

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

SYSTEM DEPLOYMENT AND BENCHMARKING

Análise da aplicação Mattermost

GRUPO 5

ANA RODRIGUES	(A78763)
CLÁUDIA CORREIA	(A77431)
FILIPPE NUNES	(A78074)

10 de Novembro de 2018

Conteúdo

1	Introdução	1
2	Arquitetatura	2
2.1	Vantagens	2
2.2	Desvantagens	3
3	Componentes	4
4	Padrões de Distribuição	5
4.1	Padrão básico	5
4.2	Padrão da <i>Team Edition</i>	5
4.3	Padrão da <i>Enterprise Edition</i>	5
5	Formas de Comunicação	6
6	Pontos de Configuração	7
6.1	Base de Dados	7
6.2	Sistema <i>File Storage</i>	7
6.3	TLS no servidor Mattermost	7
6.4	Nginx	8
6.5	SSL e HTTP/2 com Nginx	8
7	Operações com desempenho crítico	9
7.1	Armazenamento insuficiente	9
7.2	Falha na base de dados	9
7.3	Bombardeamento de pedidos	9
8	Conclusão	10

Resumo

Este documento diz respeito à primeira fase do trabalho prático proposto na unidade curricular de System Deployment e Benchmarking (SD&B) da Universidade do Minho. O objetivo desta fase consiste na caracterização e análise da aplicação distribuída Mattermost.

1 Introdução

A aplicação distribuída Mattermost é um serviço de chat *open-source* e *self-hostable*, como alternativa ao Slack. Esta aplicação mantém toda a comunicação da equipa num único sítio, estando acessível em qualquer lugar. A plataforma foi escrita em Golang e React e utiliza um sistema de gestão de base dados SQL, tipicamente MySQL ou PostgreSQL. De forma a facilitar a interação com o utilizador, o Mattermost possui também uma interface *web*.

São três as principais camadas lógicas que constituem a aplicação, nomeadamente a *Proxy Layer*, a *Mattermost Application Layer* e a *Database Layer*. No que toca a elasticidade, o Mattermost possui replicação na *Database Layer*, sendo que poderá conter réplicas de leitura da base de dados, configuradas como *backups* redundantes.

Outra das características que torna esta aplicação tão apelativa é o facto de ela ser compatível com todas as *features* do Slack mas, apesar disso, não estar limitada às mesmas. O Mattermost é apoiado por uma comunidade entusiástica, sendo que pode ser utilizado com diversas aplicações como, por exemplo, GitLab, Jira, Redmine, entre outros. Por último, esta plataforma é também fácil de gerir e atualizar, uma vez que são lançadas melhorias a cada dois meses.

2 Arquitetura

A arquitetura da aplicação Mattermost é caracterizada pela existência de um intermediário, denominado servidor *proxy*, que gere os pedidos provenientes dos clientes, encaminhando-os posteriormente para o(s) servidor(es) Mattermost. A interação entre o cliente e o servidor *proxy* é realizada com base em protocolos. Esta arquitetura distribuída é denominada Servidor *Proxy*.

