



(51) International Patent Classification:

G06F 11/20 (2006.01) H04L 69/40 (2022.01)
H04L 41/0663 (2022.01)

(21) International Application Number:

PCT/IN2024/050930

(22) International Filing Date:

26 June 2024 (26.06.2024)

(25) Filing Language:

English

(26) Publication Language:

English

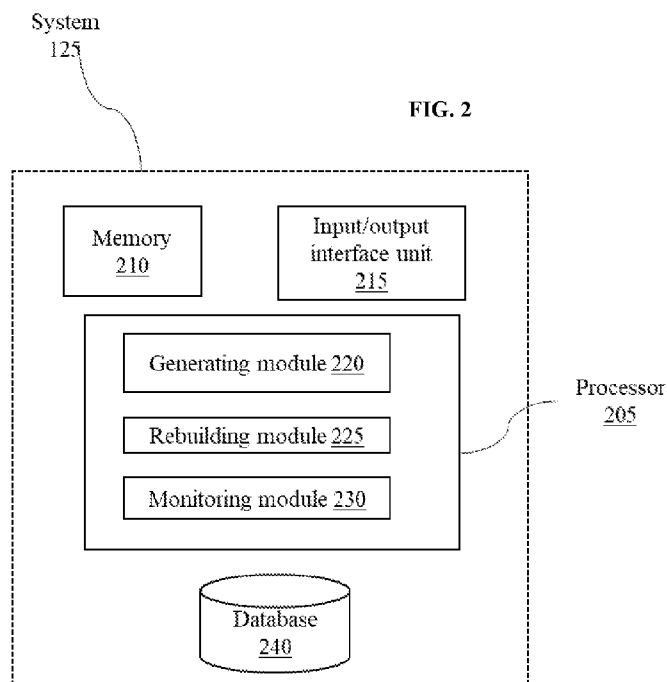
(30) Priority Data:

202321044334 03 July 2023 (03.07.2023) IN

(71) Applicant: **JIO PLATFORMS LIMITED** [IN/IN]; Of-
fice-101, Saffron, Nr. Centre Point, Panchwati 5 Rasta, Am-
bawadi, Gujarat, Ahmedabad 380006 (IN).

(72) Inventors: **BHATNAGAR, Aayush**; Tower-7, 15B,
Beverly Park, Sector-14, Koper Khairane, Maharashtra,
Navi Mumbai 400701 (IN). **BISHT, Birendra**; B-2101,
Yashaskaram CHS, Plot -39, Sector-27, Kharghar, Maha-
rashtra, Navi Mumbai 410210 (IN). **SINGH, Harbinder
Pal**; Wing B1, Flat No 402, Lakhani Suncoast, Sector 15,
CBD, Belapur, Maharashtra, Navi Mumbai 400614 (IN).
SOREN, Rohit; Flat-106, HNo-84, Sultanpur, Delhi, New
Delhi 110030 (IN). **SINGH, Priyanka**; E-802 RiverScape
CHS, Casa Rio, Palava City, Maharashtra, Dombivli East
421204 (IN). **AGGARWAL, Pravesh**; A-313, Raghubir
Nagar -New Delhi, New Delhi 110027 (IN). **SAHU, Bid-
hu**; 1702, E, RiverScape, CasaRio, Palava City, Maharas-
tra, Dombivli East 421204 (IN). **MALAV, Virendra**; A-
47, Singhawat Villa, Shubh Savera, Raipura, Near Raipura
Circle, Rajasthan, Kota 324004 (IN). **DAS, Raghav**; Flat
no 304, 3rd floor, Bhola paswan ji, house, West Mahatma

(54) Title: METHOD AND SYSTEM FOR SWITCHING FROM ACTIVE PROCESS TO STANDBY PROCESS



(57) **Abstract:** The present disclosure relates to a system (125) and a method (400) for switching from an active process to a standby process in a communication network (105). The system (125) includes a generating module (220) to generate the standby process by creating backup of data of the active process for each user of a plurality of users and for each service of a plurality of services at a backend database (240). The system (125) further includes a rebuilding module (225) to rebuild data for each of the active process failure services as per sequence. The system (125) further includes a monitoring module (230) to check a status of rebuilt data for each active process failure service. Thereby, the system (125) switches from the active process to the standby process in the communication network (105) in an optimized manner to service a new request and reduce fail over duration.



Gandhi, Nagar, Kaanti Factory Road, Near Indus Ind Bank,
Kankarbagh, Bihar, Patna 800026 (IN).

(74) **Agent: JAPHET, Chinthan;** K Law (Krishnamurthy and
Co.), 4th Floor, Prestige Takt, No 23, Kasturba Road Cross,
Bangalore 560 001 (IN).

(81) **Designated States** (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG,
KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY,
MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA,
NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO,
RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH,
TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS,
ZA, ZM, ZW.

(84) **Designated States** (*unless otherwise indicated, for every
kind of regional protection available*): ARIPO (BW, CV,
GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST,
SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ,
RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ,
DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT,
LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE,
SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN,
GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

METHOD AND SYSTEM FOR SWITCHING FROM ACTIVE PROCESS TO
STANDBY PROCESS

FIELD OF THE INVENTION

[0001] The present invention generally relates to communication networks, and more particularly relates to transition from a standby process to an active process in a communication network.

BACKGROUND OF THE INVENTION

[0002] In existing applications, there may be a number of reasons because of which an active process handling of a user in a communication network may fail such as a segment fault and the like.

[0003] When a protocol data unit (PDU) session is established, the application stores PDU states and other relevant information in PDU contexts maintained in a local cache. These contexts are used by the process to handle User Equipment (UE) related call flows. In the event of a failover, where active process restarts due to some issue or planned event, corresponding standby process takes over, it becomes crucial for the newly active process to be able to enable UE related call flows in the same manner as in the previous active process. There are two options for achieving this.

[0004] The first option is to restore all the PDU session contexts immediately after failover. However, this approach poses challenges as it could potentially overwhelm the system, leading to system hangs. Restoring all session contexts would take significant time and require occupying all the available CPUs, which would impact processing of the currently live traffic.

[0005] The second option is to restore the PDU session context only when it is required, for instance, on receiving of any network trigger. However, this approach introduces delays to start the user state machine and, in some cases, where no network trigger is received, the user state machine would not start, which could result in

missing the network handling based on the user state at the network node e.g. handling of charging triggers or data quota allocations for the affected PDU sessions.

[0006] At the moment when the active process fails, a lot of to-be-processed activities may be pending. Also, there are various timers running which are associated with the processes. Examples of processes are charging, registration, etc.

[0007] If there is a time gap between the take-over by the standby process after the active process failure happens, a lot of essential activities which need to be performed regularly and periodically for each and every subscriber may not happen such as sending trigger to Charging Function (CHF) node and the like. These activities are controlled by inbuilt timers for each activity. These activities may be pending when the standby process is turned to active process.

[0008] When the active process fails, the standby process takes over to maintain continuity of services for a user. In order to function as a replacement of the active process, the standby process needs to have all the data which was built during and within the active process. There are services and aspects associated with the active process and the standby process needs to ideally be able to map the aspects in the active process to function and provide for the failure of the active process and act as a standby process.

[0009] For example, for each user there may be a number of services and associated data which the user may have subscribed to such as messaging, media, etc. All this data related to the services needs to be transferred to the standby process for the standby process to function as a replacement for the active process.

[0010] Also, a stateful needs to be maintained and transferred between the active process and the standby process.

[0011] When the active process fails and the standby process takes over, the standby process may instantly try to rebuild the data. This puts a lot of loads on the system and

the system takes time for taking and servicing any new requests. As a result, there will be a delay in servicing new requests. Further, in case of a segment failure and the data needs to be rebuilt for a large group of users simultaneously, the system may get so overloaded that it may not be able to service any new requests and in such a case, the delay and the failure over duration is even greater. There may also be loss of data in case the system hangs.

[0012] Besides failure, when the standby process needs to take over, rebuilding of the data is also required when a request comes for a certain user or a certain number of specific users.

