



**Ciências
ULisboa**

Faculdade
de Ciências
da Universidade
de Lisboa

Faculdade de Ciências da Universidade de Lisboa

Departamento de Informática

Mestrado em Engenharia Informática

Relatório

Configuração e Gestão de Sistemas

Performance Estimate

Aluno: **Rodrigo Craveiro Rodrigues (fc64370)**

Professor: **Doutor Hugo Miranda**

2º Semestre Letivo 2024/2025

maio 2025

Índice

1. Introdução.....	3
2. Pressupostos.....	3
3. Arquitetura do Sistema.....	3
4. Metodologia.....	4
5. Análise de Capacidade	4
5.1 Conversão para Unidades Comuns	4
5.2 Cálculo da Capacidade a 30% de Utilização.....	5
5.3 Cálculo da Procura por Visualização de Página	5
5.4 Cálculo Final da Capacidade Máxima	6
6. Redundância para Duplicar Capacidade	6
7. Análise Complementar	7
7.1 Lei de Little Aplicada ao Sistema	7
7.2 Lei do Fluxo Forçado.....	7
7.3 Teoria das Filas	8
7.4 Análise de Resiliência.....	9
7.5 Análise de Capacidade em Condições de Pico.....	9
8. Avaliação	10
9. Conclusão.....	10
10. Referências	11

1. Introdução

Este relatório analisa uma **infraestrutura web** organizada em várias camadas para determinar o **débito máximo em visualizações de página por segundo (pv/s)**, garantindo que nenhum dos componentes ultrapasse **30% de utilização**. Em seguida, propõe-se um **plano de redundância** para **duplicar** esta capacidade mantendo a margem de segurança operacional, incorporando avaliações ao recorrer a **métricas e leis fundamentais de desempenho** do sistema.

2. Pressupostos

- Cada visualização de página gera **60 pedidos HTTP** e transfere **5 MB** de dados para o cliente (aproximadamente **83 KB por pedido**).
- Dos **60 pedidos HTTP**:
 - **30%** são para conteúdos estáticos (logotipos, CSS, imagens), tratados pelo sistema de cache.
 - **70%** são pedidos dinâmicos:
 - **10%** correspondem a operações de escrita na DB.
 - **50%** a operações de leitura na DB.
 - **10%** correspondem a outras operações dinâmicas (*overhead* de aplicação).
- Todos os componentes estão configurados em **modo ativo**, pelo que as suas capacidades somam linearmente.
- Objetivo de **utilização máxima de 30% da capacidade** nominal de cada componente.
- **Tempo médio de resposta** estimado para um pedido ao servidor web: **0,5 segundos**.
- **Disponibilidade base** de cada componente individual: **99,9%**.

3. Arquitetura do Sistema

O sistema é composto pelos seguintes componentes:

- **Camada de Segurança: 2 firewalls** (perímetro e interno), cada um com capacidade de **250 Mbps**.
- **Camada de Distribuição de Carga: 2 load balancers**, cada um capaz de processar **10.000 pedidos/min**.

- **Camada de Cache:** 3 *web caches*, cada uma capaz de processar **8.000 pedidos/min.**
- **Camada de Aplicação:** 10 **servidores web**, cada um capaz de processar **120 pedidos/min.**
- **Camada de Dados:** 5 **servidores de BD**, cada um com capacidade de **1 transação de escrita/seg** e **10 transações de leitura/seg.**

4. Metodologia

1. **Converter** todas as capacidades para **unidades p/segundo**.
2. **Calcular o valor** correspondente a **30% da capacidade** nominal de cada componente.
3. **Determinar quantos pedidos/megabits p/seg** cada componente consegue processar a **30%.**
4. **Dividir o valor pela "procura"** (número de pedidos/megabits exigidos por visualização de página) para **obter o débito em pv/s.**
5. **Identificar o *bottleneck*:** o componente com menor capacidade em pv/s.
6. **Aplicar leis fundamentais de performance** (Lei de Little, Lei do Fluxo Forçado) para análise complementar.
7. **Realizar análise de resiliência e comportamento** em condições de pico.

