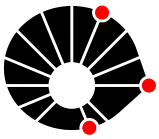

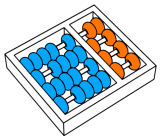

Gerência de Recursos em Sistemas Distribuídos

Emanuel De Souza Oliveira *Gabriel Braga Proença*
Lucca Costa Piccolotto Jordão *Luiz Fernando Bittencourt*
Marcelo Claudio Sousa Araújo

Relatório Técnico - IC-PFG-23-31
Projeto Final de Graduação
2023 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Gerência de Recursos em Sistemas Distribuídos

Emanuel De Souza Oliveira Gabriel Braga Proença
Lucca Costa Piccolotto Jordão Luiz Fernando Bittencourt
Marcelo Claudio Sousa Araújo

Resumo

Este projeto analisou a instanciação de serviços na nuvem, focando no Google Cloud Platform (GCP) [1]. Utilizou cálculos de fatorial e Fibonacci para benchmarking em diversas configurações de instância, avaliando desempenho e custo. A pesquisa destacou a importância de selecionar configurações otimizadas para equilibrar eficácia e custo, especialmente com orçamentos limitados. Observou-se que configurações distribuídas melhoram o desempenho, mas aumentam custos, com destaque para a eficiência do Cloud Run[1].

1 Introdução

Neste trabalho de conclusão de curso, nos debruçamos sobre as práticas de instanciação de serviços na nuvem, com foco particular no Google Cloud Platform (GCP). Este estudo é motivado pela crescente demanda por soluções eficientes e econômicas na computação em nuvem, um campo que se tornou indispensável para a infraestrutura tecnológica moderna. Em um mundo onde os dados são um ativo valioso, a capacidade de processar e manipular esses dados de forma eficiente e econômica é crucial.

O GCP, como uma das plataformas líderes de serviços de nuvem, oferece uma variedade de opções para a instanciação de serviços, cada uma com suas características específicas de desempenho e custo. Com a diversidade de serviços disponíveis, surge a questão de como escolher a configuração mais adequada para necessidades específicas, equilibrando eficácia operacional e eficiência de custos.

Nossa abordagem envolve a análise de diferentes configurações de instanciação no GCP, utilizando como base funcionalidades computacionais comuns - especificamente, o cálculo do fatorial e da sequência de Fibonacci. Estas funções foram escolhidas por sua simplicidade e capacidade de demonstrar eficazmente a carga de trabalho computacional. O objetivo é entender como diferentes abordagens de instanciação afetam o desempenho e o custo, fornecendo insights valiosos para a otimização de recursos em ambientes de nuvem.

Este estudo aborda um aspecto fundamental da computação em nuvem, que é a escolha inteligente de recursos e configurações. Ao analisar as variações de desempenho e custo nas diferentes configurações do GCP, buscamos contribuir para um planejamento mais eficiente e uma gestão de recursos mais econômica em projetos de tecnologia da informação.

2 Metodologia

Neste estudo, investigamos diferentes abordagens de instanciação de serviços na nuvem no Google Cloud Platform (GCP), com um enfoque na análise de desempenho e custo. Para todos os

[1]Cloud Run é uma plataforma serverless de computação gerenciada que permite executar contêineres na nuvem. [2]

testes, adotamos uma metodologia consistente de realizar 100 chamadas por segundo ao front-end que consequentemente chama as duas funções, durante 12 horas. Utilizamos duas funções computacionais específicas - cálculo do fatorial e da sequência de Fibonacci - como base para nossas análises.

2.1 Aplicação monolítica em VM

Inicialmente, configuramos uma Máquina Virtual (VM)[2] monolítica no GCP. Esta VM serviu como um ambiente unificado para hospedar tanto o front-end quanto o back-end das aplicações. O front-end, responsável pela interface de usuário, fazia chamadas contínuas para as funções de fibonacci e fatorial, que eram processadas no back-end hospedado na mesma VM. Esta configuração visava entender o desempenho e os custos associados a um ambiente de computação em nuvem centralizado.

2.2 Front-end em VM e FaaS Separados

Em seguida, separamos o front-end e o back-end. O front-end foi mantido em uma VM, enquanto as funções de cálculo foram migradas para a arquitetura de Function as a Service (FaaS)[3], operando cada função em uma FaaS diferente. O front-end fazia chamadas regulares para essas funções FaaS, agora hospedadas de forma independente. O foco aqui era avaliar o impacto da separação do front-end e do back-end em termos de desempenho e custo.

2.3 Front-end em Cloud Run e FaaS Separados

Implementamos o front-end no serviço Cloud Run do GCP, mantendo as funções em FaaS distintas. As chamadas eram feitas do Cloud Run para as funções hospedadas em diferentes locais. Esta etapa visava explorar a eficiência de um ambiente de nuvem mais distribuído e a escalabilidade oferecida pelo Cloud Run.

2.4 Front-end em VM com FaaS Unificada

O front-end foi novamente hospedado em uma VM, mas desta vez, ele chamava uma única FaaS que continha ambas as funções. Este arranjo testou a eficiência de um ponto de acesso unificado para múltiplas funções. Analisar a eficácia de uma FaaS unificada em comparação com funções separadas em termos de gerenciamento de carga e custo.

2.5 Front-end em Cloud Run com FaaS Unificada

Similar à configuração anterior, mas com o front-end hospedado no Cloud Run. Este teste proporcionou uma comparação direta com a configuração anterior, mas em um ambiente de Cloud Run. Avaliar os benefícios de desempenho e custo do Cloud Run combinado com uma FaaS unificada.

[2]Uma máquina virtual (VM) é uma versão digital de um computador físico. O software de máquina virtual pode executar programas e sistemas operacionais, armazenar dados, conectar-se a redes e executar outras funções de computação. Além disso, ele exige manutenção, como atualizações e monitoramento de sistema. [3]

[3]A FaaS é um serviço de back-end sem servidor que permite aos desenvolvedores escrever peças modulares de código em tempo real que podem ser executadas em resposta a determinados eventos. [4]

2.6 Front-end e Back-end Separados em Cloud Run

Ambos o front-end e o back-end foram configurados separadamente em instâncias do Cloud Run. Esta configuração permitiu uma análise da comunicação e eficiência entre dois serviços independentes no Cloud Run. Investigar a eficácia operacional e a escalabilidade de serviços completamente distribuídos no Cloud Run.

2.7 Front-end e Back-end em VMs Separadas

Nesta etapa, tanto o front-end quanto o back-end foram configurados em Máquinas Virtuais separadas. Esta abordagem ofereceu uma visão do desempenho em um ambiente mais tradicional de nuvem, com componentes separados em Máquinas Virtuais distintas. Compreender as implicações de desempenho e custo em um cenário de nuvem com separação física dos componentes.

2.8 Aplicação monolítica em Cloud Run

Por fim, instanciamos tanto o front-end quanto o back-end no Cloud Run. Esta configuração representou um cenário de nuvem totalmente gerenciado e automatizado, utilizando os serviços do Cloud Run. Avaliar as vantagens de um ambiente totalmente gerenciado em termos de manutenção, desempenho e custo.

Após a execução dessas configurações, procedemos com uma análise detalhada das métricas coletadas, incluindo capacidade de CPU, tempo de resposta e custos associados, permitindo uma comparação abrangente da eficiência e eficácia de custos nas diversas abordagens de instanciação no GCP.

3 Resultados

Vamos apresentar os resultados obtidos em cada uma das configurações testadas, considerando o tempo de resposta, o uso médio da CPU e os custos associados. Vale deixar claro que as requisições feitas pelo Postman [5], de 100 chamadas por segundo durante 12 horas, foram feitas de forma manual habilitando as chamadas de hora em hora para durarem exatamente uma hora, assim, todos os gráficos referentes ao uso da CPU (em VM ou Cloud Run) e requisições por segundo recebida pelas FaaS têm picos e vales, respectivamente, de hora em hora, justificando os atrasos ao habilitar o serviço no Postman a cada ciclo de uma hora concluído.

A Figura 1 é o gráfico que mostra o resultado do faturamento após a execução dos testes. Vale salientar que os primeiros dias foram cobrados os preços pelos serviços estarem funcionando, mas não foram executados testes, assim não houve sobrecarga nas aplicações. Esses preços serão subtraídos dos custos associados a cada dia de teste para se obter uma média mais fiel dos gastos reais.

