



jtelo ACD

Skalierung und Redundanz

Impressum

jtel ACD – Skalierung und Redundanz

jtel GmbH
Valentin-Linhof-Straße 2
81829 München
Germany

Tel.: +49 (89) 4614950-0
Fax: +49 (89) 4614950-29
EMail : info@jtel.de

Copyright

© jtel GmbH 2017 - 2019. Alle in diesem Dokument genannten Bezeichnungen von Erzeugnissen sind Warenzeichen oder Marken von jtel GmbH bzw. der jeweiligen Inhaber.

History

Date	Autor	Version	Notizen
09.04.2017	LAG	1.0	Erste Version. Zusammenführung aus diversen Quellen.
14.04.2019	LAG	1.1	Überarbeitung.

Inhaltsverzeichnis

Überblick	5
1.1 Einführung	5
1.2 High Availability und Fault Tolerance	5
1.3 Diagramme	5
1.4 Architektur	6
1.4.1 Datei Storage	6
1.4.1.1 Funktion	6
1.4.1.2 Skalierung Datei Storage	6
1.4.1.3 Redundanz Datei Storage	7
1.4.2 MySQL Datenbank	7
1.4.2.1 Funktion	7
1.4.2.2 Skalierung MySQL Datenbank	7
1.4.2.3 Redundanz MySQL Datenbank	8
1.4.3 JBOSS Webserver	8
1.4.3.1 Funktion	8
1.4.3.2 Skalierung JBOSS Webserver	8
1.4.3.3 Redundanz JBOSS Webserver	9
1.4.4 jtel Telefonie Server	9
1.4.4.1 Funktion	9
1.4.4.2 Skalierung jtel Telefonie-Server	9
1.4.4.3 Redundanz jtel Telefonie-Server	10
1.4.5 Load Balancer	10
1.4.5.1 Funktion	10
1.4.5.2 Skalierung Load-Balancer	10
1.4.5.3 Redundanz Load-Balancer	10
Anhang 1 - Copyright	12

Überblick

1.1 Einführung

Dieses Dokument beschreibt das Redundanz- sowie Skalierungskonzept der jtel ACD.

1.2 High Availability und Fault Tolerance

Grundsätzlich gibt es zwei Themen bei Redundanz:

- High Availability
- Fault Tolerance

High Availability beschäftigt sich mit einer hohen Verfügbarkeit einer Lösung, mit entsprechend geringen Ausfallzeiten.

Fault Tolerance beschäftigt sich mit der Toleranz gewisser Fehler, wenn Sie eintreten. Das System überlebt die Fehlfunktion ein oder mehrere Komponenten und funktioniert weiter.

In der jtel Lösung gehen wir meist davon aus, dass alle Komponenten „arbeiten“. Komponenten, die Redundant ausgelegt sind, bieten sowohl High Availability als auch Fault Tolerance. Eine Ausnahme kann eine Geo-Redundanz darstellen, je nach Latenzzeit zwischen den Standorten.

Die Skalierung der jtel Lösung ist nicht trivial auszurechnen, da sie stark von den Nutzerprofilen abhängig ist. Beispielsweise, ein System mit 2000 concurrent Calls und längere Gesprächszeiten im Schnitt, verhält sich entschieden anders als ein System mit 500 concurrent Calls mit einer sehr kurzen Gesprächszeit.

Des Weiteren ist die Performance der Anlage stark von der Infrastruktur abhängig. Liefert die virtuelle Umgebung zuverlässige Performance im Bezug auf RAM, CPU und Storage, so können in manchen Umgebungen deutlich höhere Kennzahlen erreicht werden als in anderen Umgebungen.

1.3 Diagramme

Siehe: [jtel ACD Skalierung und Redundanz 2019.Q2](#).

1.4 Architektur

Das jtel System benötigt folgende Server Komponenten:

- Datei-Storage
- MySQL Datenbank
- JBOSS Webserver
- jtel Telefonie Server
- Load Balancer

die in den folgenden Abschnitten kurz beschrieben werden.