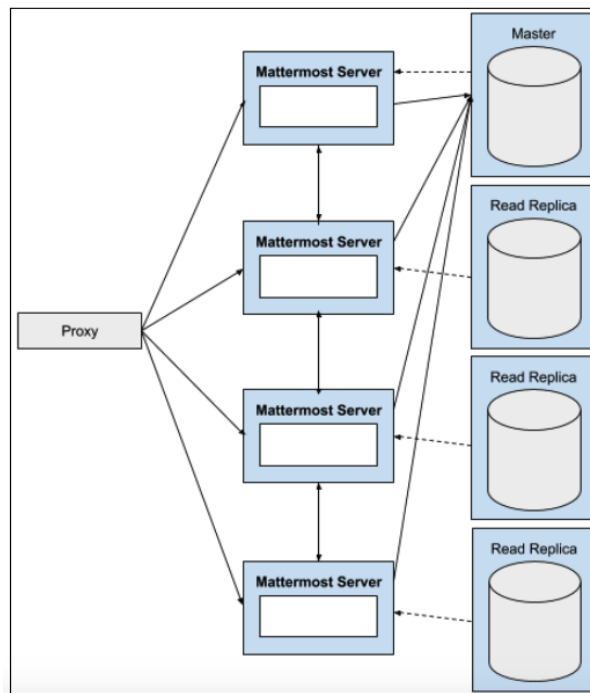


Figura 1: Arquitetura da aplicação Mattermost.

Retirado de <https://docs.mattermost.com/overview/architecture.html>.

2.1 Vantagens

Tal como seria de esperar, a arquitetura apresentada possui diversas vantagens, entre as quais se destacam:

- **Segurança** - o servidor *proxy* consegue recorrer a protocolos de segurança (TLS/SSL) e definir políticas de encaminhamento, de forma a proteger os servidores.
- **Monitorização** - o servidor *proxy* pode monitorizar o tráfego de conexão e registá-lo em *logs* específicos, que serão avaliados por ferramentas de monitorização.

Quando estamos perante a existência de múltiplos servidores Mattermost, existem alguns benefícios a considerar:

- **Transparência** - caso existam múltiplos servidores Mattermost isto será imperceptível para o cliente.
- **Performance** - caso existam vários servidores Mattermost, o servidor *proxy* tem a capacidade de distribuir a carga proveniente da rede por esses mesmos servidores, otimizando assim o desempenho da aplicação.
- **Disponibilidade** - quando estamos perante a existência de vários servidores Mattermost, possuímos múltiplas réplicas às quais podemos recorrer. Tendo isto em conta, caso falhe algum servidor, o serviço continuará disponível.

2.2 Desvantagens

Tal como acabamos de ver, quando estamos na presença de vários servidores, a disponibilidade e performance são tidas como pontos positivos desta arquitetura. No entanto, dada a existência de um único *proxy*, estas podem também ser vistas como limitações:

- **Performance** - em cada momento existem inúmeros clientes a realizar pedidos que necessitam de ser atendidos por um só *proxy*, o que irá ter consequências em termos de performance.
- **Disponibilidade** - a existência de apenas um *proxy* proporciona uma baixa tolerância a falhas visto que, caso o *proxy* falhe, o serviço permanecerá indisponível até que seja encontrada uma solução.

3 Componentes

O Mattermost consiste num sistema constituído pelos seguintes componentes:

- **Servidor Mattermost** - O Servidor Mattermost inclui os diversos serviços da aplicação, constituindo assim a unidade principal de funcionamento da mesma.
- **Base de Dados Mattermost** - Uma base de dados MySQL (ou PostgreSQL) configurada em *stand-alone* ou com alta disponibilidade.
- **File Store** - Localização predefinida, onde são armazenados os ficheiros e as imagens que são anexadas às mensagens trocadas pelos vários utilizadores.
- **Mattermost Push Notification Server** - Serviço hospedado ou guardado localmente que apresenta notificações nas aplicações do Mattermost para Android e para iOS.
- **Aplicações cliente** - As aplicações para o cliente estão disponíveis para o Mattermost nas seguintes plataformas: Android, iOS, Windows, OSX e Linux (versão beta).
- **Cliente Web** - Os utilizadores do Mattermost podem também usar o seu navegador padrão para se conectarem ao servidor Mattermost.
- **Servidor Proxy** - É recomendado o uso de um servidor *proxy* (Nginx, Apache...) com o Mattermost.