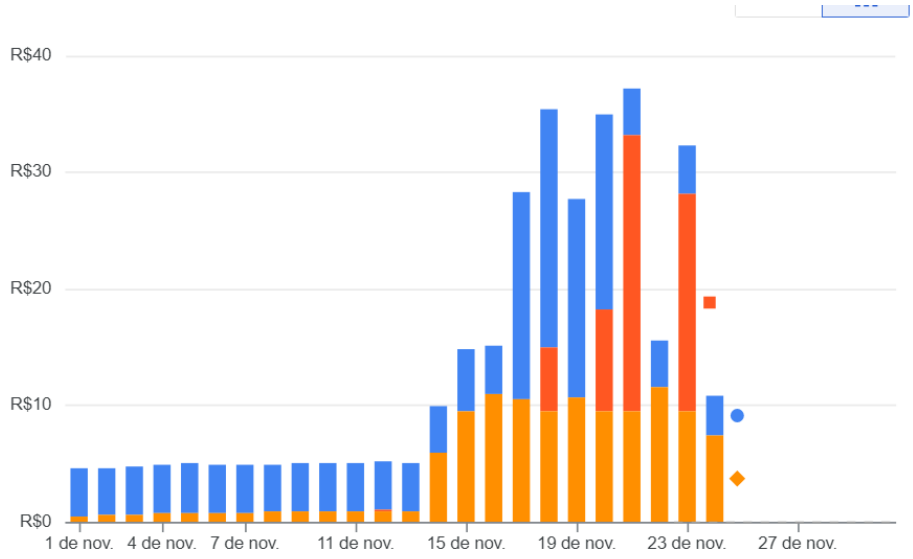

Figura 1: Faturamento durante o período completo das aplicações hospedadas e em teste

3.1 Aplicação monolítica em VM

A Figura 2 é um gráfico de amostra das requisições por segundo feitas pelo Postman aos serviços de front-end e back-end para a aplicação monolítica em VM. O tempo médio de resposta, representado pela linha azul, mostra um aumento gradual ao longo do tempo, o que pode indicar que o sistema está começando a acumular um certo atraso na resposta à medida que a carga de trabalho continua. No entanto, não vemos picos significativos ou variações abruptas, o que sugere que a aplicação está lidando com as requisições de forma consistente.

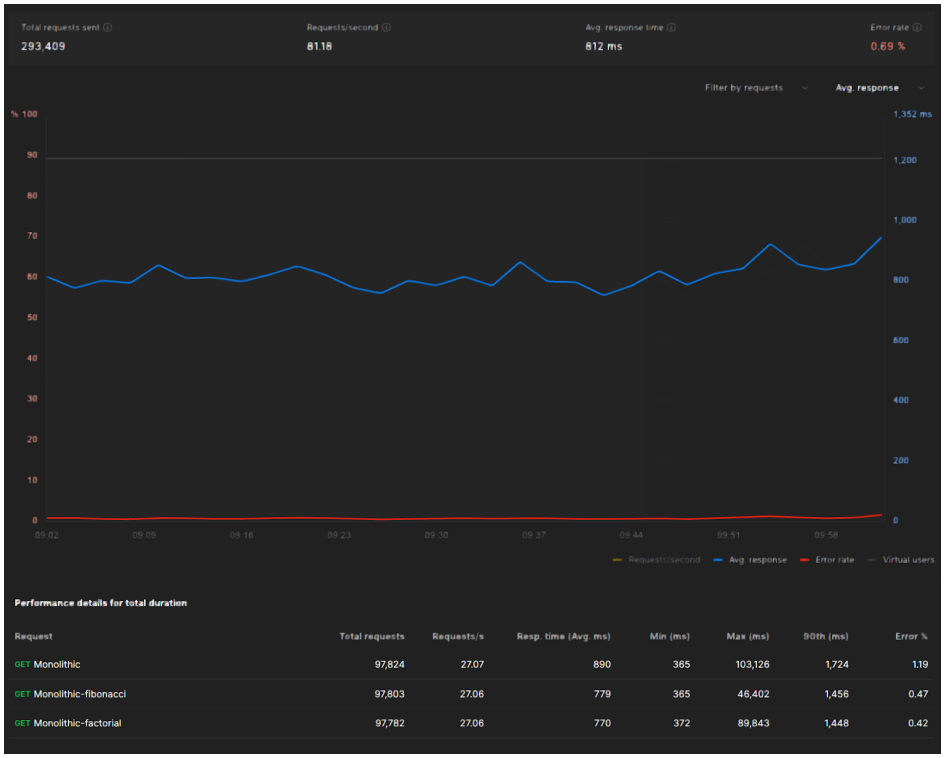

Figura 2: Requisições enviadas para aplicação com VM monolítica.

Ainda na Figura 2, as tabelas na parte inferior fornecem uma visão detalhada do desempenho das duas operações que estão sendo avaliadas: Fibonacci e Fatorial. Ambas têm um tempo médio de resposta similar (aproximadamente 770ms para Fatorial e 779ms para Fibonacci). O que chama atenção são os tempos de resposta máximos registrados, que são muito elevados, ultrapassando 46s para Fibonacci, 89s para Fatorial e 103s para o front-end. Esses valores são exceções, mas indicam que podem haver casos onde a resposta é muito mais lenta do que o normal.

A taxa de erro de 1,19%, mostra-se baixa, porém existente, indicando que a maioria das requisições está sendo processada com sucesso sem erros significativos. Isso é positivo, pois mostra que, apesar da carga, a aplicação é confiável e está lidando bem com as solicitações recebidas.

A Figura 3 mostra o consumo da CPU da VM durante as 12 horas de requisições, podendo notar alguns picos e alguns vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. O uso médio foi de 78,5%, mostrando uma alta demanda de processamento para essa abordagem monolítica, justificando as taxas de erro, tempo médio de resposta elevado e tempos máximos de resposta nas requisições.

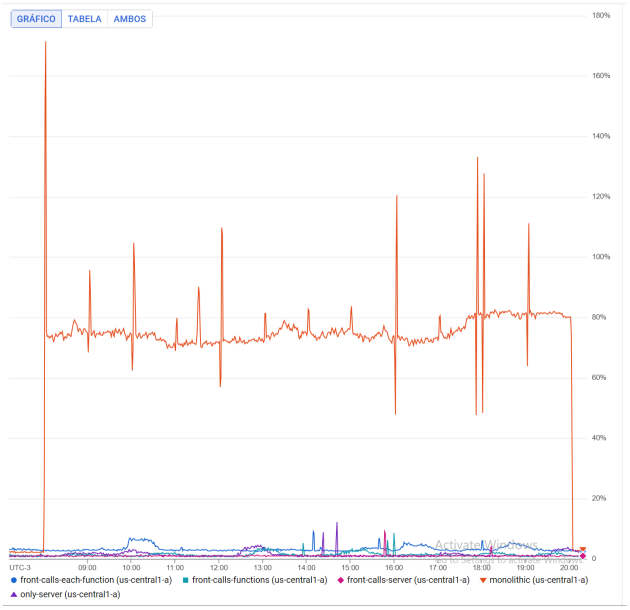

Figura 3: Consumo de CPU em aplicação com VM monolítica

A Figura 4 mostra o faturamento detalhado para o dia de realização dos envios das requisições para a aplicação monolítica em VM. Os custos foram unicamente da Compute Engine [6], R$ 5,15, subtraídos os valores médios gastos pelos demais serviços estarem apenas hospedados e operantes. Tal resultado reflete o uso intensivo de recursos da VM.