1.4.1 Datei Storage

1.4.1.1 Funktion

Das Storage wird genutzt um Bewegungsdaten und Programme der Plattform zu speichern, sowie die Grundsoftware selbst. Auf dem Dateistorage befinden sich:

- Skripte für den jtel Telefonieserver
- Aktuelle Webapplikationsdateien für den Webserver
- Konfigurationsdateien für diverse zentrale Komponenten
- Reporting Quelldateien
- Audiodateien (hochgeladen) der/s Mandanten
- Aufgezeichnete Recordings und Gesprächsmitschnitte der Mandanten
- Empfangene Faxe der Mandanten
- Dateien betreffend weitere Kommunikationskanäle der Mandanten (Multichannel Dateien)
- Logdateien
- Skins für das Webportal

Das Dateisystem wird per SMB (Windows File Share) von diversen Servern gemountet werden.

1.4.1.2 Skalierung Datei Storage

Je nach Größe des Systems, (Anzahl Mandanten, Agenten, Anrufe ...) sowie die gewünschte Vorhaltezeit für beispielsweise Anrufaufzeichnungen, sollte das Dateisystem entsprechend vergrößert werden.

Da die tatsächlich benötigte Größe von dem Nutzungsprofil der Kunden zu 100% abhängig ist, kann hier keine exakte Größe benannt oder abgeschätzt werden. Der größte Verbraucher sind die Gesprächsaufzeichnungen (1 Stunde entspricht ca. 110 MB Verbrauch).

Standardmäßig empfiehlt jtel für ein Standalone-System mit bis zu 200 Agenten 512 GB Storage zur Verfügung zu stellen. Diese Berechnung beinhaltet nicht die Aufbewahrung von Aufzeichnungen für alle Anrufe.

1.4.1.3 Redundanz Datei Storage

Um das Storage redundant auszulegen gibt es grundsätzlich drei Möglichkeiten.

Möglichkeit	Beschreibung	Vor- / Nachteile
File Mount auf Storage System	<p>Das Dateisystem befindet sich auf einem Storage-System (NetApp, Windows Server Cluster etc.), welches grundsätzlich Redundant ausgelegt ist.</p> <p>Es muss von den Clients per SMB (Windows und Linux) gemountet werden können, dies wird nicht von allen Storage-Systemen angeboten.</p>	<p>+++</p> <p>Da das Storage-System in sich eine Redundanz bietet, ist dies eine sehr gut geeignete Variante.</p>
DRBD File-System	<p>Es werden zwei Load-Balancer mit einer virtuellen IP Adresse durch das jtel System abgebildet.</p> <p>Das DRBD File-System repliziert die Daten zwischen den beiden Load-Balancern auf Block-Ebene.</p> <p>Auf beiden Servern läuft der SMB/NMB Dienst. Nur einer dieser Server besitzt die virtuelle IP Adresse.</p>	<p>++</p> <p>Ermöglicht zwar nur die Verwendung von 2 Systemen für die Redundanz des Dateisystems, ist aber sehr performant.</p> <p>Ist gegenüber Netzwerkproblemen (Ausfälle, die länger als eine Sekunde andauern) nicht tolerant.</p> <p>Durch zwei Nodes muss im Failure-Fall eine Split-Brain Situation gelöst werden.</p>

1.4.2 MySQL Datenbank

1.4.2.1 Funktion

Die MySQL Datenbank beinhaltet die Konfiguration der jtel Anlage (Mandanten, User, ACD Gruppen, Rufnummern etc.), sowie die damit assoziierte Statistik.

Sollte eine Master-Slave Implementierung vorliegen, ist die Master-Datenbank für alle Schreibvorgänge der Konfiguration und Statistik, sowie Lesevorgängen auf der Konfiguration zuständig. Bei statistischen Anfragen und im Reporting kann die Slave Datenbank verwendet werden.

1.4.2.2 Skalierung MySQL Datenbank

Die MySQL Datenbank kann als Master-Slave Architektur aufgestellt werden, um die Skalierung der Datenbank zu ermöglichen.

Durch die Separierung der Abfragen für die Echtzeit- sowie Historische-Statistik wird die Master-Datenbank entlastet, auf der in der Regel die höchste Last liegt.

