October 2023

BLUE GREEN DEPLOYMENT STRATEGY FOR APPLICATIONS USING KAFKA

Rinkle Seth

Pragya Priyadarshini

Rushil Sharma

Vrinda Rastogi

Yash Dwivedi

See next page for additional authors

Inventor(s)
Rinkle Seth, Pragya Priyadarshini, Rushil Sharma, Vrinda Rastogi, Yash Dwivedi, and Asheesh Gupta

“BLUE GREEN DEPLOYMENT STRATEGY FOR APPLICATIONS USING KAFKA”

VISA

INVENTOR: **Seth Rinkle**
Pragya Priyadarshini
Rushil Sharma
Vrinda Rastogi
Yash Dwivedi
Asheesh Gupta

TECHNICAL FIELD

[0001] The present invention is, in general, related to Distributed Platforms. In particular, the subject matter relates to the Blue Green Deployment Strategy for applications using Kafka.

BACKGROUND

[0002] The following description includes information that may be useful in understanding the present invention. It is not an admission that any of the information provided herein is prior art or relevant to the presently claimed invention, or that any publication specifically or implicitly referenced is prior art.

[0003] Software Development involves various phases a development phase, a testing phase, a release phase, an activation phase, a deployment phase and an updation phase among others. In today's dynamic world, most software needs frequent updates to introduce new functionalities and to address the issues present in the earlier version so as to meet the customer's expectations. Whenever any change or updation is carried out in the software application by the developers to introduce new functionalities or to upgrade the existing functionalities, testing is required before actual deployment of the new or updated version of the software application. The testing may be performed either on a subset of users or traffic so as to verify the performance of the introduced functionality as well as to monitor its stability of the operation. Post verification of the correctness of the intended functionality or performance, the earlier version of the software application may be replaced with a newer/updated version of the software application.

[0004] Blue-Green Deployment is one such deployment technique that may be used for software deployment in which a new version is released without impacting the current live version. This deployment technique creates two identical environments, one is "blue" environment running the live traffic and the other is a "green" environment which is the new version/updated version of the software application.

[0005] In situations when a Kafka client/service application needs to be deployed using a blue-green deployment technique, the consumers in a green environment may not be able to consume data from the Kafka topic or cluster since there are no Kafka topic partitions available in the green environment for consumers to consume/read/subscribe leading to

a situation where the new version can't be validated or tested in a green environment. If the testing is not done, then the application cannot be deployed using the blue-green deployment model.

[0006] Hence there is a need for a method and system that overcomes the limitations of the existing mechanisms and facilitates testing and validation of Kafka client/service applications in the green environment before making it fully operational.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate exemplary embodiments and, together with the description, explain the disclosed principles. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference features and components. Some embodiments of device or system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and with reference to the accompanying figures, in which:

[0008] Figure 1 illustrates an exemplary block diagram illustrating a Kafka system architecture 100, in accordance with an embodiment of the present disclosure.

[0009] Figure 2 illustrates an exemplary block diagram illustrating a system architecture of blue green deployment model 200, in accordance with an embodiment of the present disclosure.

[0010] Figure 3 illustrates a configuration of Kafka consumers 302 and topic partitions 304 in a blue (live environment), in accordance with an embodiment of the present disclosure.

[0011] Figure 4 illustrates a configuration of consumers and topic partitions in both the blue and green environment, in accordance with an embodiment of the present disclosure.

[0012] Figure 5 illustrates a configuration of consumers and topic partitions in a green environment, in accordance with an embodiment of the present disclosure.

[0013] Figure 6 illustrates a flowchart of an exemplary method 600 for implementing blue green deployment model for Kafka client applications, in accordance with an embodiment of the present disclosure.

[0014] The figures depict embodiments of the disclosure for purposes of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the disclosure described herein.

DESCRIPTION OF THE DISCLOSURE

[0015] It is to be understood that the present disclosure may assume various alternative variations and step sequences, except where expressly specified to the contrary. It is also to be understood that the specific devices and processes illustrated in the attached drawings and described in the following specification are simply exemplary and non-limiting embodiments or aspects. Hence, specific dimensions and other physical characteristics related to the embodiments or aspects disclosed herein are not to be considered as limiting.

[0016] In the present document, the word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any embodiment or implementation of the present subject matter described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other embodiments.

[0017] While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however, that it is not intended to limit the disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternative falling within the spirit and the scope of the disclosure.