No diagrama seguinte, será apresentada uma ilustração de alto nível da plataforma Mattermost, assim como a interação entre os vários componentes dentro de uma rede:

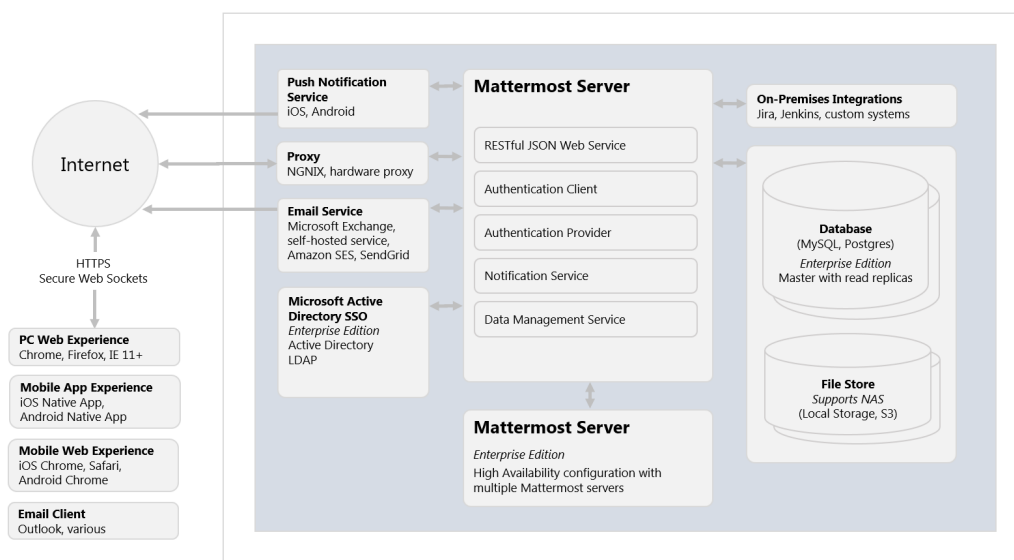


Figura 2: Diagrama da rede do Mattermost.

Retirado de https://docs.mattermost.com/getting-started/implementation_plan.html.

4 Padrões de Distribuição

Na figura 2 estão presentes 4 servidores Mattermost e 5 réplicas de leitura. No entanto, dependendo do desempenho que pretendemos obter, o número de servidores *proxy*, servidores Mattermost e réplicas de leitura é variável, isto é, o padrão de distribuição utilizado difere.

No caso do Mattermost existem três padrões evidentes, os quais serão apresentados de seguida.

4.1 Padrão básico

Quando antecipamos a utilização da aplicação para, no máximo, 200 utilizadores, não é necessária a participação de um servidor *proxy*, existindo consequentemente apenas um servidor *Mattermost*. Nestas situações estamos perante uma arquitetura Cliente-Servidor e não uma arquitetura Servidor *Proxy*.

4.2 Padrão da *Team Edition*

Nos casos em que se prevê um número reduzido de clientes a aceder à aplicação, ou cujo desempenho não é um fator tão preocupante, é desnecessário estar a desperdiçar recursos. Sendo assim, nestas situações basta recorrer a uma versão mais simples da arquitetura Servidor *Proxy*. Na *Team Edition* do Mattermost o padrão utilizado consiste apenas num *proxy*, num servidor Mattermost e numa base de dados.

4.3 Padrão da *Enterprise Edition*

Quando pretendemos melhorar a performance da nossa plataforma, permitindo o acesso a um maior número de utilizadores em simultâneo e uma diminuição da latência, basta aumentar os recursos utilizados e posicioná-los estrategicamente.

Tomemos como exemplo o teste de carga feito à *Enterprise Edition* do Mattermost, com 60,000 utilizadores ativos em simultâneo. Neste caso recorreu-se a 6 servidores Mattermost em *cluster*, 3 servidores *proxy* e um servidor de base de dados com 5 réplicas de leitura, originando assim um padrão semelhante ao da figura 1.

Com este padrão de distribuição é possível existirem recursos instalados em diferentes localizações, localizações essas que se encontram mais próximas do utilizador, resultando numa diminuição da latência. Para além disso, as réplicas de leitura utilizadas funcionam também como *backups*, permitindo recuperar dados caso ocorra algum tipo de falha.