Hier ist das Verhalten von den Nutzerprofilen abhängig. Befinden sich Kunden auf der Plattform, die mit dutzenden Supervisoren und Wallboards arbeiten, sowie Statistiken in großen Mengen abrufen oder abonnieren, ist ggf. eine weitere Slave-Datenbank einzuplanen.

Sollte die Datenbank auf Ihre Grenzen stoßen, (RAM / CPU) führt eine Skalierung des RAMs bzw. Der CPU zu einer weiteren Entlastung. Dieses Verhalten ist jedoch nicht linear, sodass bei einer Call-Rate (oder Event-Rate) von ca. 30 CPS das natürliche Ende naht. Hierfür erforderlich (bei einer performanten Systemumgebung) ist 64 GB RAM, sowie 16 Kernen auf der Master Datenbank.

Die Slave-Datenbank benötigt in der Regel nicht soviel RAM oder Kerne. wie die Master-Datenbank, und kann ca. 25-50% kleiner dimensioniert werden.

1.4.2.3 Redundanz MySQL Datenbank

Um eine vollständige Redundanz abzubilden, wird ein Master-Master paar gebildet, welches jeweils an damit assoziierte Slaves repliziert.

Die Verteilung der Datenbank-Sessions erfolgt über den Load-Balancer.

Es wird stets nur ein Master angesprochen, um gewisse Nebeneffekte (Locking und Inkonsistenz) vorzubeugen. Solange der erste Master verfügbar ist, wird immer dieser angesprochen, der zweite Master dient als Backup. Erfolgt ein Failover auf die 2. Datenbank, muss dies in der Alarmierung erkannt werden, und eine manuelle Rückschaltung eingeplant werden. Es findet kein automatischer Rückfall auf den ersten Master statt, um Inkonsistenzen bei der Replikation vorzubeugen.

Jeder der Slaves übernimmt eine Primärrolle – der Erste die Bereitstellung von Daten für die Echtzeitstatistik, der zweite die Bereitstellung von Daten für das Reporting. Bei den Slaves findet die Verteilung nach „First“ statt, mit automatischer Rückfall sobald der anderer Server wieder verfügbar ist.

1.4.3 JBOSS Webserver

1.4.3.1 Funktion

Der JBOSS Webserver bietet das Front-End für Kunden und Agenten, um das jtel System über ein Web-Frontend zu konfigurieren bzw. zu verwenden.

1.4.3.2 Skalierung JBOSS Webserver

Jeder JBOSS Server kann ca. 100 bis 200 Websessions verarbeiten. Die tatsächliche Anzahl ist stark von den Nutzerprofilen abhängig. Handelt es sich beispielsweise um 100 Websessions für einen Mandanten mit dutzenden ACD Gruppen, ist das Verhalten womöglich deutlich schlechter als bei 200 Websessions und Mandanten mit nur wenig ACD Gruppen.

Sobald die Grenze der Nutzbarkeit erreicht ist (dies wird durch eine Verlangsamung des jtel Portals bemerkt), kann die weitere Skalierbarkeit durch:

- Zusätzliches RAM
- Zusätzliche CPU Kerne
- Zusätzliche Web-Server (benötigt Load-Balancer)

realisiert werden.

1.4.3.3 Redundanz JBOSS Webserver

Jeder der Webserver arbeitet grundsätzlich autag und benötigt lediglich eine Anbindung zum jtel Cluster sowie der Datenbank und dem File-System.

Um das System redundant auszuliegen, werden mehrere Webserver implementiert Die Sessions werden über den Load-Balancer verteilt.

1.4.4 jtel Telefonie Server

1.4.4.1 Funktion

Der jtel Telefonie Server ist zuständig für die Anrufentgegennahme und Abarbeitung der Call-Flows. Er terminiert die Anrufe mit SIP / RTP und verarbeitet diese entsprechend der Konfiguration und Call-Flows weiter. Des Weiteren ist der jtel Telefonie-Server für die Event-Verteilung in Multi-Channel Umgebungen zuständig Diese Funktion kann auf separate Server ausgelagert werden.