5. Análise de Capacidade

5.1 Conversão para Unidades Comuns

Componente	Capacidade Original	Conversão	Capacidade (p/seg)
Firewalls	2 x 250 Mbps		500 Mbps
Load Balancer 1	2 x 10.000 req/m	÷ 60	333,33 req/s
Web Caches	3 x 8.000 req/m	÷ 60	400 req/s
Load Balancer 2	2 x 10.000 req/m	÷ 60	333,33 req/s
Servidores Web	10 x 120 req/m	÷ 60	20 req/s
BD (Escrita)	5 x 1 w/s		5 w/s

BD (Leitura)	5 x 10 r/s		50 r/s
---------------------	------------	--	--------

5.2 Cálculo da Capacidade a 30% de Utilização

Para cada componente, podemos calcular o **limite de 30% da sua capacidade máxima**:

Componente	Capacidade (p/segundo)	30% da Capacidade
Firewalls Externos	500 Mbps	150 Mbps
Load Balancer 1	333,33 req/s	100 req/s
Web Caches	400 req/s	120 req/s
Firewalls Internos	500 Mbps	150 Mbps
Load Balancer 2	333,33 req/s	100 req/s
Servidores Web	20 req/s	6 req/s
BD (Escrita)	5 w/s	1,5 w/s
BD (Leitura)	50 r/s	15 r/s

5.3 Cálculo da Procura por Visualização de Página

Podemos determinar quanto **cada visualização de página** exige de cada componente:

Componente	Cálculo da Procura	Procura p/Page-view
Firewalls Externos	5 MB x 8 bits/byte	40 Mb
Load Balancer 1	Todos os pedidos	60 req
Web Caches	30% dos pedidos	18 req
Firewalls Internos	70% x 5 MB x 8	28 Mb
Load Balancer 2	70% dos pedidos	42 req
Servidores Web	70% dos pedidos	42 req
BD (Escrita)	10% dos pedidos	6 req
BD (Leitura)	50% dos pedidos	30 req

5.4 Cálculo Final da Capacidade Máxima

Com base nos cálculos anteriores, podemos determinar **quantas visualizações de página p/seg** cada componente suporta a **30% de utilização**:

Componente	30% da Capacidade	Procura p/Page-view	Máximo Page-views/s
Firewalls Externos	150 Mbps	40 Mb	$150/40 = 3,75$
Load Balancer 1	100 req/s	60 req	$100/60 = 1,67$
Web Caches	120 req/s	18 req	$120/18 = 6,67$
Firewalls Internos	150 Mbps	28 Mb	$150/28 = 5,36$
Load Balancer 2	100 req/s	42 req	$100/42 = 2,38$
Servidores Web	6 req/s	42 req	$6/42 = 0,143$
BD (Escrita)	1,5 w/s	6 req	$1,5/6 = 0,25$
BD (Leitura)	15 r/s	30 req	$15/30 = 0,50$

O **bottleneck** do sistema são claramente os **servidores web**, limitando a capacidade total a **0,143 visualizações de página p/seg** (aproximadamente **8,6 visualizações p/min**).

6. Redundância para Duplicar Capacidade

Para atingir aproximadamente **0,286 pv/s** (o **dobro da capacidade atual**) **sem exceder 30% de utilização** em nenhum componente:

Componente	Capacidade atual (pv/s)	Capacidade desejada (pv/s)	Capacidade necessária	Ação recomendada
Servidores Web	0,143	0,286	20 servidores (10 para 20)	Adicionar 10 servidores
BD (Escrita)	0,25	0,286	6 servidores (5 para 6)	Adicionar 1 servidor

Os componentes **firewalls**, **caches**, **load balancers** e **réplicas de leitura da BD** mantêm **folga significativa** mesmo a **30% de utilização** e **não** requerem alterações para satisfazer o objetivo de **duplicação da capacidade**.

7. Análise Complementar

7.1 Lei de Little Aplicada ao Sistema

A **Lei de Little** estabelece que o número médio de pedidos pendentes num sistema é igual ao produto do *throughput* pelo tempo médio de resposta:

$$N = \lambda \times R$$

Onde:

- ***N*** é o número médio de pedidos no sistema.
- ***λ*** é o *throughput*.
- ***R*** é o tempo médio de resposta.