[0018] The terms “comprise”, “comprising”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device, or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a device or system or

apparatus proceeded by “comprises” does not, without more constraints, preclude the existence of other elements or additional elements in the device or system or apparatus.

[0019] The terms “an embodiment”, “embodiment”, “embodiments”, “the embodiment”, “the embodiments”, “one or more embodiments”, “some embodiments”, and “one embodiment” mean “one or more (but not all) embodiments of the invention(s)” unless expressly specified otherwise.

[0020] When a Kafka Client application is deployed using a blue-green deployment approach the consumers in the green environment are not able to consume data from the Kafka topic since there are no Kafka topic partitions available in the green environment for consumers to consume due to which testing/validation of the new version cannot be completed. Due to a lack of testing, live traffic cannot be switched from a blue environment to a green environment and thus newer versions of the application cannot be deployed using the blue-green deployment approach. There are some solutions available for this problem such as KAFKA Parallel stacks, Mirror-maker, consumer-offset management and zookeeper etc. However, these existing technique are either quite complex, needs code change or are costly. Thus, a solution is required that is simple, economical, ensure minimum rebalancing time when green consumers are launched / blue consumers are shutdown. Also, it should avoid complete blackout of the cluster where no data is being consumed.

[0021] To address the same, the present disclosure provides a partition configuration that may rebalance the blue-green consumers in a certain way that ensures green application consumers to consume messages from Kafka topic/partitions, enabling development teams to test the updated/newer version of applications in the green environment before completely switching of live traffic/consumers to the green environment. A detailed explanation of the proposed solution is disclosed in the forthcoming paragraphs.

[0022] **Fig 1** illustrates an exemplary block diagram illustrating a Kafka system architecture 100 comprising of Producers 102, Consumers 106, and a Kafka Server 104. Producers may be the client applications or processes responsible for writing/publishing data to topics in Kafka Server 104. Consumers 106 are applications or processes that subscribe to reading and process those topics. Typically, Consumers are provided with an option

of reading messages starting at a certain offset or from any offset point they desire. As a result, consumers may join the Kafka Clusters at any moment. In Kafka, there are two categories of Consumers. The first is the Low-Level Consumer, which specifies Topics and Partitions as well as the offset from which to read, which can be either fixed or variable. Other is High-Level Consumer (or Consumer groups), which consist of one or more Consumers. Topics are categories in which records are published by producers and from which records are consumed by Consumers. Each topic is subdivided into partitions which are basic units for parallel processing and distribution in Kafka. Each partition is an independent sub-stream of data within a topic. Producers 102 may write records to different partitions of the same topic concurrently, and consumers 106 can read from different partitions concurrently. However, it may be noted that the exemplary Kafka architecture is not limited to only the above configuration and may involve additional configurations as per use basis.

[0023] **Fig 2** illustrates an exemplary block diagram illustrating a system architecture of blue green deployment model 200 comprising two separate environments blue (live environment) 202 and green (test environment) 204. The blue environment 202 is the current production environment that serves user traffic 212. The green environment is a duplicate of the (live) blue environment, it may host another copy/newer version of the application that is live or being deployed in blue environment at later stage. The copy of the newer version in green environment is isolated from user traffic 212 until it is thoroughly tested and ready to become live. Both blue and green environment use shared resources 206 like Database 208, Services 210 etc. An important concept in blue green deployment is of user traffic 212 switching once the green environment has been tested and validated for updated changes. The switching of user traffic is done either using load balancer or switch which is shown via 214 in Fig 2 .

[0024] Now referring back to Fig 1 and 2, when traffic is switched from the blue 202 environment in which the Kafka Consumer application is running, to green environment 204 the consumers 106 may face challenges in consuming data from Kafka partitions 104 in the green environment due to absence of same sets of Kafka topics and topic partitions in both environments. Consumers 106 in the green environment should be configured to read from the Kafka topics with the correct partition assignment. This

may involve adjusting the consumer's configuration and topic partition 104 to match the topic and partition assignment of the green environment.

[0025] **Fig 3** illustrates a configuration of Kafka consumers 302 and topic partitions 304 in blue (live environment). Certain rules of Kafka architecture apply here, as all consumers 1-3 in a live environment belong to the same consumer group. In Kafka, a consumer group is a logical grouping of Kafka consumers that work together to consume messages from one or more Kafka topics. Within a consumer group, each consumer is responsible for reading messages from one or more partitions of the subscribed topics. Kafka automatically distributes partitions evenly among the consumers within a consumer group. This load balancing ensures that each consumer gets an approximately equal share of the workload, promoting efficient resource utilization.