[0013] It is desired that there is minimum delay, and a continuity is maintained for services for users in the network when the standby process takes over from the active process in case of failures.

[0014] Therefore, there is a need for an advancement for a system and method that can overcome at least one of the above shortcomings, particularly to switch over from the active process to the standby process.

BRIEF SUMMARY OF THE INVENTION

[0015] One or more embodiments of the present disclosure provide a system and method for switching from an active process to a standby process in a communication network.

[0016] In one aspect of the present invention, a system for switching from an active process to a standby process in a communication network is disclosed. The system includes a generating module configured to generate, the standby process by creating backup of data of the active process for each user of a plurality of users and for each service of a plurality of services at a backend database, maintain a copy of a timer in the standby process for each service of the plurality of services in the active process, and arrange a plurality of active process failure services in a sequence upon failure of

active process of one or more services of the plurality of services in order to switch to the standby process. The system further includes a rebuilding module and a monitoring module. The rebuilding module is configured to rebuild data for each of the active process failure services as per the sequence. The monitoring module is configured to check a status of rebuilt data for each active process failure service.

[0017] The system is further configured to arrange the plurality of active process failure services in the sequence by assigning a priority value for each of the plurality of active process failure services based on a respective timer status. In one embodiment, the timer status is one of a timer of the specific active process failure service expires prior compared to the respective timers of the remaining plurality of active process failure services arranged in the sequence, and the timer of the specific active process failure service has already expired. The system is further configured to rebuild data for the active process failure service arranged foremost in the sequence. The system is further configured to delay the rebuilding of data for the remaining active process failure services until the active process failure service arranged foremost is built and thereafter rebuilding data for the remaining active process failure services as per the sequence, ensures that the load on the standby process is reduced, thereby enabling the active process to receive a new request and to reduce fail over duration.

[0018] In another aspect of the present invention, a method for switching from an active process to a standby process in a communication network is disclosed. The method includes the steps of generating the standby process by creating backup of data of the active process for each user of a plurality of users and for each service of a plurality of services at a backend database, maintaining a copy of a timer in the standby process for each service of the plurality of services in the active process, and arranging a plurality of active process failure services in a sequence upon failure of active process of one or more services of the plurality of services in order to switch to the standby process. The method further includes the step of rebuilding data for each of

the active process failure services as per the sequence. Thereafter the method includes the step of monitoring a status of the rebuilt data for each active process failure service.

[0019] The method further includes arranging the plurality of active process failure services in the sequence by assigning a priority value for each of the plurality of active process failure services based on a respective timer status. In one embodiment, the timer status is one of a timer of the specific active process failure service expires prior compared to the respective timers of the remaining plurality of active process failure services arranged in the sequence, and the timer of the specific active process failure service has already expired. The method further includes rebuilding data for the active process failure service arranged foremost in the sequence. The method further includes delaying the rebuilding of data for the remaining active process failure services until the active process failure service arranged foremost is built and thereafter rebuilding data for the remaining active process failure services as per the sequence. By doing so, the method is configured to reduce the load on the standby process, to enable the active process to receive the new request and to reduce fail over duration.

[0020] Other features and aspects of this invention will be apparent from the following description and the accompanying drawings. The features and advantages described in this summary and in the following detailed description are not all-inclusive, and particularly, many additional features and advantages will be apparent to one of ordinary skill in the relevant art, in view of the drawings, specification, and claims hereof. Moreover, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes and may not have been selected to delineate or circumscribe the inventive subject matter, resort to the claims being necessary to determine such inventive subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] The accompanying drawings, which are incorporated herein, and constitute a part of this disclosure, illustrate exemplary embodiments of the disclosed methods

and systems in which like reference numerals refer to the same parts throughout the different drawings. Components in the drawings are not necessarily to scale, emphasis instead being placed upon clearly illustrating the principles of the present disclosure. Some drawings may indicate the components using block diagrams and may not represent the internal circuitry of each component. It will be appreciated by those skilled in the art that disclosure of such drawings includes disclosure of electrical components, electronic components or circuitry commonly used to implement such components.

[0022] **FIG. 1** is an exemplary block diagram of an environment for switching from an active process to a standby process in a communication network, according to various embodiments of the present invention;

[0023] **FIG. 2** is a block diagram of a system for switching from the active process to the standby process in the communication network, according to various embodiments of the present system;

[0024] **FIG. 3** is a schematic representation of the present system of **FIG. 1** workflow, according to various embodiments of the present system;

[0025] **FIG. 4** shows a flow diagram of a method for switching from an active process to a standby process in a communication network, according to various embodiments of the present system;

[0026] **FIG. 5** shows a flow diagram of a method for arranging plurality of active process failure services in a sequence, according to various embodiments of the present system; and

[0027] **FIG. 6** shows a flow diagram of a method for rebuilding data for each active process failure service in the sequence, according to various embodiments of the present system.

[0028] The foregoing shall be more apparent from the following detailed description of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0029] Some embodiments of the present disclosure, illustrating all its features, will now be discussed in detail. It must also be noted that as used herein and in the appended claims, the singular forms “a”, “an” and “the” include plural references unless the context clearly dictates otherwise.

[0030] Various modifications to the embodiment will be readily apparent to those skilled in the art and the generic principles herein may be applied to other embodiments. However, one of ordinary skill in the art will readily recognize that the present disclosure including the definitions listed here below are not intended to be limited to the embodiments illustrated but is to be accorded the widest scope consistent with the principles and features described herein.

[0031] A person of ordinary skill in the art will readily ascertain that the illustrated steps detailed in the figures and here below 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. 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.

[0032] The present invention provides a solution/a mechanism called “lazy loading” to address the problems in the prior art. The lazy loading combines the need-based

restoration of PDU sessions with restoration at time intervals. In the need-based restoration, which is stateless restoration, when a trigger related to a PDU session is received, the process checks if the PDU session context is already restored in the local cache. If it is not, the context is fetched from the Session Data Layer (SDL).

[0033] When the need-based restoration mechanism is not triggered, a timely stateful restoration mechanism is triggered. To implement this mechanism, a timer associated with each UE is run in the active process and the same is checkpointed to the standby process. After failover, when the standby process becomes active, it restores these timers of the remaining period from when it was started in active process. When a timer times out, the process checks if the corresponding PDU session context is restored in the local cache. If not, it fetches the context from the SDL and restores it in the local cache.

[0034] By combining these two approaches, the lazy loading mechanism provides an optimized balance between immediate restoration and efficient utilization of system resources during failover scenarios.

[0035] As per various embodiments depicted, the present invention discloses the system and method for switching from an active process to a standby process to service a new request and to reduce failover duration. The failover duration is defined as the time required for initiating a backup.

[0036] In various embodiments, the present invention discloses the system and method to switch from an active process to standby process to rebuild data by using a backend database based on a timer. The system and method further utilize the rebuild data to facilitate reducing delay and failover duration.

[0037] Referring to **FIG. 1**, **FIG. 1** illustrates an exemplary block diagram of an environment **100** for switching from an active process to a standby process in a communication network **105**. The environment **100** includes a user equipment **110**. For the purpose of description and explanation, the description will be explained with

respect to one or more user equipments (UE) **110**, or to be more specific will be explained with respect to a first UE **110a**, a second UE **110b**, and a third UE **110c**, and should nowhere be construed as limiting the scope of the present disclosure.

[0038] In an embodiment, each of the first UE **110a**, the second UE **110b**, and the third UE **110c** is one of, but are not limited to, any electrical, electronic, electro-mechanical or an equipment and a combination of one or more of the above devices such as virtual reality (VR) devices, augmented reality (AR) devices, laptop, general-purpose computer, desktop, personal digital assistant, tablet computer, mainframe computer, or any other computing device.

[0039] Each of the first UE **110a**, the second UE **110b**, and the third UE **110c** is further configured to transmit a request from a user via an interface module to the communication network **105** to avail one or more services. In one embodiment, the one or more services include, but not limited to, accessing a server **115**, transmitting a request, rebuilding a data, and monitoring the rebuilt data via the communication network **105**.