Para os **servidores web** (componente limitante), com um tempo médio de resposta estimado de 0,5 segundos:

$$N = 6 \text{ req/s} \times 0,5\text{s} = 3 \text{ pedidos}$$

Isto significa que, no cenário de **30% de utilização**, há uma **média de 3 pedidos** a serem processados simultaneamente pelos **servidores web**. Para outros componentes:

Componente	Throughput (pedidos/s)	Tempo médio estimado (s)	Número médio de pedidos
Load Balancer 1	100	0,01	1
Web Caches	120	0,01	1,2
Load Balancer 2	100	0,01	1
BD (Escrita)	1,5	0,1	0,15
BD (Leitura)	15	0,05	0,75

7.2 Lei do Fluxo Forçado

A **Lei do Fluxo Forçado** indica que o *throughput* através de diferentes componentes é proporcional ao número de vezes que cada componente necessita de processar cada pedido.

Com um *throughput* do sistema de **0,143 pv/s**, podemos calcular o ***throughput* efetivo** em cada componente:

Componente	Throughput efetivo
Firewalls Externos	$0,143 \times 60 = 8,58 \text{ req/s}$
Load Balancer 1	$0,143 \times 60 = 8,58 \text{ req/s}$
Web Caches	$0,143 \times 18 = 2,57 \text{ req/s}$
Firewalls Internos	$0,143 \times 42 = 6,01 \text{ req/s}$
Load Balancer 2	$0,143 \times 42 = 6,01 \text{ req/s}$
Servidores Web	$0,143 \times 42 = 6,01 \text{ req/s}$
BD (Escrita)	$0,143 \times 6 = 0,858 \text{ w/s}$
BD (Leitura)	$0,143 \times 30 = 4,29 \text{ r/s}$

Estes valores confirmam que **nenhum componente** está a operar **acima dos limites de 30% de utilização** calculados anteriormente.

7.3 Teoria das Filas

Aplicando a **teoria das filas**, podemos modelar os **servidores web** como um **sistema M/M/10** (chegadas Poisson, tempo de serviço exponencial, 10 servidores):

- Taxa de chegada (λ): 6 req/s.
- Taxa de serviço por servidor (μ): 0,033 req/s (2 req/min).
- Utilização ($\rho = \lambda/(\mu \times 10)$): $6 / (0,033 \times 10) = 18,18\%$

Para um sistema **M/M/c** com utilização ρ , o **tempo médio de espera na fila** é dado por:

$$W_q = (P_0 \times (\lambda/\mu)^c \times \rho) / (c! \times c \times \mu \times (1-\rho)^2)$$

Onde P_0 é a probabilidade de o sistema estar vazio.

Num cenário de **baixa utilização (18,18%)**, o **tempo médio** de espera na fila seria **próximo de zero**, indicando que os pedidos são processados praticamente sem atraso.

Se considerarmos um modelo **M/D/10** (**tempo de serviço determinístico**):

- O **tempo médio de espera** seria ainda **menor** do que no modelo **M/M/10**, demonstrando a vantagem de tempos de serviço previsíveis.

7.4 Análise de Resiliência

A **resiliência do sistema** pode ser avaliada calculando a **disponibilidade total**, considerando as configurações em **série** e **paralelo**:

Disponibilidade das Componentes em Paralelo:

Assumindo uma **disponibilidade individual** de **99,9%** para cada componente:

Componente	Cálculo	Disponibilidade
Firewalls (2)	$1 - (0,001)^2$	99,9999%
Load Balancers (2)	$1 - (0,001)^2$	99,9999%
Web Caches (3)	$1 - (0,001)^3$	99,9999%
Servidores Web (10)	$1 - (0,001)^{10}$	99,9999%
BD (5)	$1 - (0,001)^5$	99,9995%

Disponibilidade Total do Sistema (Componentes em Série):

Disponibilidade = $0,999999 \times 0,999999 \times 0,999999 \times 0,999999 \times 0,999995 = 99,999\%$

Isto indica que o sistema tem uma **disponibilidade de "cinco noves"** (99,999%), equivalente a aproximadamente **5,26 minutos** de **indisponibilidade p/ano**.