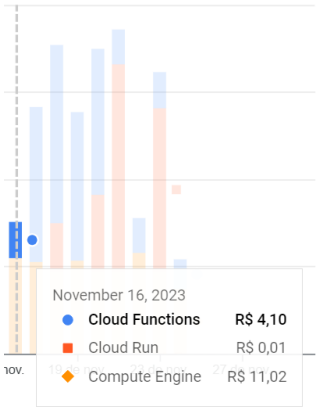

Figura 4: Faturamento para o dia que foram utilizados os serviços da aplicação monolítica em VM

3.2 Front-end em VM e FaaS Separados

A Figura 5 é um gráfico de amostra das requisições feitas pelo Postman aos serviços de front-end e back-end para a aplicação com front-end em VM e FaaS separadas. O tempo médio de resposta, representado pela linha azul, mostra uma queda inicial acentuada no número de requisições por segundo, que se estabiliza rapidamente. Esta queda pode ser devida ao "aquecimento"inicial da aplicação, onde a primeira carga de requisições é processada e, em seguida, um estado de equilíbrio é alcançado. A linha então se mantém constante, indicando que, após o período inicial, o sistema é capaz de lidar com um fluxo estável de requisições com um desempenho consistente.

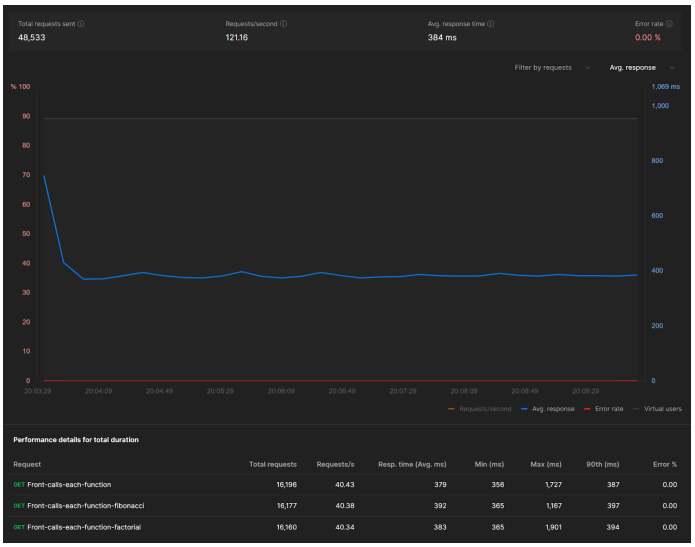

Figura 5: Requisições enviadas para aplicação com front-end em VM e FaaS separados

O tempo médio de resposta, representado pela linha azul, permanece estável próximo de 384 ms durante o período registrado, sugerindo que o sistema está respondendo bem sob a carga de trabalho atual. Esta estabilidade na resposta é um bom indicador de que a infraestrutura está dimensionada adequadamente para o volume de requisições recebidas.

A taxa de erro é de 0%, o que é ideal e indica que todas as requisições foram atendidas com sucesso sem falhas registradas. Também os tempos máximos de respostas são aceitáveis, isso sugere que tanto o front-end na VM quanto as FaaS estão funcionando corretamente e que a integração entre eles é robusta.

A Figura 6 mostra o consumo da CPU pela VM do front-end durante as 12 horas de requisições, podendo notar alguns vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. O uso médio da CPU foi mais baixo, cerca de 25%, mostrando uma carga muito menor em relação à aplicação monolítica.

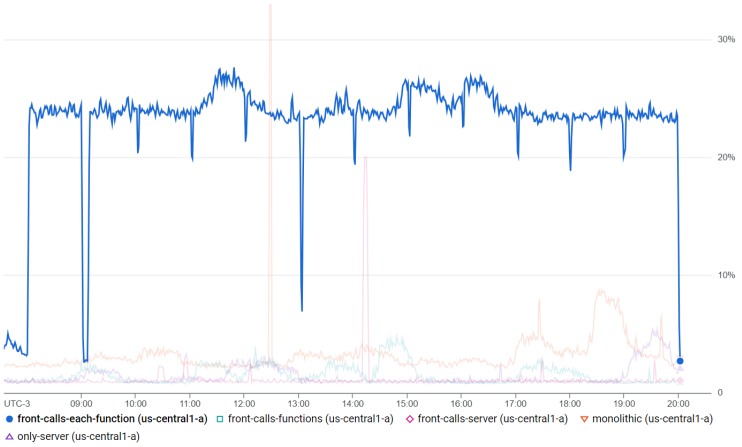

Figura 6: Consumo de CPU em aplicação com front-end em VM

A Figura 7 apresenta o número de requisições que os dois FaaS recebem conjuntamente durante as 12 horas de teste vindas do front-end em VM, sendo válido salientar que existem vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. É aparente uma regularidade de 87 chamadas por segundo durante todo o período de teste, impactando no preço final para o uso do serviço de FaaS.

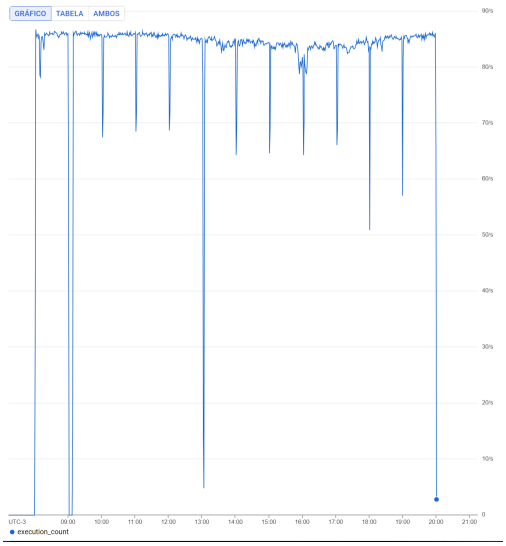

Figura 7: Quantidade de requisições que o FaaS recebe por segundo do front-end em VM

A Figura 8 mostra o tempo médio de execução do FaaS para responder às requisições feitas pelo front-end em VM durante as 12 horas do período de teste. Há uma constância em 5ms para tempo de resposta, sabendo que o tempo de execução do FaaS também impacta no preço final.

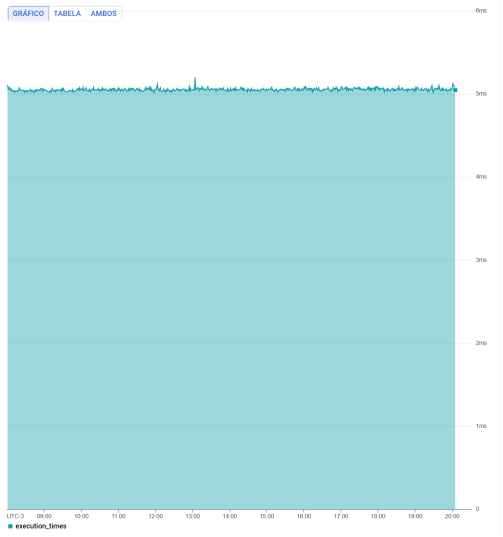

Figura 8: Tempo de execução do FaaS para responder às requisições feitas pelo front-end em VM

A Figura 9 exibe o faturamento detalhado para o dia de realização dos envios das requisições para a aplicação com front-end em VM e dois FaaS separados. Os custos foram divididos entre

Compute Engine, R$4,60, e Cloud Functions [7], R$16,41, subtraídos os valores médios gastos pelos demais serviços estarem apenas hospedados e operantes. Tal resultado reflete o menor uso da VM em relação ao modelo anterior e um uso intenso dos FaaS.

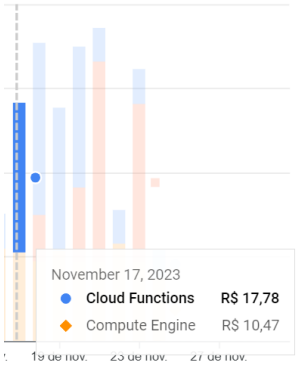

Figura 9: Faturamento para o dia que foram utilizados os serviços da aplicação com front-end em VM e FaaS separados

3.3 Front-end em Cloud Run com FaaS Separados

A Figura 10 é um gráfico de amostra das requisições por segundo feitas pelo Postman aos serviços de front-end em Cloud Run e back-end em FaaS separados. O tempo médio de resposta, representado pela linha azul, está estável em torno de 400 ms, o que é um indicativo de bom desempenho, considerando a quantidade de requisições processadas. Além disso, a taxa de erro está em 0%, o que é um sinal excelente de confiabilidade do sistema.