5 Formas de Comunicação

A comunicação feita entre o utilizador e o servidor Mattermost é feita recorrendo aos protocolos HTTPS e WSS, definindo assim o tipo de conexão estabelecida. Tendo isto em conta, existem dois tipos de conexão possíveis:

- **Conexão HTTPS** (*Secure Hypertext Transfer Protocol*)

Este tipo de conexões utiliza maioritariamente o protocolo criptográfico HTTPS, para renderizar páginas e aceder às funcionalidades principais da plataforma. Em situações de teste e configuração iniciais pode acontecer de serem utilizadas conexões HTTP, mas nunca em ambiente de produção.

- **Conexão WSS** (*Secure WebSocket Protocol*)

Ao contrário das conexões HTTPS, as conexões WSS disponibilizam interações em tempo real entre o cliente e o servidor, mais concretamente atualizações e notificações. Caso este tipo de conexões esteja indisponível, o sistema aparentará estar a funcionar, no entanto, deixarão de existir interações em tempo real. Assim, só recarregando a página é que teremos acesso a atualizações.

É importante mencionar que a principal diferença entre uma conexão HTTPS e uma WSS é que a segunda possui uma conexão persistente enquanto o cliente estiver conectado, e a HTTPS possui uma ligação intermitente visto que se conecta ao servidor apenas quando é efetuado um pedido.

6 Pontos de Configuração

Para instalar a aplicação Mattermost com sucesso é necessário efetuar determinadas configurações que permitem a correta conexão entre os diferentes componentes. Estas configurações serão especificadas de seguida.

6.1 Base de Dados

Para conectar a base de dados ao servidor Mattermost é essencial configurar devidamente o Sistema de Gestão de Base de Dados que estamos a utilizar.

- **MySQL**

Quando estamos na presença de MySQL basta criar um utilizador com o endereço IP do servidor Mattermost (`<ip_address>`):

```
create user 'mmuser'@'<ip_address>' identified by 'mmuser-password';
```

Para além disto, é também necessário comentar a linha `bind-address = 127.0.0.1` do ficheiro `/etc/mysql/mysql.conf.d/mysqld.cnf`.

- **PostgreSQL**

À semelhança de MySQL, também com PostgreSQL é necessário editar um ficheiro, neste caso `/etc/postgresql/9.5/main/postgresql.conf`. Neste ficheiro deve-se substituir a linha `#listen_addresses = 'localhost'` por `listen_addresses = '*'`, permitindo assim ao PostgreSQL atender todos os endereços IP atribuídos.

Assumindo que o servidor Mattermost e a base de dados estão em máquinas separadas, para terminar a configuração deste SGBD é necessário modificar o ficheiro `/etc/postgresql/9.5/main/pg_hba.conf`. Assumindo que `<ip_address>` é, mais uma vez, o endereço IP do servidor Mattermost, basta acrescentar a `pg_hba.conf` a linha `host all all <ip_address>/32 md5`.

6.2 Sistema *File Storage*

Os ficheiros e as imagens anexados às mensagens dos utilizadores não são armazenados na base de dados, mas sim num *file store*. Este *file store* pode corresponder ao *file system* local (definir *File Storage System* como *Local File System*) ou ao Amazon S3 (definir *File Storage System* como *Amazon S3*).

6.3 TLS no servidor Mattermost

Quando não é utilizado um servidor *proxy*, o protocolo TLS é configurado diretamente no servidor Mattermost, sendo necessário definir o seguinte:

- Definir *Listen Address* como *:443*
- Definir *Connection Security* como *TLS*
- Definir *Forward port 80 to 443* como *true*

Com isto o servidor recebe diretamente os pedidos HTTPS (porta 443) e HTTP (porta 80), provenientes do cliente.