[0026] Referring back to Fig 3, the initial configuration in live environment 302 comprises three consumers, consumer 1, 2, and 3 (302) consuming data from various topic partitions 304, wherein consumer 1 is consuming data from partitions P1 and P2, consumer 2 is consuming data from partitions P3 and P4, and consumer 3 is consuming data from partitions P5 and P6. It may be noted that the initial environment is blue so consumers in the blue environment are active and are able to consume messages. It may be noted by a skilled person that topic partition 304 depicted in Fig. 3 may be analogous to topic partition 104 depicted in Fig. 1. Now if there is requirement of introduction of green environment for launching an updated or new version of application. For launching the green environment, the present disclosure needs to ensure minimum rebalancing time when green consumers are launched / blue consumers are shutdown. Therefore, there is a requirement to devise Partition Strategies to reduce the time required in rebalancing. Also, it should be kept in the mind that a complete blackout of the cluster should not be there i.e., focus should be on avoiding the complete blackout of the cluster where no data is being consumed.

[0027] Present disclosure takes care of both the above-mentioned requirements by suggesting a rebalancing strategy where number of the partitions should always be kept either twice of the number of consumers or greater than twice of the number of consumers to ensure that all active consumers are able to consume data from Kafka topic partitions when both environments are active. This may also help in avoiding the complete blackout of

the cluster as there must be few consumers present in both the environment and that would be shut down one by one once it is verified that green environment is good to go for final deployment. Same is explained in detail in figure 4.

[0028] **Fig 4** illustrates a configuration of consumers and topic partitions 406 deployed in both the blue and green environment, in which the newer/updated version of the application is deployed to the green environment 404 and live version is deployed to the blue environment 402. Considering the earlier scenario, where only blue environment was present (as depicted in figure 3), there were only three consumers and each of them was using two partitions each. Now, since a green environment has been introduced thus, it may require consumers for testing as well. To cater to this requirement, rebalancing of the partitions is required. Therefore, as proposed in above paragraph the rebalancing is performed in such a way that number of the partition should always be kept either twice of the number of consumers or greater than twice of the number of consumers to ensure that all active consumers are able to consume data from Kafka topic partitions when both environments are active. Since there are three consumers present earlier therefore, either 2* number of consumers or more than 2* number of consumers partitions are required. Considering the current case where there are 3 consumers are present (depicted in figure 3). Now, 6 or more than 6 partitions are required so that all active consumers (1,2,3) are able to consume data from Kafka topic partitions when both environments are active. Once rebalancing is performed then in both environments, consumers are now active. The Consumers 4-6 in green environment are able to consume data from partition 406 based on the partition P2, P4 and P6 assigned in the green environment 404 along with the consumers 1-3 present in the blue environment. It may be noted that for the purpose of depicting the configuration of partitions in figures, a single producer publishing data to topic with 6 partitions, along with 3 consumers each in both environments is assumed.

[0029] The rebalancing of partitions in a configuration mainly ensures each consumer in green environment 404 is able to consume data from at least one partition. The concept of partition rebalancing in Kafka is a process through which Kafka automatically redistributes partitions of a topic among the consumers within a consumer group, whenever a new consumer joins, an existing consumer dies/leaves or becomes unreachable etc. The present disclosure proposes a configuration of partitions 406 in

such a way that some of the partitions assigned to consumers 1-3 in the blue environment earlier (acc. to figure 3) should be reassigned to consumers 4-6 in green environment 404. Particularly, Figure 4, illustrates an assignment strategy where consumers 1, 2, and 3 are active in the blue environment and consumers 4, 5 and 6 are active in green environment. As shown in Figure 3, consumer 1 was consuming from P1, P2, consumer 2 was consuming from P3, P4 and consumer 3 was consuming from P5 and P6, when the consumers 4, 5 and 6 are introduced in green environment 404, there was no partition available to them for consuming data, therefore there was rebalancing of partition performed in a way that P2 is now assigned to the consumer 4, P4 is assigned to consumer 5 and P6 is assigned to consumer 6 as shown in figure 4. Further, it may be noted that any of the existing assignment strategies like ranger assignor or cooperative sticky assignor may be applied for the assignment of partition to consumers in both blue and green environment. The cooperative sticky assignor is shown in figures 4 and 5 for the purpose of showcasing the partition configurations. It may be noted by a skilled person that the topic partitions 406 depicted in Fig. 4 may be analogous to topic partitions 104 depicted in Fig. 1. Further, the blue environment 402 and green environment 404 depicted in Fig. 4 may be analogous to topic blue and green environment depicted in fig 2,3 and 5.