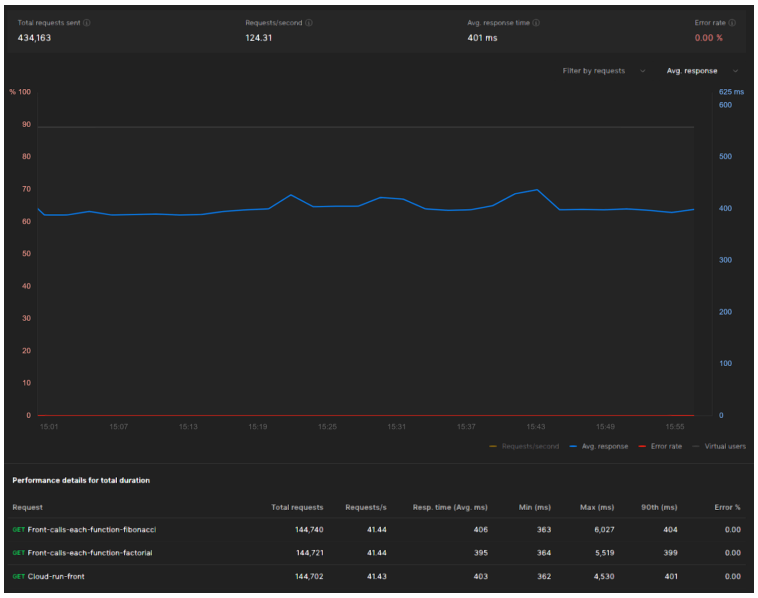

Figura 10: Requisições enviadas para aplicação com front-end em Cloud Run e FaaS separados

Ainda na Figura 10, as tabelas na parte inferior fornecem uma visão detalhada do desempenho das requisições, chamando atenção o tempo máximo de 4,5 segundos para o front-end e 5,5s e

6s para os serviços FaaS, evidenciando um bom funcionando e que a integração entre eles é robusta.

A Figura 11 mostra o consumo da CPU pela Cloud Run do front-end durante as 12 horas de requisições, podendo notar alguns vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. O uso médio da CPU foi baixo, cerca de 7%, destacando a eficiência desta configuração.

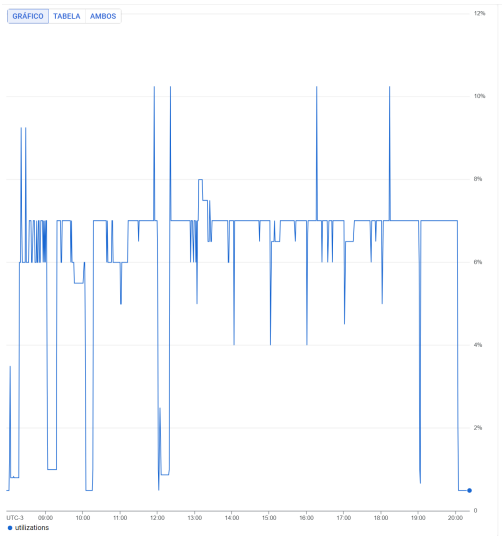

Figura 11: Consumo de CPU em aplicação com front-end em Cloud Run

A Figura 12 apresenta o número de requisições que os dois FaaS recebem conjuntamente durante as 12 horas de teste vindas do front-end em Cloud Run, sendo válido salientar que existem vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. É aparente uma regularidade de 86 chamadas por segundo durante todo o período de teste, impactando no preço final para o uso do serviço de FaaS.

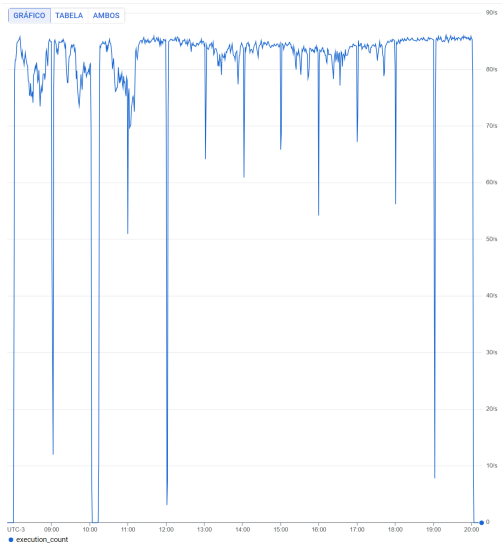

Figura 12: Quantidade de requisições que o FaaS recebe por segundo do front-end em Cloud Run

A Figura 13 exibe o tempo médio de execução do FaaS para responder às requisições feitas pelo front-end em Cloud Run durante as 12 horas do período de teste. Há uma constância em 5ms para tempo de resposta, sabendo que o tempo de execução do FaaS também impacta no preço final.

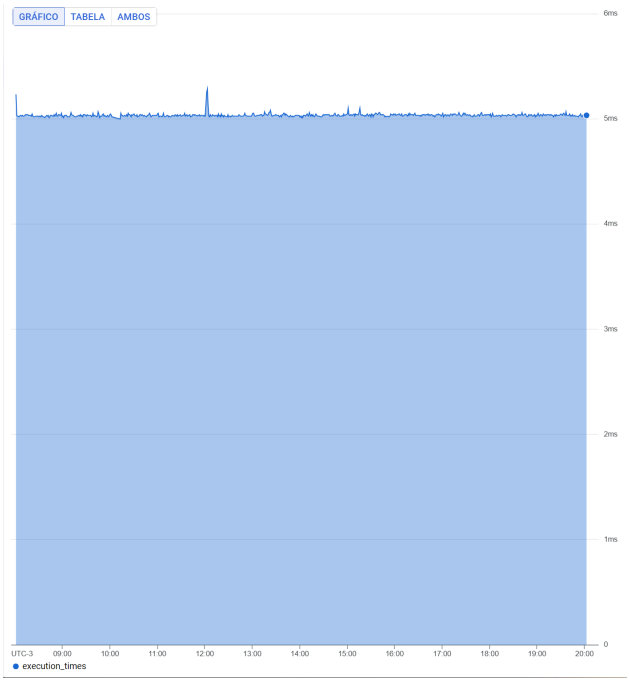

Figura 13: Tempo de execução do FaaS para responder às requisições feitas pelo front-end em Cloud Run

A Figura 14 mostra o faturamento detalhado para o dia de realização dos envios das requisições para a aplicação com front-end em Cloud Run e dois FaaS separados. Os custos foram divididos entre Cloud Run, R$5,47, e Cloud Functions, R$19,06, totalizando R$24,53, subtraídos os valores médios gastos pelos demais serviços estarem apenas hospedados e operantes. Tal resultado reflete um custo mais elevado, porém mais escalável e confiável que os anteriores.

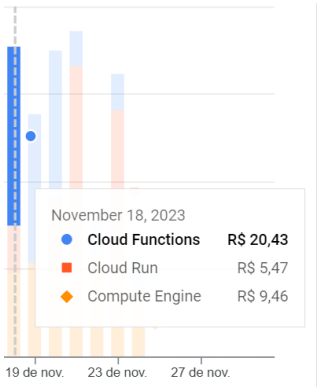

Figura 14: Faturamento para o dia que foram utilizados os serviços da aplicação com front-end em Cloud Run e FaaS separados

3.4 Front-end em VM com FaaS Unificado

A Figura 15 é um gráfico de amostra das requisições feitas pelo Postman aos serviços de front-end em VM e FaaS unificado. O tempo médio de resposta, representado pela linha azul, se mantém constante em torno de 394ms, sugerindo um modelo de aplicação robusto, também indicado pela taxa de erro de 0%.

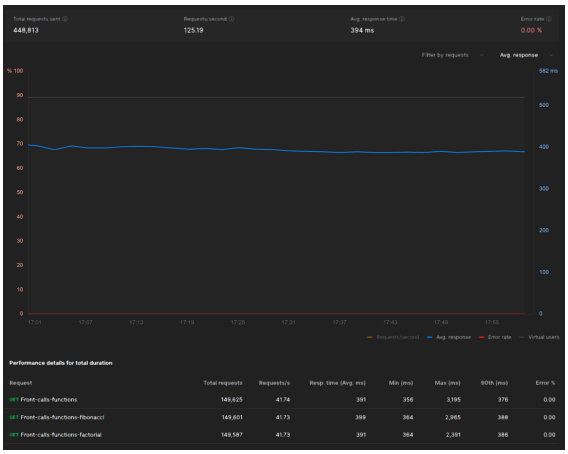

Figura 15: Requisições enviadas para aplicação com front-end em VM e FaaS unificada

Ainda na Figura 15, as tabelas na parte inferior fornecem uma visão detalhada do desempenho das requisições, chamando atenção o tempo máximo de resposta, sendo 3,195s para o front-end, 2,965s para a função Fibonacci e 2,391s para a função Fatorial. Resultados totalmente aceitáveis, sem atrapalhar na experiência do usuário.