Das Thema SIP Verteilung, sprich Verteilung der Gespräche auf mehrere jtel Server, wird durch die TK-Anlage übernommen.

1.4.4.2 Skalierung jtel Telefonie-Server

Jeder jtel Telefonie-Server verarbeitet bis zu 1000 parallele Anrufe. Ob dies sinnvoll ist, hängt im Wesentlichen von der Call-Rate (CAPS) ab. Jeder Server sollte eine Rate von ca. 10 CAPS nicht überschreiten (obwohl der Server, als reine IVR ohne Nutzung des jtel Portals ca. 100 CAPS ermöglicht, skaliert dies nicht wegen der benötigten Datenbankabfragen).

Für die Verteilung der Anrufe auf mehreren Servern, gibt es grundsätzlich zwei sinnvolle Möglichkeiten:

Möglichkeit	Beschreibung	Vor- / Nachteile
Verteilung durch die TK-Anlage	Die TK-Anlage verteilt die Anrufe auf mehrere jtel Telefonie-Server selbstständig und erkennt auch die Nichterreichbarkeit eines Servers.	+++ Dies ist die einfachste und beste Möglichkeit. Beachte: nicht alle TK-Anlagen unterstützen die Verteilung der Gespräche. Es ist nicht ausreichend, auf das erste SIP Signal vom jtel System zu warten (nach INVITE erfolgt TRYING). Es muss ein RINGING vorliegen, oder ein finales Ergebnis (4xx oder 5xx).

Möglichkeit	Beschreibung	Vor- / Nachteile
Verteilung durch einen SIP Proxy	Ein SIP Proxy wird den jtel Telefonie-Servern vorgeschaltet und übernimmt selbst die Verteilung.	+++ jtel Lösung erst ab Q4.2019 verfügbar. Beachte: gleiches gilt wie bei Verteilung über die TK-Anlage. Der SIP Proxy muss auf ein RINGING oder finales Ergebnis warten, um sicherzustellen dass die Verteilung auch erfolgt ist.

1.4.4.3 Redundanz jtel Telefonie-Server

Vorausgesetzt, das Thema der Anrufverteilung ist gelöst, arbeiten die jtel Telefonie-Server im Verbund und können beliebig horizontal skaliert werden.

1.4.5 Load Balancer

1.4.5.1 Funktion

Einige Funktionen der Anlage in Verbindung mit Redundanz bzw. Skalierung bedingen den Einsatz des Load-Balancers. Diese Lösung wird mit haproxy realisiert.

Der Load-Balancer verteilt:

- Datenbank-Master Verbindungen
- Datenbank-Slave Verbindungen
- http / https Web-Sessions

und stellt Zugriff auf das File-System dar, für den Fall, dass das File-System nicht auf ein externes Storage implementiert wurde.

1.4.5.2 Skalierung Load-Balancer

Der Load-Balancer kann durch Hinzufügen von weiteren CPU Kernen sowie RAM skaliert werden.

1.4.5.3 Redundanz Load-Balancer

Soll der Load-Balancer redundant ausgelegt werden, so wird ein Paar mit einem PCS Cluster (Pacemaker / Corasync) realisiert. Jeder Load-Balancer fährt ständig folgende Dienste:

- SMB / NMB für Dateizugriff
- Eigene IP Adresse
- Haproxy

Folgende Dienste werden über den PCS Cluster implementiert:

- Virtuelle IP Adresse für den Client-Zugriff

Im Failover-Fall übernimmt die zweite Maschine die virtuelle IP Adresse und stellt die entsprechenden Ressourcen darüber zur Verfügung.

jtel GmbH
Valentin-Linhof-Str. 2
81829 München
Germany

Tel : +49 (89) 4614950-00

Fax : +49 (89) 4614950-29

Email : info@jtel.de

Support : support@jtel.de

Web : <http://www.jtel.de>

jtel® / the jtel logo and 8-Server® are registered trademarks of jtel GmbH. Copyright© 1997 - 2019 jtel GmbH. All rights reserved. All jtel product names are trademarks or registered trademarks of jtel GmbH.

Other brand and product names are trademarks or registered trademarks of their respective holders.