[0030] **Fig 5** illustrates a configuration of consumers 4-6 and topic partitions 504 in a green environment 502. The green environment is tested/validated with consumers 4-6 which are now able to consume data from partitions P2, P4 and P6 initially and partitions P1, P3 and P5 are used by consumers 1-3. In a situation when everything works fine in green environment 502 with respect to application usability, consumers 1-3 in the blue environment (402, Figure 4) are shut down one by one (to avoid complete blackout) and Partitions P1, P3 and P5 are re-assigned to consumers 4-6 in green environment 502 as shown in figure 5.

[0031] Further before shutting down consumers 1-3 in blue environment 402, make sure that consumers in “blue” environment commit their current offsets to Kafka Server, this step is crucial to maintain message processing continuity in the green environment.

[0032] Further in case, there is any sort of instability, performance bugs, downtime, or issue in user access among others in the green test environment 502, then the consumers in

green environment can be rolled back to the blue environment 402 and there will be no downtime during deploying minor updates and newer release. Table 1 depicts assignment of consumers for different partitions based on different strategies in accordance with an embodiment of the present disclosure. For example, initially, in the blue environment Consumer 1 is assigned with partitions 1,2, Consumer 2 is assigned with partitions 3 and 4, and Consumer 3 is assigned with partitions 5 and 6. Post introduction of the green environment along with the blue environment, results for different partition strategies such as ranger assignor and Cooperative Sticky Assignor is provided in below table 1:

Partition Strategy	**Initial Configuration (Blue Environment) Figure 3**	**Configuration (when both blue and green consumers are active) Figure 4**	**Final Configuration (Green Environment) Figure 5**
Ranger Assignor	C1-> P1, P2 C2-> P3, P4 C3-> P5, P6	C1-> P1 C2-> P2 C3-> P3 C4-> P4 C5-> P5 C6-> P6	C4-> P1, P2 C5-> P3, P4 C6-> P5, P6
Cooperative Sticky Assignor	C1-> P1, P2 C2-> P3, P4 C3-> P5, P6	C1-> P1 C2-> P3 C3-> P5 C4-> P2 C5-> P4 C6-> P6	C4-> P1, P2 C5-> P3, P4 C6-> P5, P6

Table 1

[0033] Fig 6 illustrates a flowchart of an exemplary method 600 for implementing blue green deployment model for Kafka client applications. The method 600 may also be described

in the general context of computer-executable instructions. Generally, computer-executable instructions may include routines, programs, objects, components, data structures, procedures, modules, and functions, which perform specific functions or implement specific abstract data types.

[0034] The order in which the method 600 is described is not intended to be construed as a limitation, and any number of the described method blocks may be combined in any order to implement the method. Additionally, individual blocks may be deleted from the methods without departing from the spirit and scope of the subject matter described.

[0035] At step 602, the method 600 includes assigning one or more consumers active in blue environment with two or more partitions in blue environment. In an exemplary embodiment, consumers 1, 2 and 3 which are active in blue environment are assigned Partitions P1-P6 as per one of the assignment strategies of Kafka architecture namely either as per ranger assignor or cooperative sticky assignor. Particularly, consumer 1 is assigned to P1 and P2, consumer 2 is assigned to P3 and P4, and consumer 3 is assigned to P5 and P6. The respective assignment of consumer 1-3 in blue environment 302 is also shown in table 1 as mentioned above.

[0036] At step 604, the method 600 determines whether a new version or update is available for the service application so as to confirm deployment of a new version in the green environment. If an update or new version is deployed then at step 606, the method performs rebalancing of load operation in such a manner that the number of the partition should always be kept either twice of the number of consumers or greater than twice of the number of consumers to ensure that all active consumers are able to consume data from Kafka topic partitions when both environments are active. However, if no update or new version is available then there is no change required in the existing environment and no action is performed as mentioned in step 608.

[0037] At step 610, the system verifies whether the intended functionality provided through the updates or new version is meeting the expectations or not. For example, consumers 4-6 now consume data from their respective assigned partitions and provide feedback during testing and validation purposes in green test environment.