A Figura 16 mostra o consumo da CPU da VM durante as 12 horas de requisições, podendo notar alguns picos e alguns vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. O uso médio foi de 23%, mostrando uma baixa demanda de processamento para essa abordagem de front-end isolado em VM, sendo uma boa alternativa para melhor distribuição de carga.

Figura 16: Consumo de CPU em aplicação com front-end em VM

A Figura 17 apresenta o número de requisições que o FaaS recebe durante as 12 horas de teste vindas do front-end em VM, sendo válido salientar que existem vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. É aparente uma regularidade de 87 chamadas por segundo durante todo o período de teste, impactando no preço final para o uso do serviço de FaaS.

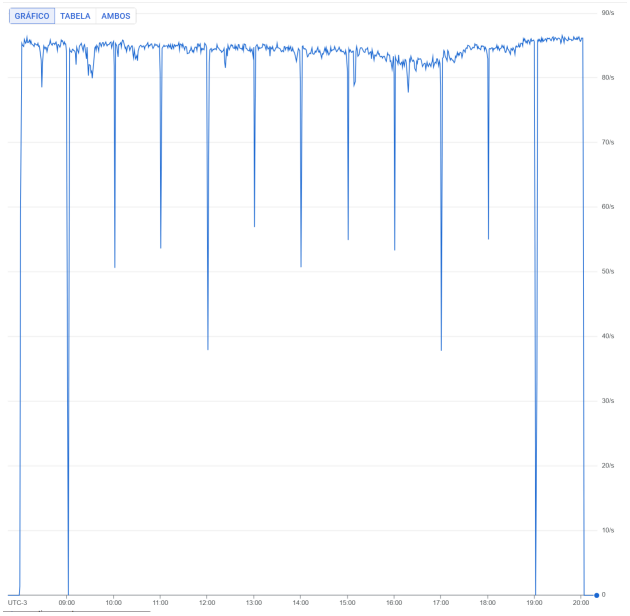

Figura 17: Quantidade de requisições que o FaaS recebe por segundo do front-end em VM

A Figura 18 mostra o tempo médio de execução do FaaS para responder às requisições feitas pelo front-end em VM durante as 12 horas do período de teste. Há uma constância em 5ms para tempo de resposta, sabendo que o tempo de execução do FaaS também impacta no preço final.

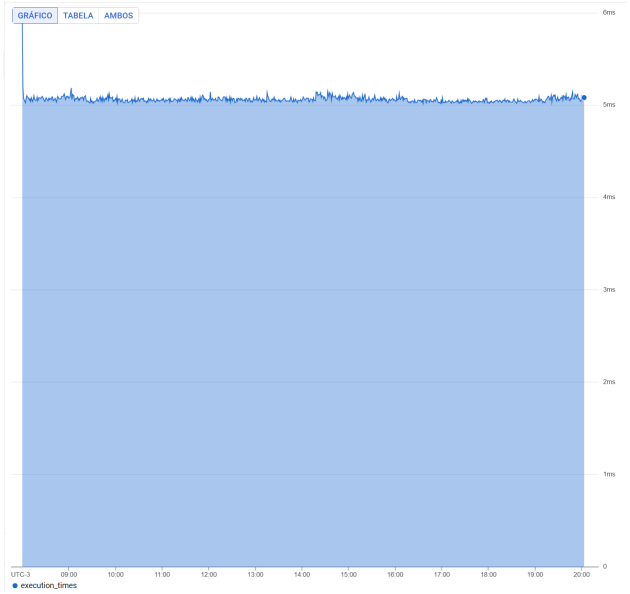

Figura 18: Tempo de execução do FaaS para responder às requisições feitas pelo front-end em VM

A Figura 19 mostra o faturamento detalhado para o dia de realização dos envios das requisições para a aplicação com front-end em VM e back-end em FaaS unificado. Os custos foram de Compute Engine e Cloud Functions, totalizando R$19,15, subtraídos os valores médios gastos pelos demais serviços estarem apenas hospedados e operantes.

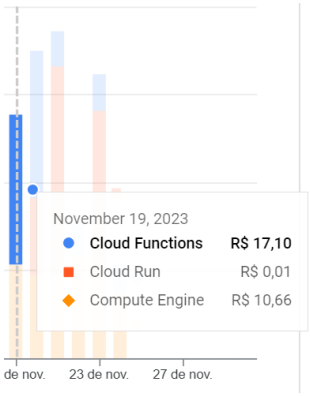

Figura 19: Faturamento para o dia que foram utilizados os serviços da aplicação com front-end em VM e FaaS unificado

3.5 Front-end em Cloud Run com FaaS Unificado

A Figura 20 é um gráfico de amostra das requisições feitas pelo Postman aos serviços de front-end em Cloud Run e FaaS unificado. O tempo médio de resposta, representado pela linha azul, apesar de alguns picos de variação, se mantém constante em torno de 407ms, sugerindo um modelo de aplicação robusto, também indicado pela taxa de erro de 0%.

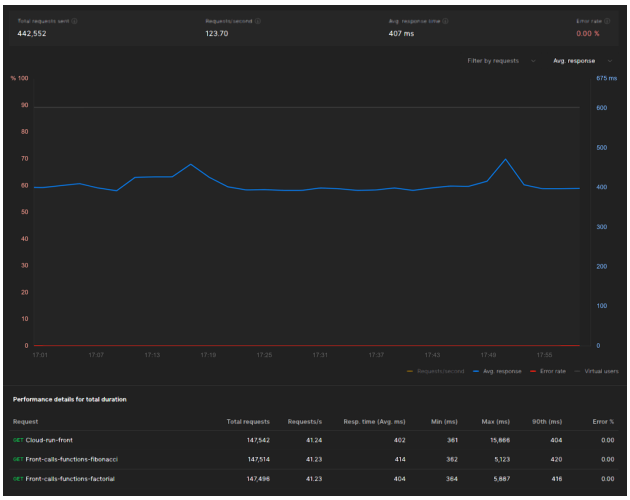

Figura 20: Requisições enviadas para aplicação com front-end em Cloud Run e FaaS unificado

Ainda na Figura 20, as tabelas na parte inferior fornecem uma visão detalhada do desempenho das requisições, tendo um destaque para tempos máximos de resposta altos (15s para front-end e 5s para funções), o que atrapalha na experiência do usuário, apesar de ser uma exceção, pois o percentil 90 dos tempos de resposta está em torno de 410ms.

A Figura 21 mostra o consumo da CPU da Cloud Run do front-end durante as 12 horas de requisições, podendo notar alguns picos e alguns vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. O uso médio foi de 1%, mostrando uma baixíssima demanda de processamento para essa abordagem de front-end isolado em Cloud Run, sendo uma ótima alternativa para melhor distribuição de carga, performance e confiabilidade do sistema.

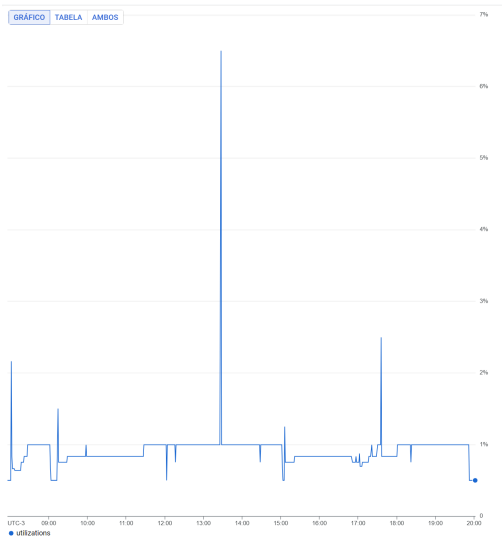

Figura 21: Consumo de CPU em aplicação com front-end em Cloud Run

A Figura 22 apresenta o número de requisições que o FaaS recebe durante as 12 horas de teste vindas do front-end em Cloud Run, sendo válido salientar que existem vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. É aparente uma regularidade de 86 chamadas por segundo durante todo o período de teste, impactando no preço final para o uso do serviço de FaaS.

