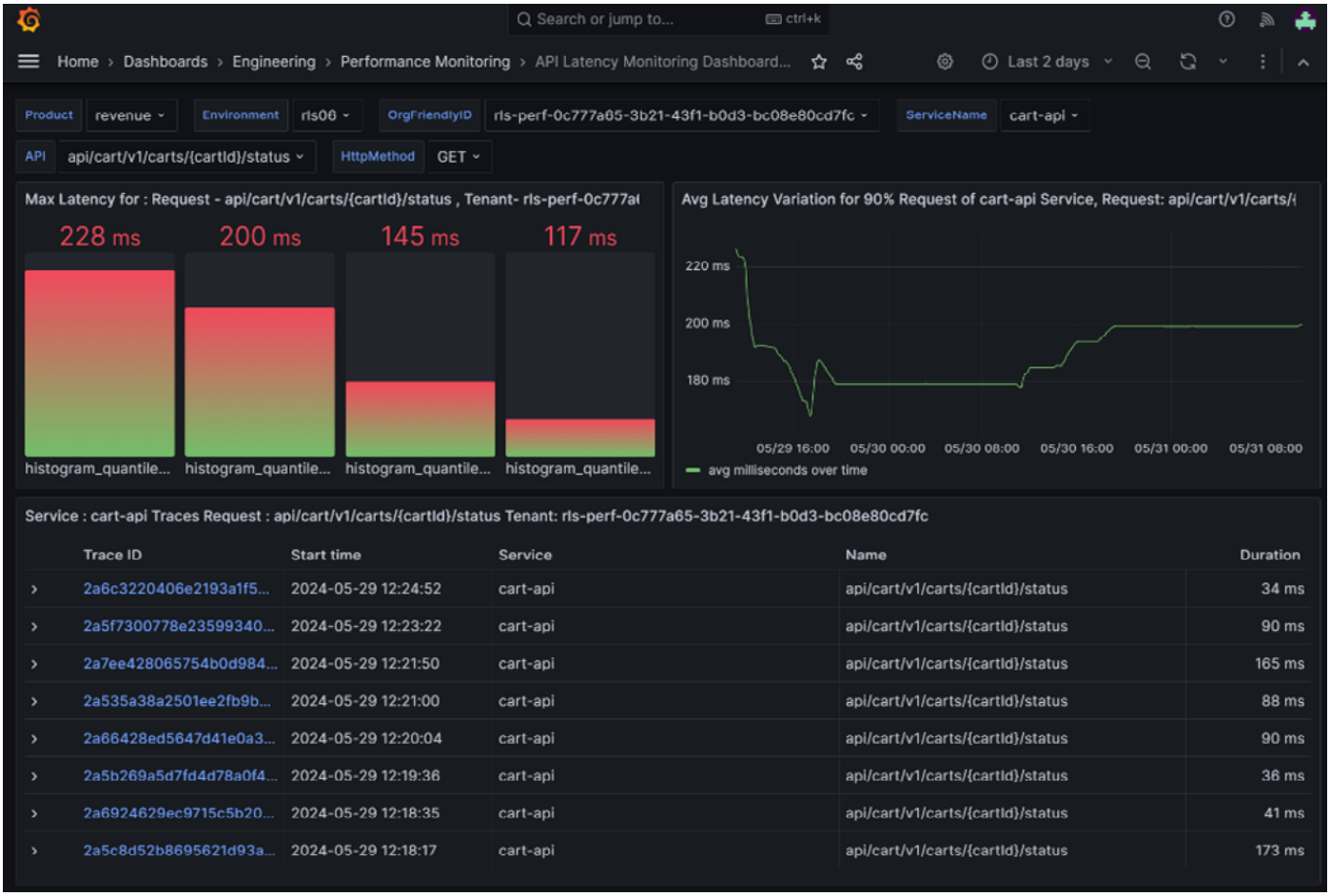
CPQ is built on industry-leading open standards of Open Telemetry. Conga CPQ microservices and platform services emit metrics and signals that are logged to these standards so that they are portable and can be visualized from various tools that support these standards.

Below is an illustration of a performance benchmark showcasing the CPU, Memory, Network bandwidth metrics in Grafana.