[0038] If the expectations are met then at step 612, the blue environment is shut down for consumers one by one. In the exemplary scenario, Consumers 1-3 consuming data from P1, P3 and P5 are shut down one by one in case there is no issue faced with respect to application usability in green environment and green environment indicates expectations are met. Thereafter, consumer 4 is also assigned P1, consumer 5 is also assigned P3, and consumer 6 is also assigned P5. Each of the respective consumer 4-6 now has two partitions available to consume data. The whole traffic is switched to green environment and now green environment is now made the live version with the help of Load Balancers/DNS Switchers 214. In particular, the green environment of current case may become blue environment for traffic.

[0039] Else if the expectations are not met or the issue in the earlier version is not fixed by the updates then at step 614 of method 600, rollback to the blue environment takes place and consumers from green environment are shut down one by one. Thereafter, consumer 1 is also assigned partition P2, consumer 2 is also assigned P4, and consumer 3 is also assigned P6. Each of the respective consumer 1-3 now again has two partitions available to consume data. The whole traffic is switched to a blue environment once again.

[0040] The illustrated method steps are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed.

[0041] Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments.

[0042] Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer-

readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer-readable storage medium may store instructions for execution by one or more processors, including instructions for causing the processor(s) to perform steps or stages consistent with the embodiments described herein. The term “computer- readable medium” should be understood to include tangible items and exclude carrier waves and transient signals, i.e., are non-transitory. Examples include random access memory (RAM), read-only memory (ROM), volatile memory, non-volatile memory, hard drives, CD ROMs, DVDs, flash drives, disks, and any other known physical storage media.

[0043] Suitable processors include, by way of example, a general-purpose processor, a special purpose processor, a conventional processor, a digital signal processor (DSP), a graphic processing unit (GPU), a plurality of microprocessors, one or more microprocessors in association with a DSP core, a controller, a microcontroller, Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs) circuits, any other type of integrated circuit (IC), and/or a state machine.

[0044] Advantages of the disclosed technique:

- Easy to implement by mere a configuration change.
- Avoiding the complete blackout of the cluster as there must be few consumers present in both the environment
- Very less rebalancing time required
- No additional cost is required.
- No code change process is required.
- Deploying new version without data loss.

ABSTRACT

BLUE GREEN DEPLOYMENT STRATEGY FOR APPLICATIONS USING KAFKA

Disclosed herein is a blue-green deployment strategy for consumer applications using the Kafka system. The present disclosure proposes setting producer and consumer configurations in a way that green application consumers are able to consume messages from Kafka topic, hence enabling application teams to test their changes in green environment. To achieve this, present disclosure provides rebalancing of partitions technique where the number of partitions are configured either twice the number of consumers or greater than twice the number of consumers so that all consumers in a consumer group are active and are able to consume data from Kafka topic.

FIG 6

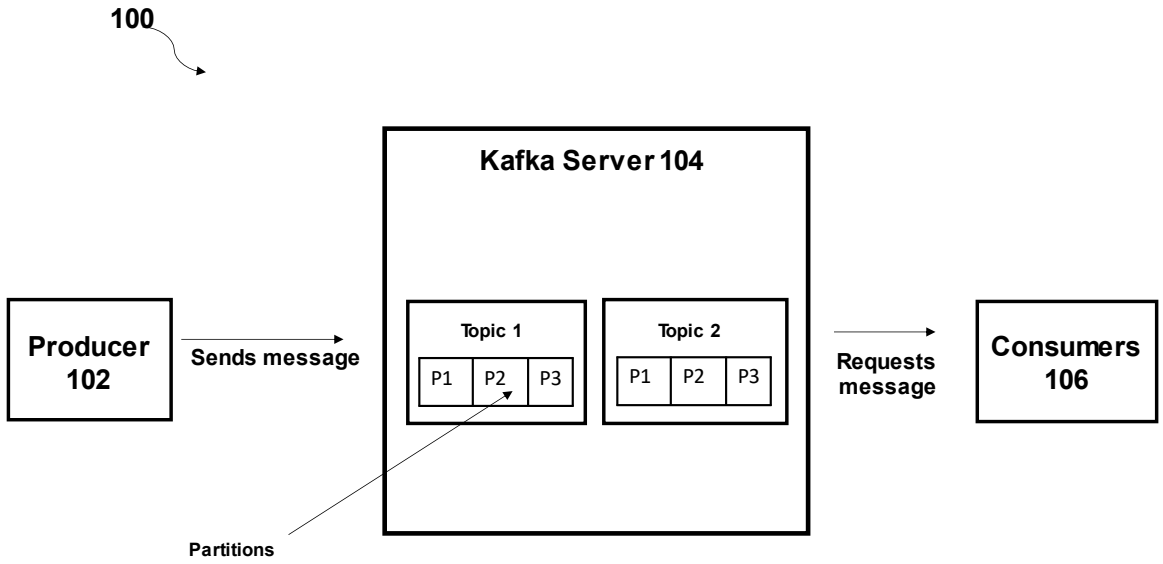
100
Kafka Server 104
Producer 102
Sends message
Topic 1
P1
P2
P3
Topic 2
P1
P2
P3
Requests message
Consumers 106
Partitions

Fig 1.