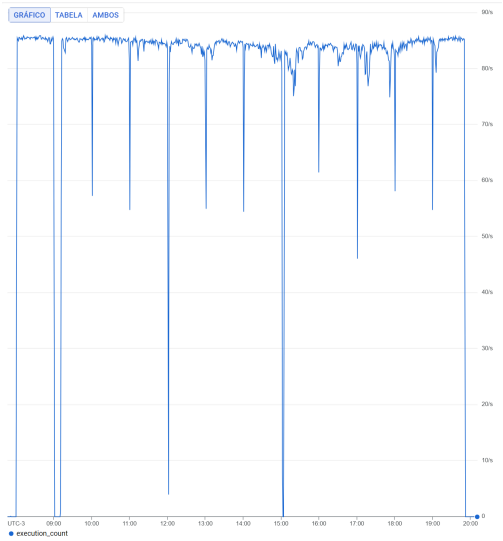

Figura 22: Quantidade de requisições que o FaaS recebe por segundo do front-end em Cloud Run

A Figura 23 exibe o tempo médio de execução do FaaS para responder às requisições feitas pelo front-end em Cloud Run durante as 12 horas do período de teste. Há uma constância em 5ms para tempo de resposta, sabendo que o tempo de execução das FaaS também impacta no preço final.

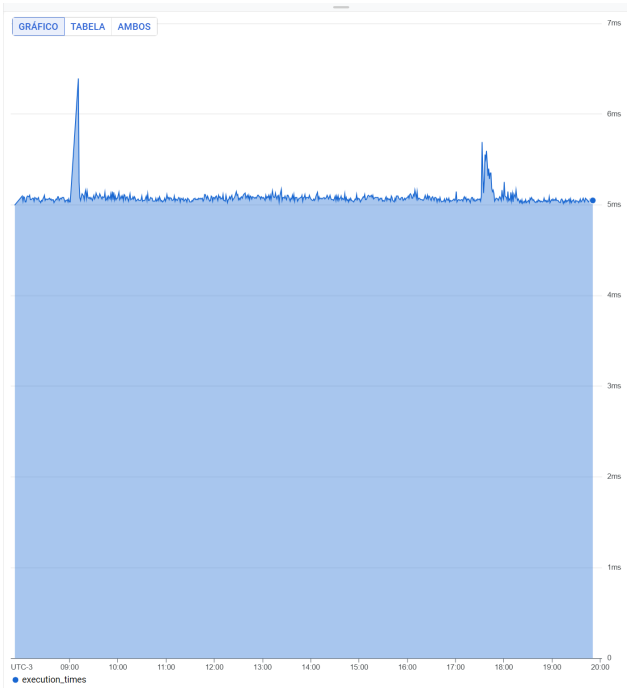

Figura 23: Tempo de execução do FaaS para responder às requisições feitas pelo front-end em Cloud Run

A Figura 24 mostra o faturamento detalhado para o dia de realização dos envios das requisições para a aplicação com front-end em Cloud Run e back-end em FaaS unificado. Os custos foram de Cloud Run e Cloud Functions, totalizando R$22,71, desconsiderando os valores de Compute Engine e subtraído os valores médios gastos pelos demais serviços estarem apenas hospedados e operantes.

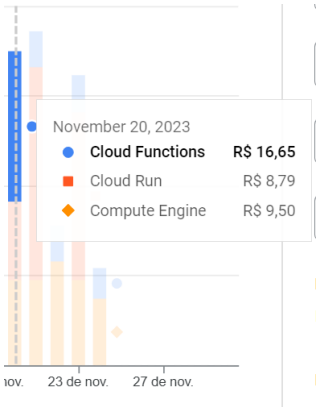

Figura 24: Faturamento para o dia que foram utilizados os serviços da aplicação com front-end em Cloud Run e FaaS unificado

3.6 Front-end e Back-end Separados em Cloud Run

A Figura 25 é um gráfico de amostra das requisições por segundos feitas pelo Postman aos serviços de front-end e back-end separados em Cloud Run. O tempo médio de resposta, representado pela linha azul, está muito estável em torno de 380ms, o melhor tempo de resposta entre todos modelos. Além disso, um taxa de erro de 0%, mostrando um modelo muito robusto para receber chamadas em alta demanda.

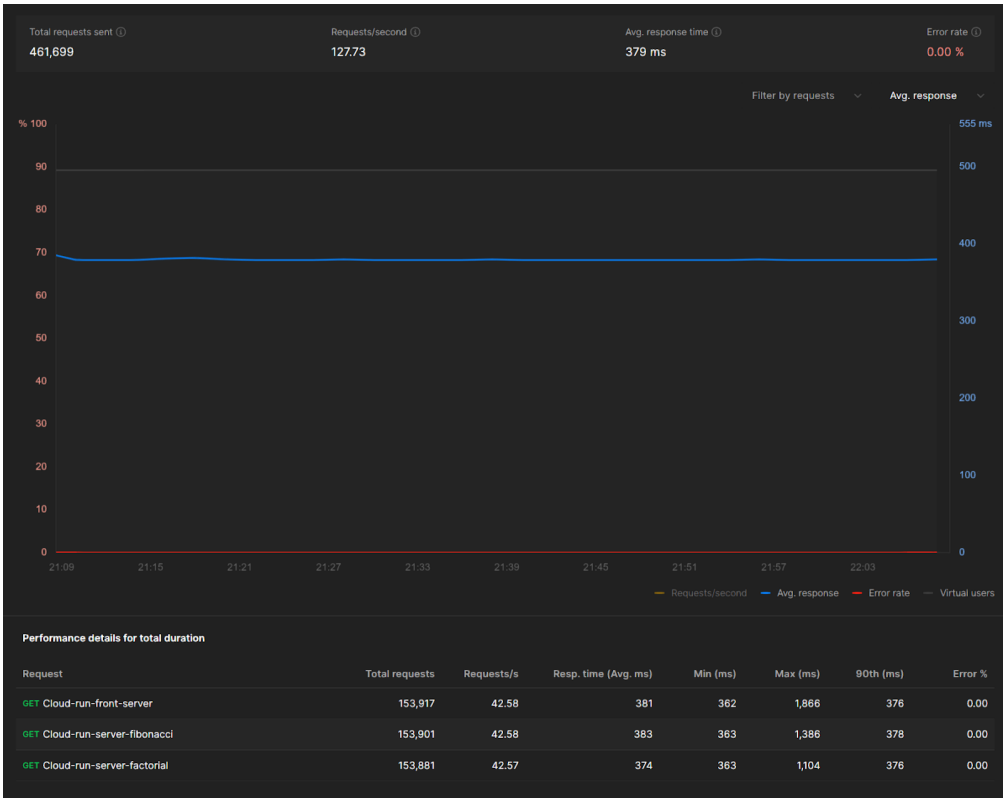

Figura 25: Requisições enviadas para aplicação com front-end e back-end em Cloud Run separados

Ainda na Figura 25, as tabelas na parte inferior fornecem uma visão mais detalhada do desempenho das requisições, chamando atenção para o tempo máximo das requisições para 1,8s para o front-end e 1,1s e 1,3s para as funções, mostrando-se um sistema muito confiável e robusto.

A Figura 26 mostra o consumo da CPU da Cloud Run somados front-end e back-end durante as 12 horas de requisições, podendo notar alguns picos e alguns vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. O uso médio foi de 4,8%, mostrando uma baixíssima demanda de processamento para essa abordagem de front-end e back-end separados em Cloud Run, sendo uma excelente alternativa para melhor distribuição de carga, performance e confiabilidade do sistema.