[0040] The server **115** may include by way of example but not limitation, one or more of a standalone server, a server blade, a server rack, a bank of servers, a server farm, hardware supporting a part of a cloud service or system, a home server, hardware running a virtualized server, one or more processors executing code to function as a server, one or more machines performing server-side functionality as described herein, at least a portion of any of the above, some combination thereof. In an embodiment, the entity may include, but is not limited to, a vendor, a network operator, a company, an organization, a university, a lab facility, a business enterprise, a defence facility, or any other facility that provides content.

[0041] The communication network **105** includes, by way of example but not limitation, one or more of a wireless network, a wired network, an internet, an intranet, a public network, a private network, a packet-switched network, a circuit-switched

network, an ad hoc network, an infrastructure network, a Public-Switched Telephone Network (PSTN), a cable network, a cellular network, a satellite network, a fiber optic network, or some combination thereof. The communication network **105** may include, but is not limited to, a Third Generation (3G), a Fourth Generation (4G), a Fifth Generation (5G), a Sixth Generation (6G), a New Radio (NR), a Narrow Band Internet of Things (NB-IoT), an Open Radio Access Network (O-RAN), and the like.

[0042] The environment **100** further includes a system **125** communicably coupled to the server **115** and each of the first UE **110a**, the second UE **110b**, and the third UE **110c** via the communication network **105**. The system **125** is configured to switch from the active process to the standby process in the communication network **105**.

[0043] In various embodiments, the system **125** may be generic in nature and may be integrated with any application including a System Management Facility (SMF), an Access and Mobility Management Function (AMF), a Business Telephony Application Server (BTAS), a Converged Telephony Application Server (CTAS), any SIP (Session Initiation Protocol) Application Server which interacts with core Internet Protocol Multimedia Subsystem (IMS) on Industrial Control System (ISC) interface as defined by Third Generation Partnership Project (3GPP) to host a wide array of cloud telephony enterprise services, a System Information Blocks (SIB)/ and a Mobility Management Entity (MME).

[0044] The system **125** is further configured to employ Transmission Control Protocol (TCP) connection to identify any connection loss in the communication network **105** and thereby improving overall efficiency. The TCP connection is a communication standard enabling applications and the system **125** to exchange information over the communication network **105**.

[0045] Operational and construction features of the system **125** will be explained in detail with respect to the following figures.

[0046] Referring to **FIG. 2**, **FIG. 2** illustrates a block diagram of the system **125** for switching from the active process to the standby process in the communication network **105**, according to one or more embodiments of the present invention. The system **125** is adapted to be embedded within the server **115** or is embedded as an individual entity. However, for the purpose of description, the system **125** is described as an integral part of the server **115**, without deviating from the scope of the present disclosure.

[0047] As per the illustrated embodiment, the system **125** includes one or more processors **205**, a memory **210**, and an input/output interface unit **215**. The one or more processor **205**, hereinafter referred to as the processor **205** may be implemented as one or more microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries, single board computers, and/or any devices that manipulate signals based on operational instructions. As per the illustrated embodiment, the system **125** includes one or more processors **205**. However, it is to be noted that the system **125** may include multiple processors as per the requirement and without deviating from the scope of the present disclosure. Among other capabilities, the one or more processors **205** is configured to fetch and execute computer-readable instructions stored in the memory **210**. The memory **210** may be configured to store one or more computer-readable instructions or routines in a non-transitory computer-readable storage medium, which may be fetched and executed to create or share data packets over a network service. The memory **210** may include any non-transitory storage device including, for example, volatile memory such as RAM, or non-volatile memory such as EPROM, flash memory, and the like.

[0048] In an embodiment, the input/output (I/O) interface unit **215** includes a variety of interfaces, for example, interfaces for data input and output devices, referred to as Input/Output (I/O) devices, storage devices, and the like. The I/O interface unit **215** facilitates communication of the system **125**. In one embodiment, the I/O interface unit **215** provides a communication pathway for one or more components of the system

125. Examples of such components include, but are not limited to, the UE **110** and a backend database **240**.

[0049] The backend database **240** is one of, but is not limited to, one of a centralized database, a cloud-based database, a commercial database, an open-source database, a distributed database, an end-user database, a graphical database, a No-Structured Query Language (NoSQL) database, an object-oriented database, a personal database, an in-memory database, a document-based database, a time series database, a wide column database, a key value database, a search database, a cache databases, and so forth. The foregoing examples of the backend database **240** types are non-limiting and may not be mutually exclusive e.g., a database can be both commercial and cloud-based, or both relational and open-source, etc.

[0050] Further, the one or more processors **205**, in an embodiment, may be implemented as a combination of hardware and programming (for example, programmable instructions) to implement one or more functionalities of the one or more processors **205**. In the examples described herein, such combinations of hardware and programming may be implemented in several different ways. For example, the programming for the one or more processors **205** may be processor-executable instructions stored on a non-transitory machine-readable storage medium and the hardware for one or more processors **205** may comprise a processing resource (for example, one or more processors), to execute such instructions. In the present examples, the memory **210** may store instructions that, when executed by the processing resource, implement the one or more processors **205**. In such examples, the system **125** may comprise the memory **210** storing the instructions and the processing resource to execute the instructions, or the memory **210** may be separate but accessible to the system **125** and the processing resource. In other examples, the one or more processors **205** may be implemented by electronic circuitry.

[0051] In order for the system **125** to switch from the active process to the standby process in the communication network **105**, the processor **205** includes a generating

module **220**, a rebuilding module **225**, and a monitoring module **230** communicably coupled to each other.

[0052] The generating module **220** of the processor **205** is communicably connected to each of the first UE **110a**, the second UE **110b**, and the third UE **110c** via the communication network **105**. Accordingly, the generating module **220** is configured to generate the standby process by creating backup of data of the active process for each user of a plurality of users and for each service of a plurality of services at the backend database **240**. In one embodiment, each service of the plurality of services may include, but not limited to, at least one of, messaging, media, and the like. After the active process fails and the standby process takes over. The backup of data created for each user and each service includes a correlation identifier between each user and the associated one or more services at the backend database **240**. As used herein, the correlation identifier is a unique identifier assigned to a set of related events, messages, or transactions that need to be correlated or linked together within a system or across systems. This identifier allows systems or components to associate related pieces of information and track the flow of a particular process or transaction through a distributed system.

[0053] The generating module **220** is configured to maintain a copy of a timer in the standby process for each service of the plurality of services in the active process and arrange a plurality of active process failure services in a sequence upon failure of active process of one or more services of the plurality of services in order to switch to the standby process. In one embodiment, the plurality of active process failure services includes services whose respective active processes have failed. In one embodiment, the timer, may include, but not limited to, a retransmission timer, a keepalive timer, persistent timer and the like. As used herein, the retransmission timer is used to retransmit a TCP segment when a sender does not receive an acknowledgment within a certain time frame. For example, if a TCP sender sends the segment and doesn't receive the acknowledgment within the Retransmission Timeout (RTO) period, it will retransmit the segment.

[0054] In a preferred embodiment, the sequence refers to an order of the plurality of active process failure services which are considered in order to rebuild data. Let us consider for example the one or more active process failure services include, but not limited to, S1, S2, S3, ..., Sn which are rebuilt by using timers T1, T2, T3,...,Tn respectively. Let us consider that for the first active process failure service S1, the timer T1 has 5 seconds left and for the second active process failure service S2, the timer T2 has 10 seconds left and for the third active process failure service S3, the timer T3 has 15 seconds left. In this scenario, the rebuilding module **225** is configured to rebuild the data for the first active process failure service S1 first, since the timer T1 has 5 seconds left which is less compared to timers T2 and T3 respectively. Similarly, after rebuilding data for the first active process failure service S1, the rebuilding module rebuilds data for the second active process failure service S2 and thereafter for the third active process failure service S3. In an embodiment, the rebuilding module rebuilds data for all the plurality of active process failure services.