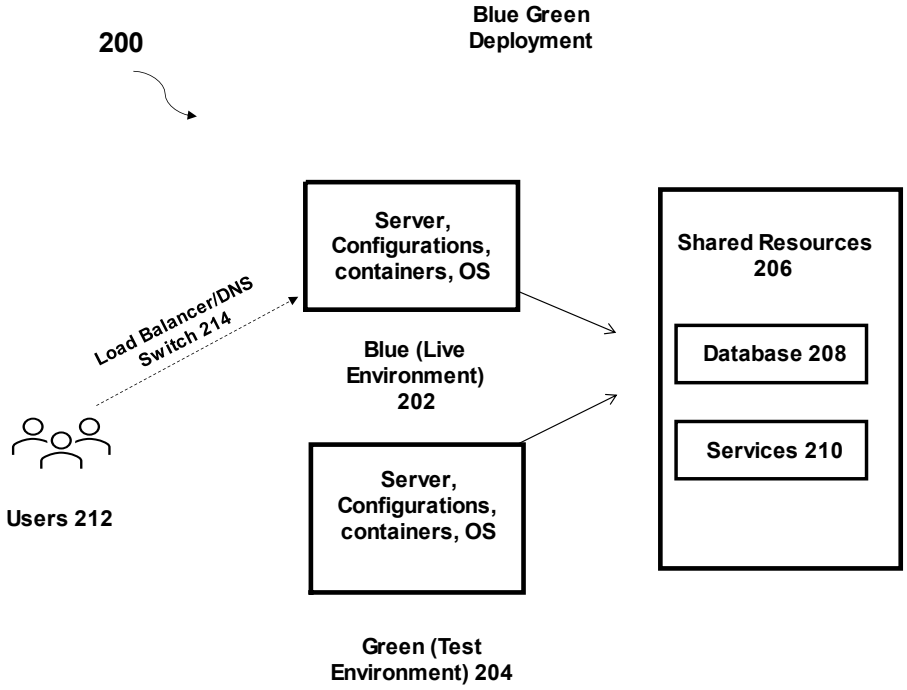
Blue Green Deployment
200
Server, Configurations, containers, OS
Load Balancer/DNS Switch 214
Blue (Live Environment) 202
Shared Resources 206
Database 208
Services 210
Users 212
Server, Configurations, containers, OS
Green (Test Environment) 204

Fig 2.

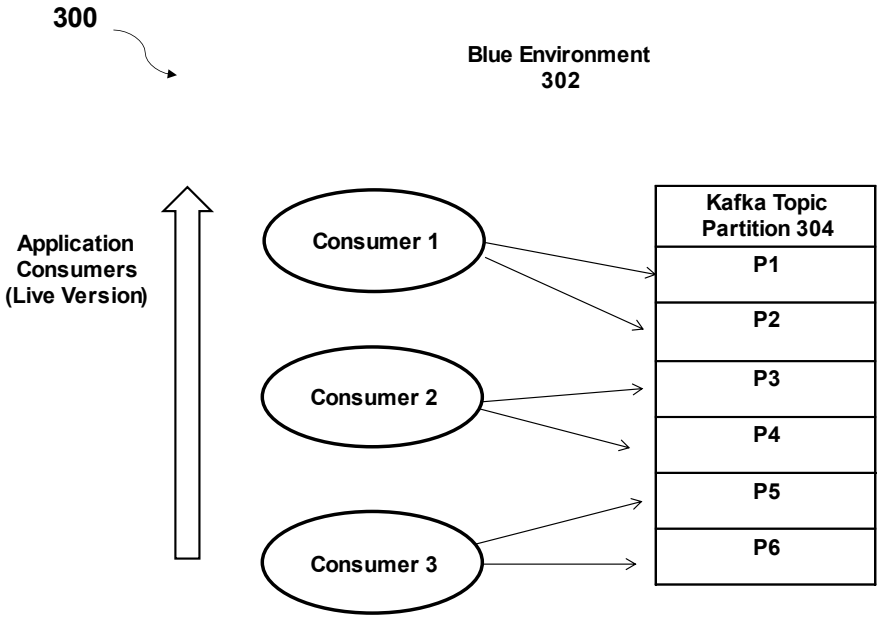
300
Blue Environment
302
Application
Consumers
(Live Version)
Consumer 1
Consumer 2
Consumer 3
Kafka Topic
Partition 304
P1
P2
P3
P4
P5
P6

Fig 3.