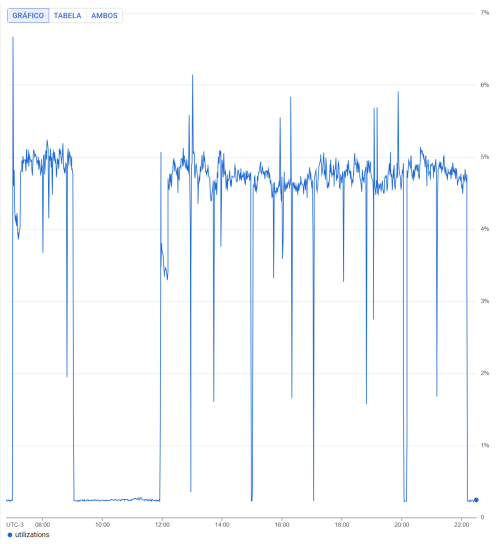

Figura 26: Consumo de CPU em aplicação com front-end e back-end em Cloud Run separadas

A Figura 27 mostra o faturamento detalhado para o dia de realização dos envios das requisições para a aplicação com front-end e back-end separados em Cloud Run diferentes. Os custos foram unicamente de Cloud Run, R$23,64, desconsiderando os valores de Cloud Functions e Compute Engine gastos pelos demais serviços estarem apenas hospedados e operantes.

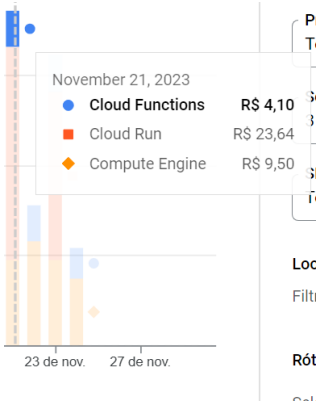

Figura 27: Faturamento para o dia que foram utilizados os serviços da aplicação com front-end e back-end em Cloud Run separados

3.7 Front-end e Back-end em VMs Separadas

A Figura 28 é um gráfico de amostra das requisições feitas pelo Postman aos serviços de front-end e back-end em VMs separadas. O tempo médio de resposta, representado pela linha azul, se mantém constante em torno de 380ms para o front-end e 460ms para as funções. Com uma taxa de erro, apesar de pequena, 0,15%, porém, existente.

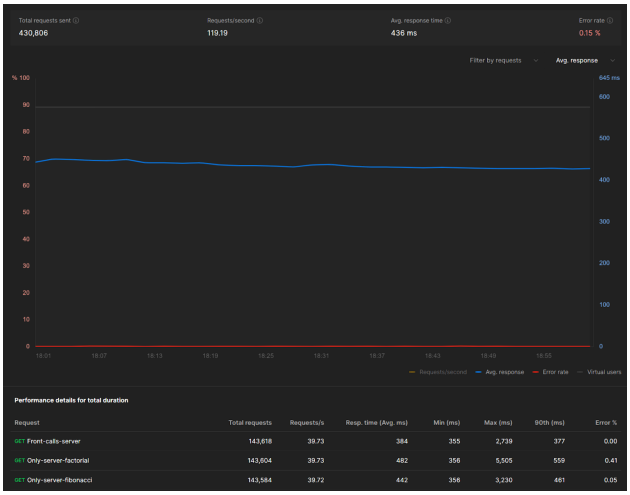

Figura 28: Requisições enviadas para aplicação com front-end e back-end em VMs separadas

Ainda na Figura 28, as tabelas na parte inferior fornecem uma visão mais detalhada do desempenho das requisições, tendo um tempo máximo mais problemático, que acontece em algumas exceções, sendo 2,7s para o front-end, 5,5s para Fatorial e 3,2s para Fibonacci. Mostrando que VMs com baixa capacidade não atendem muito bem a uma alta demanda de requisições.

A Figura 29 mostra o consumo da CPU das VMs de front-end (em rosa) e back-end (em roxo) durante as 12 horas de requisições, podendo notar alguns picos e alguns vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. O uso médio foi de 23% e 70%, respectivamente, mostrando uma alta demanda de processamento para a abordagem de back-end em VM.

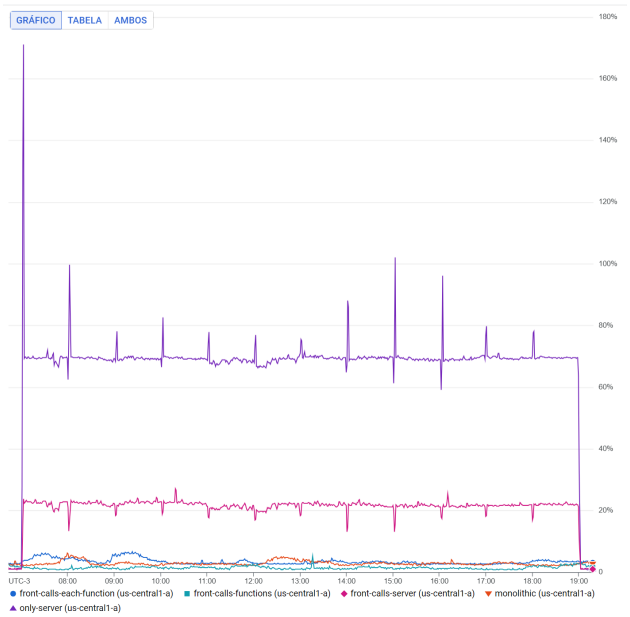

Figura 29: Consumo de CPU em aplicação com front-end e back-end em VMs separadas

A Figura 30 mostra o faturamento detalhado para o dia de realização dos envios das requisições para a aplicação com front-end e back-end separados em VMs diferentes. Os custos foram unicamente de Compute Engine, R$5,64, desconsiderando os valores de Cloud Functions e subtraídos os valores médios gastos pelos demais serviços estarem apenas hospedados e operantes. Tal resultado reflete o uso mais ameno das VMs financeiramente falando.

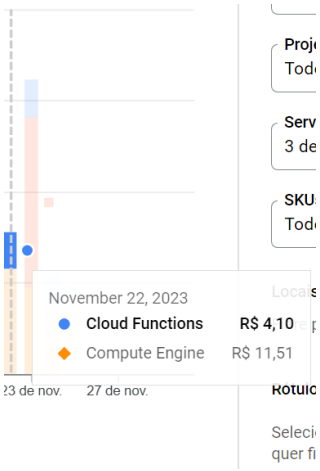

Figura 30: Faturamento para o dia que foram utilizados os serviços da aplicação com front-end e back-end em VMs separadas

3.8 Monolítico em Cloud Run

A Figura 31 é um gráfico de amostra das requisições feitas pelo Postman à aplicação monolítica em Cloud Run. O tempo médio de resposta, representado pela linha azul, se mantém constante em torno de 400ms para todos serviços. Com uma taxa de erro nula.

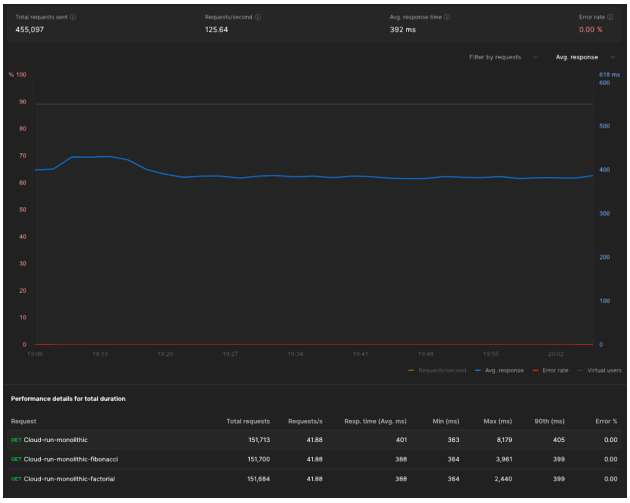

Figura 31: Requisições enviadas para aplicação monolítica em Cloud Run

Ainda na Figura 31, as tabelas na parte inferior fornecem uma visão mais detalhada do desempenho das requisições, tendo um tempo máximo de 8s para o front-end, algo problemático, 3,9s

para a função Fibonacci e 2,4s para Fatorial, mas sendo esses exceções.

A Figura 32 mostra o consumo da CPU da Cloud Run para aplicação monolítica durante as 12 horas de requisições, podendo notar alguns picos e alguns vales que são justificados pelo tempo de transição entre o término de um ciclo e o início do próximo. O uso médio foi de 7%, mostrando uma baixíssima demanda de processamento para essa abordagem de front-end e back-end em aplicação monolítica em Cloud Run, sendo uma excelente alternativa de performance e confiabilidade do sistema, mesmo sendo um modelo bem centralizado.