[0055] In an alternate embodiment, for example, for the first active process failure service S1, the timer T1 may have already expired, the standby process turn to active process will preferentially rebuild the first active process failure service S1 foremost. In another alternate embodiment, for the second active process failure service S2, the timer T2 is not expired, and for the third active process failure service S3, the timer T3 is not expired. The rebuilding module **225** will preferentially rebuild the first active process failure service S1 foremost from turning the standby process to the active process and then move on to the data for which the associated timer is expiring next.

[0056] The timer associated with each UE is run in the active process and the same is checkpointed to the standby process. After failover in the active process, when the standby process becomes active, it restores these timers of the remaining period from when it is started in active process. When the timer times out, the process checks if the corresponding PDU session context is restored in a local cache. If not, it fetches a context from the SDL and restores it in the local cache.

[0057] The generating module **220** of the one or more processors **205** is configured to assign a priority value for each of the plurality of active process failure services based on a respective timer status. In one embodiment, the timer status may include, at least one of a timer of the specific active process failure service expires prior compared to the respective timers of the remaining plurality of active process failure services arranged in the sequence, and the timer of the specific active process failure service has already expired. The timers running in the active process are check pointed in the standby process also.

[0058] Further, the generating module **220** of the one or more processors **205** is configured to arrange each of the plurality of active process failure services in the sequence based on the assigned priority value. The one or more active process failure services whose priority value is higher is arranged prior compared to the remaining plurality of active process failure services in the sequence from the plurality of active process failure services.

[0059] The one or more processors **205** further includes the rebuilding module **225** in communication with the generating module **220**. More specifically, the rebuilding module **225** is communicably coupled with the generating module **220** to rebuild data for each of the active process failure services as per the sequence.

[0060] In one embodiment, the rebuilding module **225** is configured to rebuild data for each service of the active process failure service as per the sequence. In an embodiment, the rebuilding data may refer to reconstructing or restoring the data that has been damaged, lost, or corrupted. The rebuilding data can involve using backups, specialized software, or other methods to restore the data by using rebuilding techniques. Accordingly, the rebuilding module **225** is configured to implement rebuilding techniques such as, but not limited to, backup and restore, Redundant Array of Independent Disks (RAID), Error Correcting Codes (ECC) and utilization of Application Programming Interface (API), to rebuild the data using the backend database **240** based on the respective timer status. By rebuilding data, data loss

operational disruption, financial losses, reputational damage, and legal and compliance issues are avoided. Further, rebuilding data for the plurality of active process failure services as per the sequence ensures that the load on the standby process is reduced, thereby enabling the active process to receive a new request and to reduce fail over duration.

[0061] The rebuilding module **225** is configured to rebuild the data for the active process failure service arranged foremost in the sequence and give first and foremost preference to rebuild the data for the timer has already expired when the standby process takes over. Subsequently, the rebuilding will be done for the data for which the timer is expiring next.

[0062] Further, the rebuilding module **225** is configured to delay rebuilding of data for the remaining active process failure services until the active process failure service arranged foremost is built and thereafter rebuilding data for the remaining active process failure services as per the sequence ensures that the load on the standby process is reduced, thereby enabling the active process to receive a new request and to reduce fail over duration. Each active process failure service is rebuilt using the respective backup of data stored at the backend database **240**.

[0063] The monitoring module **230** of the one or more processors **205** is communicably connected to the rebuilding module **225**. The monitoring module **230** is configured to check one or more statuses of the rebuilt data for each active process failure service and expiry of each timer whether the associated data has been rebuilt using the backend database **240** in the standby process turn to active process. In one embodiment, the one or more statuses may include, but not limited to, failure of the active process, and expiry of each timer.

[0064] Referring to **FIG. 3**, **FIG. 3** describes a preferred embodiment of the system **125**. It is to be noted that the embodiment with respect to **FIG. 3** will be explained

with respect to the first UE **110a** for the purpose of description and illustration and should nowhere be construed as limited to the scope of the present disclosure.

[0065] As mentioned earlier, the first UE **110a** includes one or more primary processors **305** communicably coupled to the one or more processors **205** of the system **125**. The one or more primary processors **305** are coupled with a memory unit **310** storing instructions which are executed by the one or more primary processors **305**. Execution of the stored instructions by the one or more primary processors **305** enables the first UE **110a** to transmit, a request from a user via an interface module to a communication network **105**, in order to avail the one or more services. The execution of the stored instructions by the one or more primary processors **305** further enables the first UE **110a** to transmit, the request from the user via the interface module to the communication network **105**.

[0066] As mentioned earlier, the one or more processors **205** of the system **125** is configured to rebuild the data by using the backend database **240**. More specifically, the one or more processors **205** of the system **125** is configured to rebuild the data from a kernel **315** of at least one of the first UE **110a** in response to switch from the active process to the standby process for one or more services of the plurality of services.

[0067] The kernel **315** is a core component serving as the primary interface between hardware components of the first UE **110a** and the plurality of services at the backend database **240**. The kernel **315** is configured to provide the plurality of services on the first UE **110a** to resources available in the communication network **105**. The resources include one of a Central Processing Unit (CPU), memory components such as Random Access Memory (RAM) and Read Only Memory (ROM).

[0068] In the preferred embodiment, the generating module **220** of the one or more processors **205** is communicably connected to the kernel **315** of the first UE **110a**. The generating module **220** is configured to generate the standby process by creating

backup of data of the active process for each user of the plurality of users and for each service of the plurality of services at the backend database **240**.

[0069] In the preferred embodiment, the rebuilding module **225** of the one or more processors **205** is communicably connected to the generating module **220** to rebuild the data corresponding to the timer status. Upon rebuilding the data, the rebuilding module **225** implements rebuilding techniques such as, but not limited to, backup and restore, Redundant Array of Independent Disks (RAID), Error Correcting Codes (ECC) and utilization of Application Programming Interface (API), to rebuild the data using the backend database **240** based on the respective timer status.

[0070] In one embodiment, the system **125** further includes a timer running module **320** registered to the kernel **315**. The timer running module **320** aids in running the timer for each user and each service and activity for data updating, data rebuilding, data transmitting etc.

[0071] The one or more processors **205** further includes the monitoring module **230** in communication with the rebuilding module **225**. Upon checking of the status of the rebuilt data for each active process failure service. The monitoring module **230** monitors expiry of the timer available in the backend database **240**.

[0072] **FIG. 4** is a flow diagram of a method **400** for switching from an active process to a standby process for at least one service in a communication network **105**, according to one or more embodiments of the present invention. The method **400** is adapted to generate the standby process by creating backup of data of the active process for each user of a plurality of users and for each service of a plurality of services at a backend database. More specifically, the method further utilizes the rebuilding data for each of the active process failure services as per the sequence. The method further utilizes monitoring a status of the rebuilt data for each active process failure service. For the purpose of description, the method **400** is described with the

embodiments as illustrated in **FIG. 2** and should nowhere be construed as limiting the scope of the present disclosure.

[0073] At step **405**, the method **400** includes the step of generating, by the one or more processors **205**, the standby process by creating backup of data of the active process for each user of a plurality of users and for each service of a plurality of services at the backend database **240** by using the generating module **220**. In one embodiment, each service of the plurality of services may include, but not limited to, at least one of, messaging, media, and the like. After the active process fails and the standby process takes over. The backup of data created for each user and each service includes a correlation identifier between each user and the associated one or more services at the backend database **240**.

[0074] At step **410**, the method **400** includes the step of maintaining, by the one or more processors, a copy of a timer in the standby process for each service of the plurality of services in the active process. In one embodiment, the timer, may include, but not limited to, a transmission timer, a keepalive timer and the like. When the timer times out, the process checks if the corresponding PDU session context is restored in a local cache. If not, it fetches the context from the SDL and restores it in the local cache. In a preferred embodiment, all the timers running in the active process are check pointed in the standby process as well, since the copy of the timer of the active process is also maintained at the standby process.