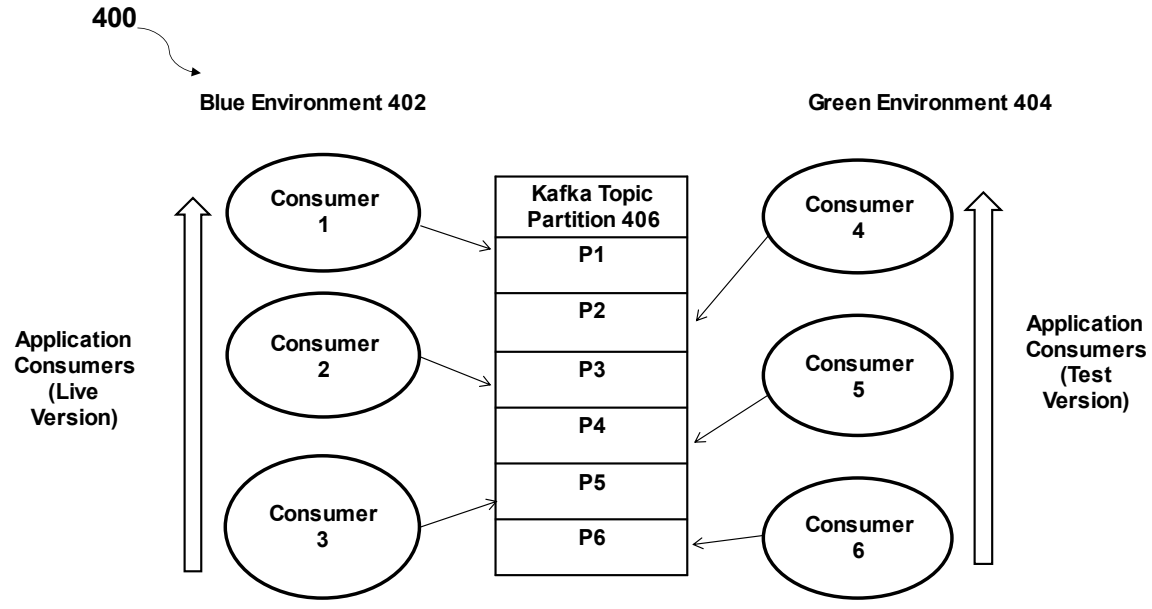
400
Blue Environment 402
Green Environment 404
Application
Consumers
(Live
Version)
Consumer
1
Consumer
2
Consumer
3
Kafka Topic
Partition 406
P1
P2
P3
P4
P5
P6
Consumer
4
Consumer
5
Consumer
6
Application
Consumers
(Test
Version)

Fig 4.

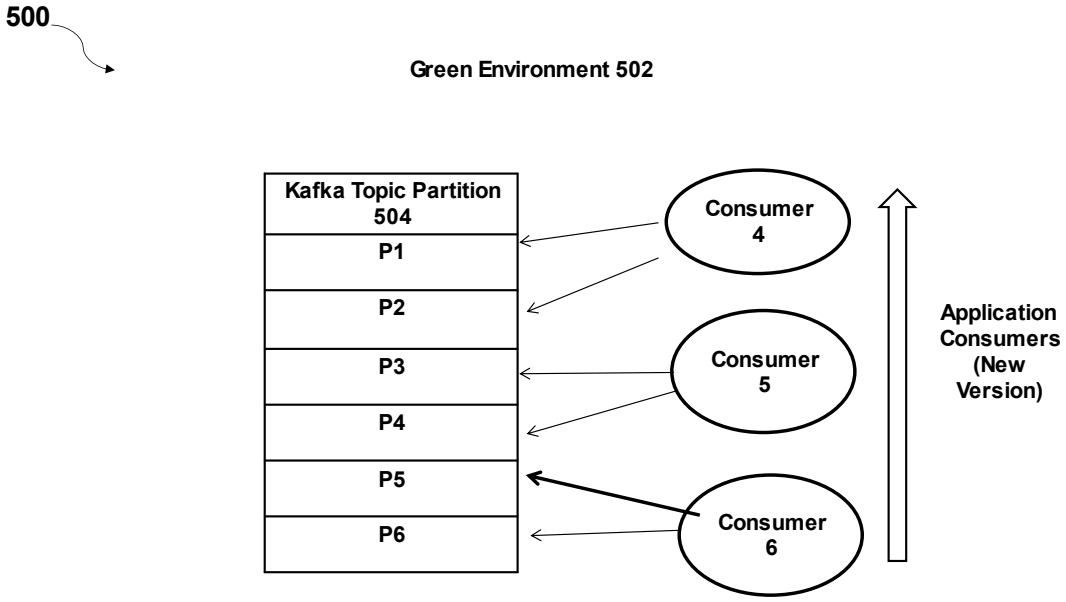
500
Green Environment 502
Kafka Topic Partition 504
P1
P2
P3
P4
P5
P6
Consumer 4
Consumer 5
Consumer 6
Application Consumers (New Version)