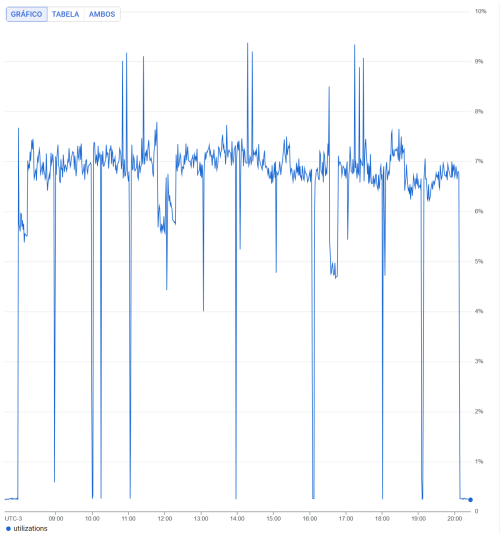

Figura 32: Consumo de CPU em aplicação monolítica em Cloud Run

A Figura 33 mostra faturamento detalhado para o dia de realização dos envios das requisições para a aplicação monolítica em Cloud Run. Os custos foram unicamente de Cloud Run, R$18,65, desconsiderando os valores de Cloud Functions e Compute Engine gastos pelos demais serviços estarem apenas hospedados e operantes.

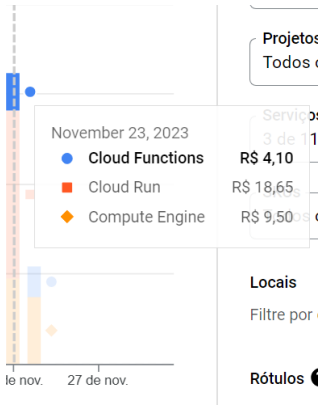

Figura 33: Faturamento para o dia que foram utilizados os serviços da aplicação monolítica em Cloud Run

3.9 Discussão

3.9.1 Análise Detalhada de Desempenho e Custo

Nossa investigação das diferentes configurações de instanciação no Google Cloud Platform ofereceu uma visão ampla do comportamento e eficiência de diversas arquiteturas na nuvem. Observamos que as configurações que segregavam o front-end e as funções computacionais (como FaaS) tendiam a apresentar um desempenho superior, refletido em tempos de resposta mais baixos e um número maior de requisições processadas por segundo. Essa melhora no desempenho, contudo, estava atrelada a um aumento proporcional nos custos, evidenciando a relação custo-benefício inerente às soluções de nuvem. O aumento dos custos pode ser atribuído à maior alocação de recursos e à necessidade de manter infraestruturas separadas operacionalmente. Esse fenômeno ressalta a importância de um planejamento cuidadoso em termos de alocação de recursos e estratégias de otimização de custos, especialmente para organizações com orçamentos restritos ou em ambientes de produção onde a eficiência de custos é uma prioridade.

3.9.2 O Papel do Cloud Run na Eficiência Operacional

As configurações que empregaram o Cloud Run se destacaram por sua eficiência notável em termos de utilização da CPU, com a maioria apresentando um uso médio da CPU bem abaixo de 10%. Este achado sublinha o Cloud Run como uma solução altamente eficaz para aplicações que demandam eficiência de recursos, particularmente quando combinado com Function as a Service (FaaS) para tarefas computacionais específicas. Além de sua eficiência, o Cloud Run demonstrou uma capacidade impressionante de escalabilidade, ajustando-se automaticamente às variações na demanda de requisições. Esta característica o torna particularmente atraente para aplicações que experimentam flutuações imprevisíveis na carga de trabalho, oferecendo uma maneira de otimizar os custos operacionais sem sacrificar o desempenho. Contudo, é importante notar que, embora o Cloud Run apresente um uso eficiente de recursos, o custo total das operações pode aumentar com a escalabilidade, especialmente em cenários de alta demanda.

3.9.3 Benefícios e Desafios da Arquitetura Distribuída

Comparativamente, as configurações que adotaram uma arquitetura distribuída, separando o front-end do back-end, mostraram uma melhoria substancial no desempenho, com tempos de resposta reduzidos e um número maior de requisições por segundo. Essa melhoria pode ser atribuída à eficiente distribuição da carga e ao uso otimizado dos recursos computacionais. Ao separar as funções computacionais em instâncias distintas, conseguimos mitigar os gargalos de desempenho que são típicos em ambientes monolíticos. No entanto, essa abordagem também introduz complexidades adicionais em termos de gerenciamento e integração de sistemas, além de potencialmente aumentar a superfície de ataque para questões de segurança. Além disso, as configurações distribuídas exigem uma consideração cuidadosa do tráfego de rede entre os componentes, o que pode influenciar o desempenho geral e a latência do sistema.

3.9.4 Considerações de Custo nas Diferentes Configurações

Embora a separação de front-end e back-end e a implementação de FaaS tenham melhorado o desempenho, essas abordagens resultaram em um aumento nos custos operacionais. Configurações que combinaram Cloud Run e FaaS, em particular, apresentaram custos mais elevados devido à maior demanda de recursos e à natureza escalável desses serviços. Isso destaca a importância de um equilíbrio entre o desempenho desejado e os custos operacionais. Para aplicações que exigem alta

disponibilidade e rápida resposta, como sistemas críticos ou plataformas de alto tráfego, investir em arquiteturas distribuídas e serviços escaláveis pode ser justificável. Em contraste, para projetos com limitações de orçamento ou requisitos menos exigentes de desempenho, uma abordagem mais centralizada ou uma separação menos intensiva de serviços pode ser mais adequada. Esta decisão deve ser baseada em uma avaliação criteriosa das necessidades específicas da aplicação e dos recursos disponíveis.

4 Conclusão

Este estudo proporcionou uma compreensão aprofundada das complexidades e nuances da instanciação de serviços na nuvem no Google Cloud Platform. A análise minuciosa de várias configurações de instância, usando benchmarks de cálculos de fatorial e Fibonacci, revelou insights cruciais sobre o equilíbrio entre desempenho e custo em ambientes de nuvem. Constatou-se que, embora configurações distribuídas ofereçam melhor desempenho, elas implicam em custos mais elevados.

Em contrapartida, o Cloud Run se destacou por sua notável eficiência e escalabilidade, servindo como uma solução ideal para cargas de trabalho variáveis. Este trabalho sublinha a importância de uma análise detalhada das necessidades de aplicativos específicos ao escolher a arquitetura de nuvem, enfatizando que a decisão deve levar em conta tanto as demandas de desempenho quanto as limitações orçamentárias. As descobertas aqui apresentadas são fundamentais para orientar futuras pesquisas e práticas no campo da otimização de recursos em nuvem, destacando a necessidade de uma abordagem equilibrada e incisiva na escolha das configurações de nuvem.

Em estudos futuros, pode ser explorado o equilíbrio entre custo e eficiência em uma gama mais ampla de aplicações e cenários de carga de trabalho. Seria particularmente valioso investigar como diferentes configurações de nuvem se comportam sob cargas de trabalho dinâmicas e em cenários de uso variados. Além disso, a otimização de custos em arquiteturas de nuvem híbrida e multi-nuvem, áreas que estão se tornando cada vez mais relevantes no panorama da computação em nuvem. Essas investigações ajudarão nas tomadas de decisão mais assertivas sobre a configuração de infraestrutura de nuvem, equilibrando eficácia operacional e eficiência de custos de acordo com suas necessidades específicas.

Referências

[1] Google Cloud Platform, `https://cloud.google.com/docs?hl=pt-br`

[2] O que é o Cloud Run,
`https://cloud.google.com/run/docs/overview/what-is-cloud-run?hl=pt-br`

[3] O que é uma máquina virtual?,
`https://cloud.google.com/learn/what-is-a-virtual-machine?hl=pt-br`

[4] O que é função como serviço?, `https://www.cloudflare.com/pt-br/learning/serverless/glossary/function-as-a-service-faas`

[5] What is Postman?, `https://www.postman.com/product/what-is-postman`

[6] Compute Engine, `https://cloud.google.com/compute?hl=pt-BR`

[7] Cloud Functions, https://cloud.google.com/functions?hl=pt-BR