[0075] At step **415**, the method **400** includes the step of arranging, by the one or more processors **205**, a plurality of active process failure services in a sequence upon failure of active process of one or more services of the plurality of services in order to switch to the standby process. In a preferred embodiment, the sequence refers to an order of the plurality of active process failure services which are considered in order to rebuild data. In one embodiment, the plurality of active process failure services includes services whose respective active processes have failed and assigning a priority value for each of the plurality of active process failure services based on a

respective timer status. In one embodiment, the timer status may include, at least one of a timer of the specific active process failure service expires prior compared to the respective timers of the remaining plurality of active process failure services arranged in the sequence. In an alternate embodiment, the timer status of the specific active process failure service has already expired.

[0076] At step **420**, the method **400** includes the step of rebuilding, by the one or more processors **205**, the data for the active process failure service arranged foremost in the sequence and give first and foremost preference to rebuild the data for the timer has already expired when the standby process takes over. Subsequently, the rebuilding will be done for the data for which the timer is expiring next.

[0077] At step **420**, further the method **400** includes the step of delaying, by the one or more processors **205**, rebuilding of data for the remaining active process failure services until the active process failure service arranged foremost is built and thereafter rebuilding data for the remaining active process failure services as per the sequence. Let us consider for example the one or more active process failure services include, but not limited to, S1, S2, S3, ..., Sn which are rebuilt by using timers T1, T2, T3, ..., Tn. Let us consider that for the first active process failure service S1, the timer T1 has 5 seconds left and for the second active process failure service S2, the timer T2 has 10 seconds left and for the third active process failure service S3, the timer T3 has 15 seconds left. In this scenario, the rebuilding module **225** is configured to rebuild the data for the first active process failure service S1 first, since the timer T1 has 5 seconds left which is less compared to timers T2 and T3 respectively. Similarly, after rebuilding data for the first active process failure service S1, the rebuilding module rebuilds data for the second active process failure service S2 and thereafter for the third active process failure service S3. In an embodiment, the rebuilding module rebuilds data for all the plurality of active process failure services.

[0078] In an alternate embodiment, for example, for the first active process failure service S1, the timer T1 may have already expired, the standby process turn to active

process will preferentially rebuild the first active process failure service S1 foremost. In another alternate embodiment, for the second active process failure service S2, the timer T2 is not expired, and for the third active process failure service S3, the timer T3 is not expired. The rebuilding module **225** will preferentially rebuild the first active process failure service S1 foremost from turning the standby process to the active process and then move on to the data for which the associated timer is expiring next.

[0079] At step **425**, the method **400** includes the step of checking one or more statuses of rebuilt data for each active process failure service and expiry of each timer whether the associated data has been rebuilt using the backend database **240** in the standby process turn to active process. In one embodiment, the one or more statuses may include, but not limited to, failure of the active process, and expiry of each timer.

[0080] In a preferred embodiment, the method **400** for switching from the active process to the standby process in a communication network **105** is provided. During operation the one or more processors **205** performs the step of transmitting a new request from at least one of the first UE **110a**, the second UE **110b**, and the third UE **110c**. The one or more processors **205** further performs rebuilding the data for each of the active process failure service as per the sequence. The one or more processors **205** further configured to perform the step of monitoring the status of the rebuilt data for each active process failure service.

[0081] **FIG. 5** shows a flow diagram of a method **500** for arranging plurality of active process failure services in a sequence, according to various embodiments of the present system.

[0082] At step **505**, the method **500** includes the step of arranging, by the one or more processors **205**, a plurality of active process failure services in a sequence upon failure of active process of one or more services of the plurality of services in order to switch to the standby process. In one embodiment, the plurality of active process failure services includes services whose respective active processes have failed.

Thereafter, a priority value is assigned for each of the plurality of active process failure services based on a respective timer status. In one embodiment, the timer status may include, at least one of a timer of the specific active process failure service expires prior compared to the respective timers of the remaining plurality of active process failure services arranged in the sequence. In an alternate embodiment, the timer status of the specific active process failure service has already expired. In a preferred embodiment, all the timers running in the active processes are check pointed in the standby process as well, since a copy of the timer of the active process is also maintained at the standby process.

[0083] At step **510**, the method **500** includes the step of arranging each of the plurality of active process failure services in the sequence based on the assigned priority value. The one or more active process failure services whose priority value is higher is arranged prior compared to the remaining plurality of active process failure services in the sequence from the plurality of active process failure services.

[0084] Let us consider for example the one or more active process failure services include, but not limited to, S1, S2, S3, ..., Sn which are rebuilt by using timers T1, T2, T3, ..., Tn. Let us consider that for the first active process failure service S1, the timer T1 has 5 seconds left and for the second active process failure service S2, the timer T2 has 10 seconds left and for the third active process failure service S3, the timer T3 has 15 seconds left. In this scenario, the rebuilding module **225** is configured to rebuild the data for the first active process failure service S1 first, since the timer T1 has 5 seconds left which is less compared to timers T2 and T3 respectively. Similarly, after rebuilding data for the first active process failure service S1, the rebuilding module rebuilds data for the second active process failure service S2 and thereafter for the third active process failure service S3. In an embodiment, the rebuilding module rebuilds data for all the plurality of active process failure services. In an alternate embodiment, for the first active process failure service S1, the timer T1 may have already expired, for the second active process failure service S2, the timer T2 is not expired, for the third active process failure service S3, the timer T3 is not expired. The

rebuilding module **225** will preferentially rebuild the first active process failure service S1 foremost to turning the standby process to the active process and then move on to the data for which the associated timer is expiring next.

[0085] **FIG. 6** shows a flow diagram of a method **600** for rebuilding data for each active process failure service in the sequence, according to various embodiments of the present system.

[0086] At step **605**, the method **600** includes the step of rebuilding data for the active process failure service arranged foremost in the sequence and giving first and foremost preference to rebuild the data for the timer has already expired when the standby process takes over. Subsequently, the rebuilding will be done for the active process failure service for which the timer is expiring next. For example, for the second active process failure service S1, the timer T1 may have already expired, for the second active process failure service S2, the timer T2 is not expired, and for the third active process failure service S3, the timer T3 is not expired. The rebuilding module **225** will preferentially rebuild the data of the second active process failure service S1 foremost from turn the standby process to the active process and then move on to the data for which the associated timer is expiring next.

[0087] At step **610**, the method **600** includes the step of delaying the rebuilding of data for the remaining active process failure services until the active process failure service arranged foremost is built and thereafter rebuilding data for the remaining active process failure services as per the sequence. Each active process failure service is rebuilt using the respective backup of data stored at the backend database 240 based on the respective timer status. In one embodiment, the timer status is one of a timer of the specific active process failure service expires prior compared to the respective timers of the remaining plurality of active process failure services arranged in the sequence, and the timer of the specific active process failure service has already expired. Thereafter rebuilding data for the plurality of active process failure services

as per the sequence to ensure that the load on the standby process is reduced, thereby enabling the active process to receive a new request and to reduce fail over duration.

[0088] The present invention further discloses a non-transitory computer-readable medium having stored thereon computer-readable instructions. The computer-readable instructions are executed by a processor **205**. The processor **205** is configured to transmit a new request from a UE **110** pertaining to maintain a copy of a timer in a standby process when failure of the active process of one or more services of the plurality of services. The processor **205** is further configured to rebuild the data for each of the active process failure services as per the sequence based on the respective timer status. The processor **205** is further configured to check the status of rebuilt data for each active process failure service.

[0089] A person of ordinary skill in the art will readily ascertain that the illustrated embodiments and steps in description and drawings (FIG.1-6) 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. 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.

[0090] The present disclosure incorporates technical advancement of servicing a new request without delay by using the timer to bring out an advantage of rapidly rebuilding the data when the active process service is failed. Once the active process service is failed, the active process failure service is switched to the standby process for rebuilding the data of the active process failure service based on the respective

timer status to ensure that the load on the standby process is reduced, thereby enabling the active process to receive a new request and to reduce fail over duration.

[0091] The present disclosure significantly reduces, by incorporating switching from active process to standby process, improves system throughput enabling faster rebuilding of data, and rebuilding data for the plurality of active process failure services as per the sequence ensures that the load on the standby process is reduced, thereby enabling the active process to receive the new request and to reduce fail over duration.