Telemetry – CPU, Memory metrics during a benchmark

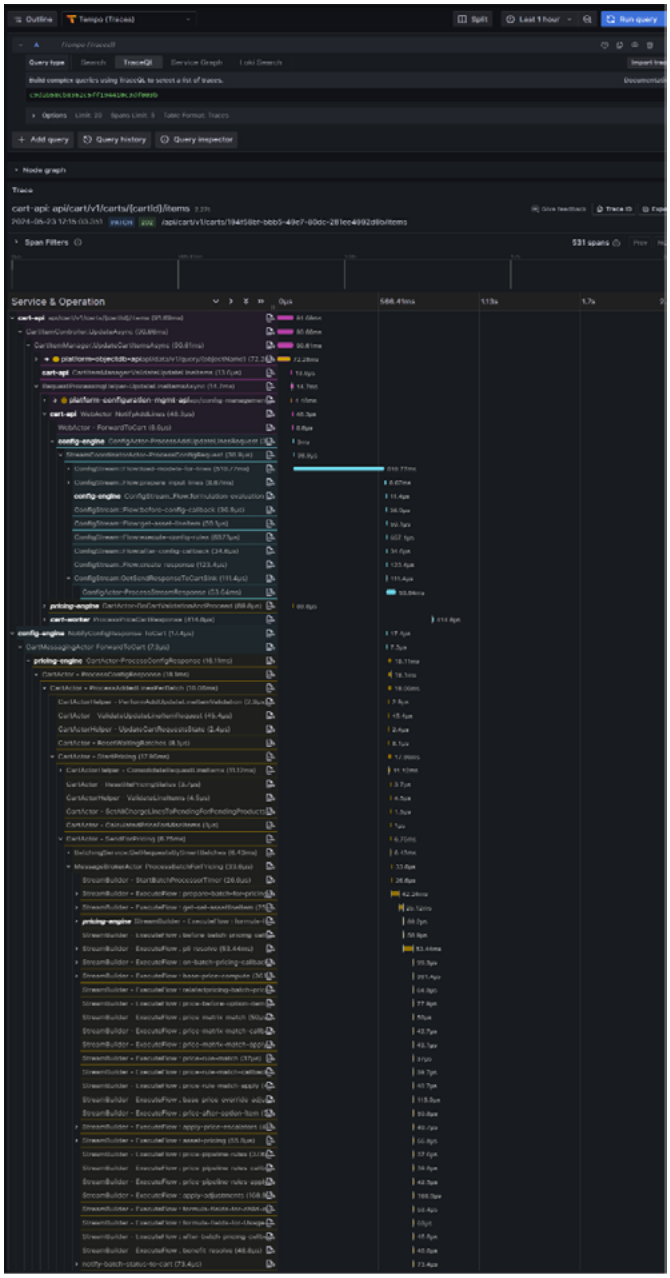
Also, we emit open telemetry signals so that each test can be stored, visualized, and shared with the engineering teams and customers. A performance test Grafana dashboard illustration is shown below.



Grafana dashboard showing a performance run

5.1 Distributed tracing

CPQ processing is also captured through a detailed distributed trace, that logs the flow path, time duration for each path and any associated logs. This adheres to the OpenTracing and OpenTelemetry standard so that its portable and can be visualized from various tools that support these standards and enable performance and functional issues to be easily resolved. Below is an illustration of CPQ trace.



Open tracing snippet

6 Conclusion

The Conga Advantage Platform offers a powerful and scalable platform for managing revenue processes. Conga CPQ, with its robust architecture and comprehensive large cart performance with testing and benchmarking by the performance team with industry-leading tools, methodologies, telemetry, and observability features, businesses can leverage Conga CPQ to drive growth and optimize their revenue management processes. The platform's roadmap includes continuous enhancements and innovations to meet evolving business needs.

About Conga

Conga, the Revenue Company, is the pioneer and market leader in Revenue Lifecycle Management. Its platform is chosen by the world's growth champions to accelerate the end-to-end revenue lifecycle and achieve a Revenue Advantage. Conga brings Configure, Price, Quote, Contract Lifecycle Management, and Document Automation capabilities together on a single open platform that works with any ERP, any CRM, and any Cloud. Conga is born for the top line—powered by a unified revenue data model, complete revenue intelligence, and purpose-built AI—to help companies grow, protect, and expand their revenue.

Conga delivers a Revenue Advantage to over 10,000 customers and 6.4 million users around the world. More than 7 million contracts and 46 million quotes are generated annually with Conga. Founded in 2006, the company is headquartered in Broomfield, CO and has offices across the United States, India, and Ireland. Visit conga.com for more information.

For more information

Email info@conga.com or visit conga.com