A arquitetura do sistema, com múltiplos componentes em **paralelo** em cada camada, proporciona uma **elevada resiliência**, mesmo que algum componente individual falhe.

7.5 Análise de Capacidade em Condições de Pico

Utilizando a **regra dos 3 σ** , podemos dimensionar o sistema para lidar com **picos de tráfego**:

Se assumirmos que o tráfego segue uma **distribuição normal** com:

- **Média:** 0,143 pv/s
- **Desvio padrão estimado:** 0,05 pv/s.

O sistema deve ser **capaz de lidar** com: $0,143 / 3 \times 0,05 = 0,293 \text{ pv/s}$ (aprox. 17,6 pv/min)

Esta análise reforça a recomendação de **duplicar a capacidade** para aproximadamente **0,286 pv/s**, garantindo que o sistema possa lidar com **picos de tráfego sem exceder os limites de utilização** estabelecidos.

8. Avaliação

Para além da análise fundamental realizada, podemos identificar aspetos importantes a considerar:

1. **Desequilíbrio na utilização dos componentes:** O sistema apresenta um **desequilíbrio significativo** na utilização dos seus componentes. Enquanto os **servidores web** funcionam a **30% da sua capacidade** no cenário de capacidade máxima, **outros componentes** operam com **utilização muito abaixo** deste valor:
 - **Firewalls:** 3,8% da capacidade máxima.
 - **Web Caches:** 2,1% da capacidade máxima.
 - **Load Balancers:** 8,6% e 6% da capacidade máxima.

Isto sugere um possível **sobredimensionamento** destes componentes ou a necessidade de **reconfigurar** o sistema para **melhor balanceamento de recursos**.

2. **Otimização de custos:** Considerando o desequilíbrio identificado, poderia ser mais económico reduzir o número de componentes sobredimensionados e investir em mais servidores web para aumentar a capacidade geral do sistema.
3. **Monitorização dinâmica:** Implementar um sistema de monitorização em tempo real que aplique as métricas discutidas (**throughput**, **utilização**, **tempo de resposta**) para ajustar recursos dinamicamente seria benéfico para otimizar o desempenho do sistema.
4. **Testes de carga:** Seria recomendável realizar testes simulando diferentes padrões de tráfego para validar se o comportamento real do sistema corresponde às previsões teóricas baseadas nas leis de performance aplicadas.
5. **Estratégia de escalonamento:** Definir gatilhos de escalonamento automático baseados nas métricas de utilização, tempo de resposta e *throughput* para otimizar custos e desempenho em função das necessidades reais.

9. Conclusão

- **Capacidade atual:** 0,143 pv/s (aprox. 8,6 visualizações de página p/min) com todos os componentes a $\leq 30\%$ de utilização.
- **Plano de redundância:**
 - **Servidores Web:** 10 para 20 (acrécimo de 10).
 - **Servidores BD para Escrita:** 5 para 6 (acrécimo de 1).

Esta configuração garante capacidade para **0,286 pv/s** (aprox. 17,2 visualizações de página p/min), mantendo a margem de segurança operacional (máximo de **30% de utilização**).

A aplicação das **leis fundamentais de performance (Lei de Little, Lei do Fluxo Forçado)** e a **análise de resiliência e capacidade em condições de pico** reforçam a adequação da solução proposta, validando que o sistema:

1. Terá **capacidade suficiente** para lidar com **picos de tráfego**.
2. Manterá **tempos de resposta** adequados.
3. Apresentará **alta disponibilidade (99,999%)**.
4. **Distribuirá a carga** de forma **eficiente** entre os componentes.

Esta proposta assegura não só a **duplicação da capacidade** como também uma **estrutura equilibrada** em termos de **redundância N+1** para todos os componentes críticos do sistema, garantindo uma operação **eficiente e escalável**.

10. Referências

- [1] Moodle 2024/2025, Configuração e Gestão de Sistemas, Prof. Hugo Miranda:
https://moodle.ciencias.ulisboa.pt/pluginfile.php/569945/mod_resource/content/1/t130-slides.pdf