[0092] The present invention offers multiple advantages over the prior art and the above listed are a few examples to emphasize on some of the advantageous features. The listed advantages are to be read in a non-limiting manner.

REFERENCE NUMERALS

- [0093] Environment - 100;
- [0094] Communication Network - 105;
- [0095] User Equipment - 110;
- [0096] Server - 115;
- [0097] System - 125;
- [0098] One or more processors -205;
- [0099] Memory – 210;
- [00100] Generating Module- 220;
- [00101] Rebuilding Module - 225;
- [00102] Monitoring Module - 230;
- [00103] Database - 240;
- [00104] One or more primary processors -305;
- [00105] Memory- 310;
- [00106] Kernel-315.

We Claim:

1. A method (400) for switching from an active process to a standby process for at least one service in a network (105), the method (400) comprises the steps of:
 - generating (405), by one or more processors (205), the standby process by creating backup of data of the active process for each user of a plurality of users and for each service of a plurality of services at a backend database (240);
 - maintaining (410), by the one or more processors (205), a copy of a timer in the standby process for each service of the plurality of services in the active process;
 - upon failure of active process of one or more services of the plurality of services, in order to switch to the standby process, arranging (415), by the one or more processors (205), a plurality of active process failure services in a sequence;
 - rebuilding (420) data, by the one or more processors (205), for each of the active process failure service as per the sequence; and
 - checking (425), by the one or more processors (205), a status of rebuilt data for each active process failure service.
2. The method (400) as claimed in claim 1, wherein the backup of data created for each user and each service includes a correlation identifier between each user and the associated one or more services at the backend database.
3. The method (400) as claimed in claim 1, wherein the plurality of active process failure services includes services whose respective active processes have failed.
4. The method (400) as claimed in claim 1, wherein the one or more processors (205) arranges the plurality of active process failure services in the sequence by:
 - assigning, a priority value for each of the plurality of active process failure services based on a respective time status;

arranging each of the plurality of active process failure services in the sequence based on the assigned priority value, wherein from the plurality of active process failure services, the one or more active process failure services whose priority value is higher is arranged prior compared to the remaining plurality of active process failure services in the sequence.

5. The method (400) as claimed in claim 1, wherein a specific active process failure service is assigned a higher priority value compared to the remaining plurality of active process failure services, when the timer status is one of:
 - a timer of the specific active process failure service expires prior compared to the respective timers of the remaining plurality of active process failure services arranged in the sequence; and
 - the timer of the specific active process failure service has already expired.
6. The method (400) as claimed in claim 1, wherein the step of rebuilding data, for each active process failure service in the sequence, includes the steps of:
 - rebuilding data for the active process failure service arranged foremost in the sequence; and
 - delaying, rebuilding of data for the remaining active process failure services in the sequence until the active process failure service arranged foremost is built and thereafter rebuilding data for the remaining active process failure services as per the sequence, in order to reduce the load on the standby process, to service a new request and to reduce fail over duration.
7. The method (400) as claimed in claim 1, wherein each active process failure service is rebuilt using the respective backup of data stored at the backend database (240).

8. The method (400) as claimed in claim 1, wherein each service of the plurality of services includes at least one of, messaging, media, etc.
9. A system (125) for switching from an active process to a standby process for at least one service in a network (105), the system (125) comprising:
 - a generating module (220) configured to:
 - generate, the standby process by creating backup of data of the active process for each user of a plurality of users and for each service of a plurality of services at a backend database (240);
 - maintain, a copy of a timer in the standby process for each service of the plurality of services in the active process; and
 - upon failure of active process of one or more services of the plurality of services, in order to switch to the standby process, arrange, a plurality of active process failure services in a sequence;
 - a rebuilding module (225) configured to, rebuild data, for each of the active process failure service as per the sequence; and
 - a monitoring module (230) configured to check, a status of rebuilt data for each active process failure service.
10. The system (125) as claimed in claim 9, wherein the backup of data created for each user and each service includes a correlation identifier between each user and the associated one or more services at the backend database (240).
11. The system (125) as claimed in claim 9, wherein the generating module (220) arranges the plurality of active process failure services in the sequence by:
 - assigning, a priority value for each of the plurality of active process failure services based on a respective time status; and
 - arranging each of the plurality of active process failure services in the sequence based on the assigned priority value, wherein from the plurality of active process failure services, the one or more active process failure services

whose priority value is higher is arranged prior compared to the remaining plurality of active process failure services in the sequence.

12. The system (125) as claimed in claim 11, wherein a specific active process failure service is assigned a higher priority value compared to the remaining plurality of active process failure services, when the timer status is one of:

a timer of the specific active process failure service expires prior compared to the respective timers of the remaining plurality of active process failure services arranged in the sequence; and

the timer of the specific active process failure service has already expired.

13. The system (125) as claimed in claim 9, wherein the rebuilding module (225) of the system (125) is further configured to:

rebuild data for the active process failure service arranged foremost in the sequence; and

delay, rebuilding of data for the remaining active process failure services until the active process failure service arranged foremost is built and thereafter rebuilding data for the remaining active process failure services as per the sequence, in order to reduce a load on the standby process, to service a new request and to reduce fail over duration.

14. A User Equipment (UE) (110), comprising:

one or more primary processors (305) coupled with a memory (310), communicatively coupled to one or more processors (205), wherein said memory (310) stores instructions which when executed by the one or more primary processors (305) causes the UE (110) to:

transmit, a request from a user via an interface module to a network (105), in order to avail the one or more services; and

wherein the one or more processors (205) are further configured to perform the method as claimed in claim 1.

15. A non-transitory computer-readable medium having stored thereon computer-readable instructions that, when executed by a processor (605), causes the processor (605) to:

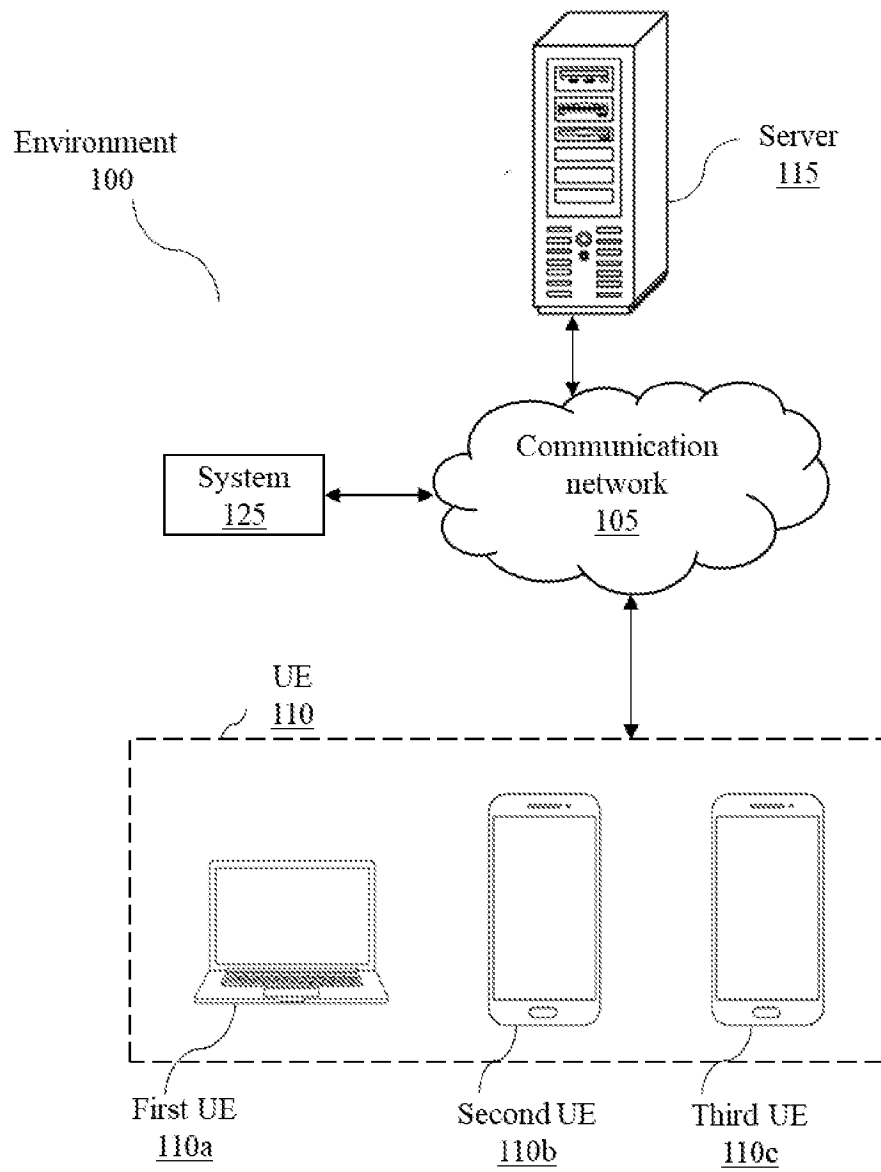
- generate, the standby process by creating backup of data of the active process for each user of a plurality of users and for each service of a plurality of services at a backend database (240);