Fig 5.

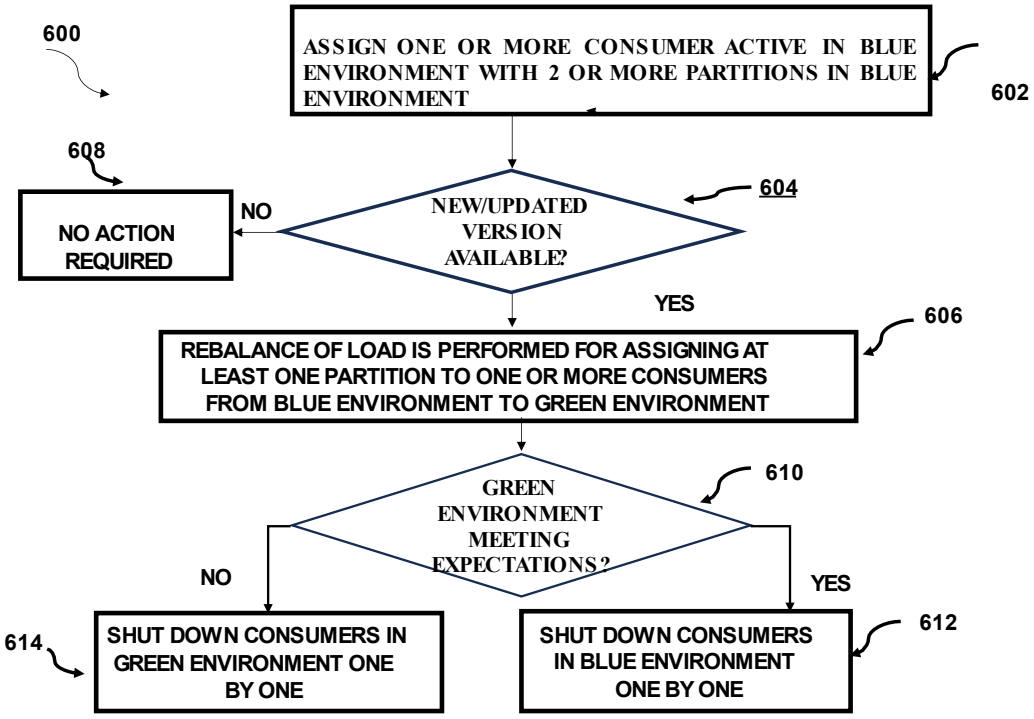
600
ASSIGN ONE OR MORE CONSUMER ACTIVE IN BLUE ENVIRONMENT WITH 2 OR MORE PARTITIONS IN BLUE ENVIRONMENT
602
608
NO ACTION REQUIRED
NO
NEW/UPDATED VERSION AVAILABLE?
604
YES
606
REBALANCE OF LOAD IS PERFORMED FOR ASSIGNING AT LEAST ONE PARTITION TO ONE OR MORE CONSUMERS FROM BLUE ENVIRONMENT TO GREEN ENVIRONMENT
610
GREEN ENVIRONMENT MEETING EXPECTATIONS ?
NO
YES
614
SHUT DOWN CONSUMERS IN GREEN ENVIRONMENT ONE BY ONE
SHUT DOWN CONSUMERS IN BLUE ENVIRONMENT ONE BY ONE
612

Fig 6