6.4 Nginx

Outro ponto de configuração é o Nginx, ou seja, o servidor *proxy*. Este tem que se conectar ao servidor Mattermost e, para isso, é importante especificar os seguintes aspetos no ficheiro de configuração do Nginx:

```
upstream backend {
    server <ip_address>:8065;
    ...
}

server {
    listen 80;
    server_name    <domain_name>;
    ...
}
```

Desta forma, sendo *<ip_address>* o endereço IP do servidor Mattermost e *<domain_name>* o FQDN do *website* Mattermost, conectamos o servidor *proxy* ao servidor Mattermost e fazemos com que os pedidos sejam encaminhados dos clientes (porta 80) para o servidor.

6.5 SSL e HTTP/2 com Nginx

Ao configurar os protocolos SSL e HTTP/2 com o Nginx é necessário alterar o ficheiro de configuração mencionado anteriormente para o seguinte:

```
server {
    listen 80 default_server;
    server_name    <domain_name>;
    ...
}

server {
    listen 443 ssl http2;
    server_name    <domain_name>;
    ...
}
```

Neste caso *<domain_name>* tem o mesmo significado que em 6.4.

7 Operações com desempenho crítico

De seguida encontram-se especificados os recursos necessários para ser possível suportar a instalação do Mattermost *Enterprise Edition*:

Tabela 1: Requisitos para uma instalação recomendada do Mattermost *Enterprise Edition*.

Serviço	Quantidade	Processador	Memória	Armazenamento
Servidor Mattermost	2	2 vCPUs/cores	4 GB	45-90 GB
Servidor da BD	2	2 vCPUs/cores	8 GB	100 GB
<i>Proxy</i>	2	1 vCPUs/cores	4 GB	10 GB

7.1 Armazenamento insuficiente

A aplicação Mattermost necessita de bastante espaço de armazenamento para funcionar corretamente e com uma boa performance, como se pode verificar na tabela descrita acima. Deste modo, quando as máquinas que estão a correr cada um dos serviços (Mattermost, BD e *Proxy*) ficam sem espaço livre de armazenamento ou ficam com este reduzido, prejudicam o desempenho da aplicação.

7.2 Falha na base de dados

A base de dados é crucial numa aplicação, visto que é nesta que se encontram os dados relativos ao funcionamento da aplicação e respetivos utilizadores. Quando o servidor da base de dados falha, toda a aplicação Mattermost é afetada, podendo sofrer uma descida de desempenho ou, até mesmo, ficar “em baixo” até o problema ser resolvido.

7.3 Bombardeamento de pedidos

O Mattermost possui um ou mais servidores *proxy* responsáveis por atender os pedidos dos clientes e balancear a respetiva carga para os servidores disponíveis. Quando estes servidores são bombardeados por milhares de pedidos ao mesmo tempo, podem sofrer falhas e não conseguir processar todos os pedidos. Consequentemente, a performance do Mattermost será afetada.

8 Conclusão

Após uma primeira análise da aplicação Mattermost, foi-nos possível adquirir conhecimentos quer no que toca ao propósito da aplicação, quer em termos do seu aspeto físico e principais vulnerabilidades.

Uma vez identificadas as características do Mattermost e os componentes que o constituem e aplicando as configurações necessárias, conseguimos efetuar uma instalação bem sucedida da aplicação, conhecendo, assim, os pontos cruciais desta operação. Para além disto, após prever quais os aspetos críticos que representam alvos vulneráveis do desempenho da plataforma, ao planear a nossa própria arquitetura podemos ter em conta estes aspetos, podendo utilizá-los, futuramente, como casos de estudo.

Para tornar possível a concretização da segunda etapa deste projeto, foi imprescindível a realização do estudo apresentado ao longo deste documento, dado que agora possuímos pleno conhecimento acerca da aplicação, tornando muito mais simples a automatização do processo de *deployment* e *benchmarking* que se segue.