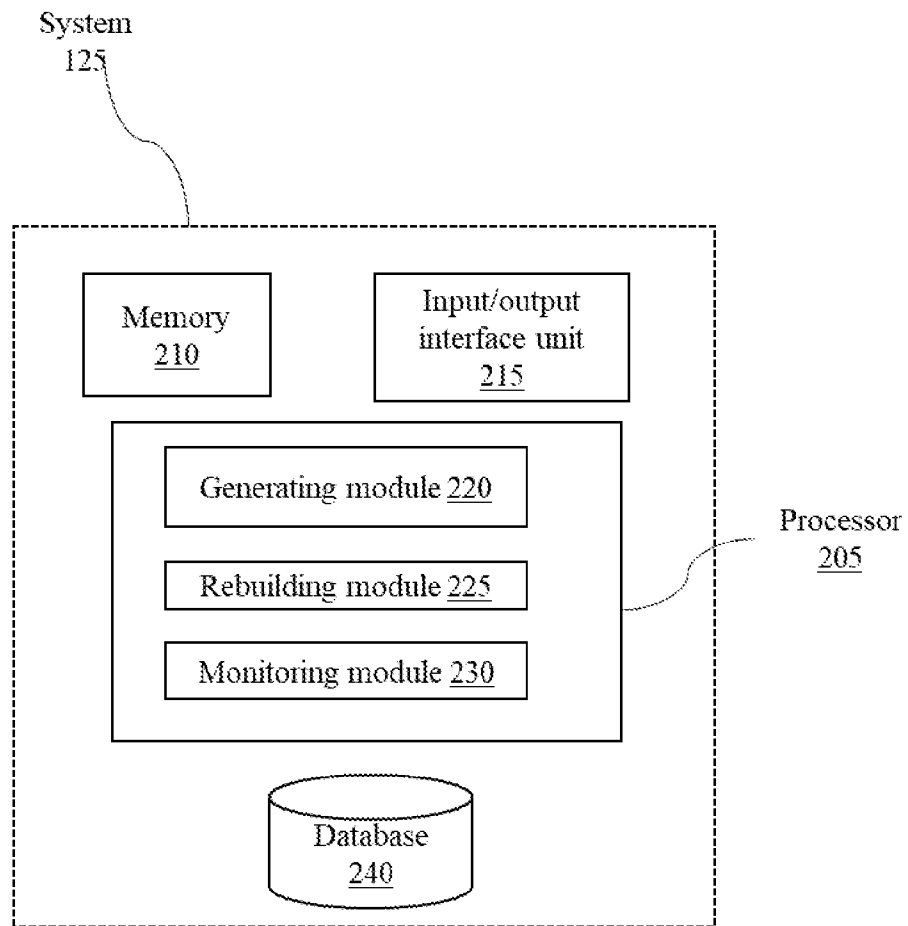
- maintain, a copy of a timer in the standby process for each service of the plurality of services in the active process;

- upon failure of active process of one or more services of the plurality of services, in order to switch to the standby process, arrange, a plurality of active process failure services in a sequence;

- rebuild data, for each of the active process failure service as per the sequence; and

- check, a status of rebuilt data for each active process failure service.

**FIG. 1**

**FIG. 2**

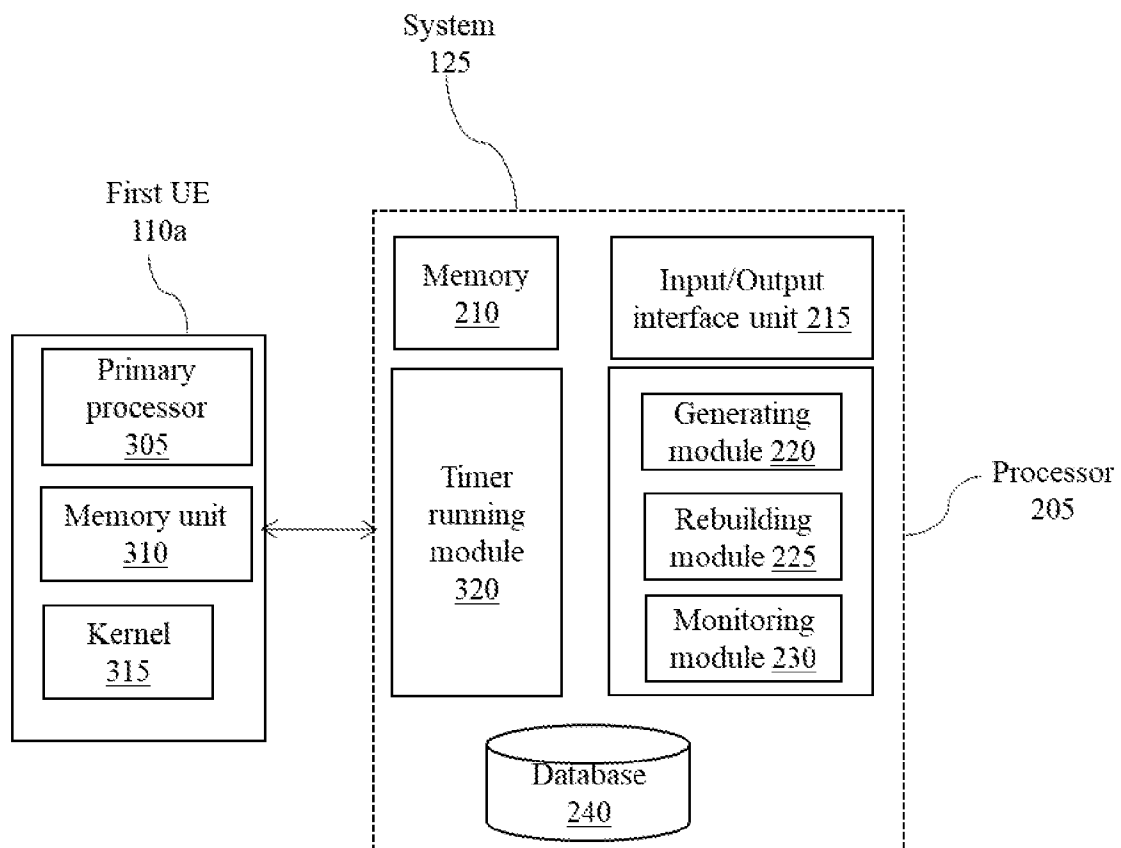


FIG. 3

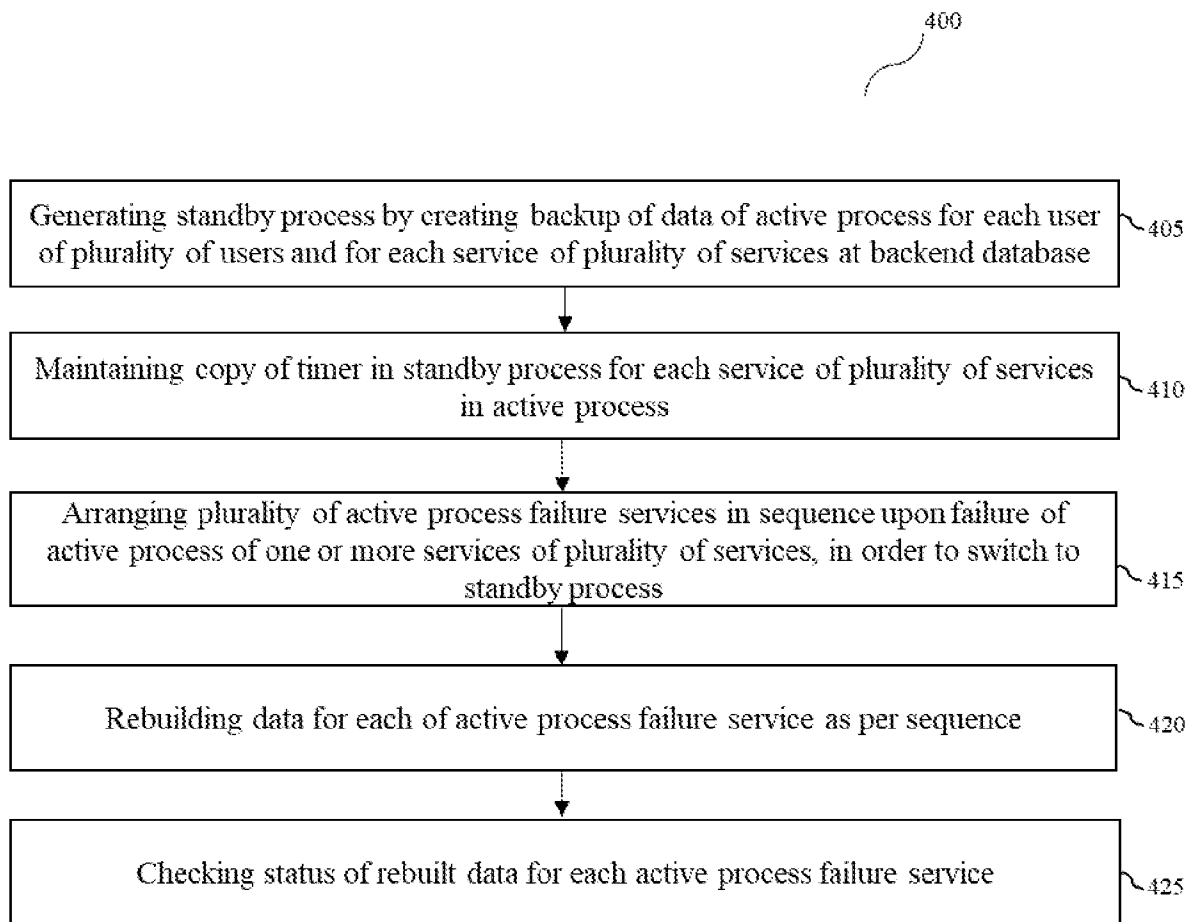
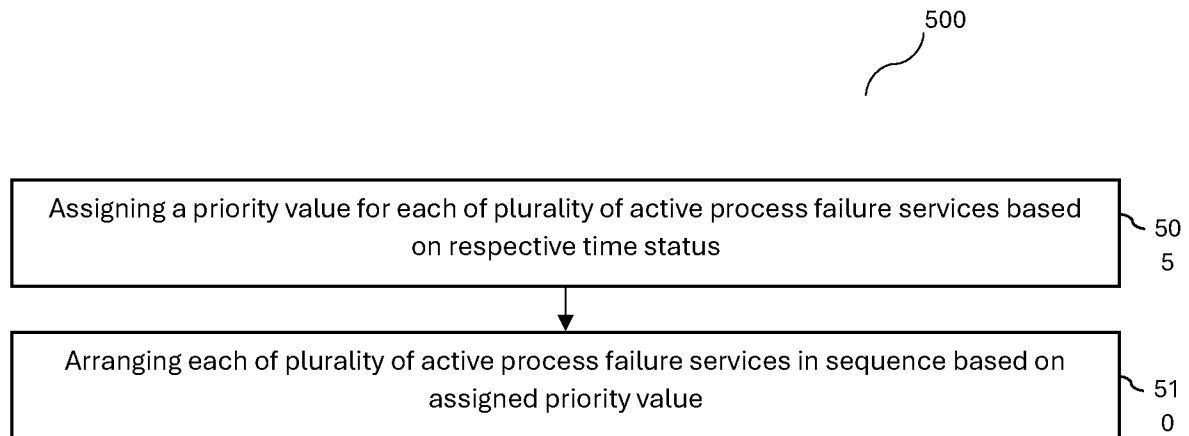
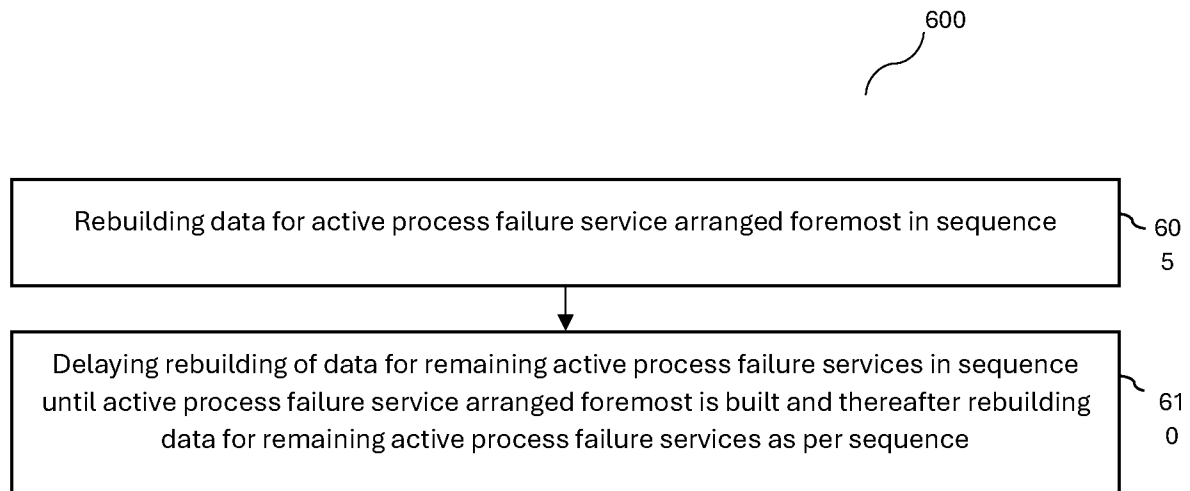


FIG. 4

**FIG. 5**

**FIG. 6**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IN2024/050930

A. CLASSIFICATION OF SUBJECT MATTER

G06F11/20, H04L41/0663, H04L69/40 Version=2024.01

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F, H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database consulted during the international search (name of database and, where practicable, search terms used)

PatSeer, IPO Internal Database

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US20210321325A1 (CISCO TECHNOLOGY INC) 14 October 2021 (14/10/2021) Abstract, figure 1-6, paragraphs 0019-0020, 0027, 0041-0046, 0053, 0057, 0087	1-15
A	WO2023280041A1 (SUNWAVE COMM CO LTD) 12 January 2023 (12/01/2023) whole document	1-15
A	US20230198838A1 (ARISTA NETWORKS INC) 22 June 2023 (22/06/2023) whole document	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

22-10-2024

Date of mailing of the international search report

22-10-2024

Name and mailing address of the ISA/

Indian Patent Office

Plot No.32, Sector 14, Dwarka, New Delhi-110075

Facsimile No.

Authorized officer

Dinesh Chandra

Telephone No. +91-1125300200

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/IN2024/050930

Citation	Pub.Date	Family	Pub.Date
WO 2023280041 A1	12-01-2023	CN 113645650 B	04-06-2024
		EP 4351205 A1	10-04-2024
		US 20240106699 A1	28-03-2024