

Otimização do desempenho de uma aplicação de pagamentos pela melhoria de sua infraestrutura

Optimization of the performance of an online payment application by the improvement of its infrastructure

Fernando Rocha Moreira, Rafael Rabelo Nunes, William Ferreira Giozza, Georges Amvame Nze
Universidade de Brasília, Departamento de Energia Elétrica
Campus Universitário Darcy Ribeiro, Brasília, Brazil
frochamoreira@gmail.com, rafaelrabelo@unb.br, giozza@unb.br, georges@unb.br

Resumo — Esse trabalho tem como objetivo relatar uma experiência prática de um banco brasileiro de grande porte, que utilizava uma aplicação de alta criticidade de pagamentos. Com o aumento das transações, a aplicação começou a apresentar diversas limitações, gerando indisponibilidades. Como não era possível alterar o código da aplicação para proceder com otimizações visto que a aplicação tinha o código fechado, foi-se aplicado o método MIASP para identificar e resolver as limitações por meio de ajustes na infraestrutura. Os problemas foram contornados por meio da utilização de conceitos de sistemas distribuídos como arquitetura *middleware*; processamento em concorrência; banco de dados; e rotinas programadas em sistemas operacionais.

Palavras Chave – desempenho; infraestrutura; pagamentos; *middleware*; software

Abstract — This work aims to report a practical experience of a big Brazilian bank, which used a highly critical payment application. With the raise in transactions, the application started showing some limitations, generating unavailabilities. As it was not possible to change the application code to proceed with optimizations since the application had a closed code, it was applied the MIASP method to identify and solve the limitations by adjustments in the infrastructure. The problems were overcome by concepts of systems distributed like *middleware architecture*; concurrent processing; database and programmed routines in operational systems.

Keywords - performance; infrastructure; payments; *middleware*; Software

I. INTRODUÇÃO

Com o avanço da tecnologia e da competitividade no ramo financeiro, as instituições desse setor buscam cada vez mais espaço através da Internet, para fornecerem melhores produtos, aplicativos de celulares mais robustos, tornando a competitividade cada vez mais acirrada, onde milhões de dólares são transacionados a cada momento por canais digitais [1][2].

Baseado neste fundamento, o desempenho de um aplicativo é um fator fundamental para o sucesso da experiência do usuário[3]. O conceito de sistemas distribuídos, e uma arquitetura robusta em *middleware* [4][5] auxilia para melhorar essa experiência, provendo ao aplicativo uma capacidade de

processamento em concorrência que otimiza o trabalho ganhando velocidade e desempenho[6][7].

Este trabalho baseia-se em um estudo de caso com um aplicativo implementado por um *software core* de código fechado para pagamentos em tempo real, em uma instituição financeira de grande porte que possui mais de 90.000 funcionários, e que está presente no Brasil e no Exterior. A principal vantagem desse aplicativo com relação a outros é a possibilidade de customização de cada tipo de operação financeira, além de operar nos diferentes tipos de processos de pagamentos no mundo, a saber: CHIPS (*Clearing House Interbank Payment System*), FED (*Federal Reserve System*), SWIFT (*Society for Worldwide Interbank Financial Telecommunication*).

II. ESTUDO DE CASO

A infraestrutura do aplicativo consistia em um servidor de aplicações Weblogic, com apenas uma máquina com duas instâncias, responsável por efetuar a conexão com um banco de dados Oracle, disponível em outro servidor [4][5]. Desde quando foi implantado, no final de 2015, o aplicativo vinha conseguindo atender a demanda. Contudo, com a expansão das agências e a inserção da instituição nos diferentes meios de operações financeiras existentes nos Estados Unidos, o aplicativo passou a apresentar limitações gerando indisponibilidades e transtornos para os clientes. Um estudo feito pela própria instituição financeira constatou que para cada hora de indisponibilidade, ela deixa movimentar em média US\$ 153.307.076,88. Por questões contratuais a instituição financeira não pode efetuar manutenções no código do aplicativo, e dessa forma, foi necessário encontrar uma outra forma de endereçar as suas limitações.

III. METODOLOGIA

A metodologia empregada neste trabalho usa um modelo [8] baseado no MIASP – Método de Identificação Análise e Solução de Problemas dividido em 5 fases [9][10][11]: Fase de “Identificação e Observação do problema”, onde foi realizado um *brainstorming* com todos os especialistas, de forma a levantar quais os principais incidentes com o aplicativo; Fase de “Análise do problema”, onde foram feitas as análises e

investigações dos problemas causados pelo aplicativo. Estas análises se basearam em evidências como: coleta e análise de *logs*, registros de incidentes, e desempenho com o aplicativo; Fase de “Planejamento da ação”, onde, de acordo com a análise feita na fase anterior, foi elaborado juntamente com os especialistas um plano de ação; Fase de “Ação”, onde foram realizadas as implementações das melhorias do aplicativo para mitigar os incidentes de acordo com o plano de ação. Foram realizados testes nos ambientes de homologação e desenvolvimento. Em seguida foi feita uma produção assistida das implementações, ou seja, elas entraram em produção em diverentes datas para que não houvesse grande impacto no ambiente em caso de insucesso; e por fim, a fase de “Conclusão” onde foram feitas análises dos resultados obtidos.

IV. RESULTADOS OBTIDOS

A Tabela 1 consolida os principais incidentes com a suas causas-raiz.

TABELA I. INCIDENTES E SUAS CAUSAS RAIZ

<i>Incidente</i>	<i>Causa raiz</i>
1. Estouro de conexões no <i>datasource</i>	Baixa capacidade de conexões configurados no <i>datasource</i> do <i>Weblogic</i> - A ferramenta não fecha as conexões com o banco de dados
2. Intervenções humanas feitas de forma errônea	Alta complexidade para levantar a ferramenta quando há indisponibilidade
3. Alto nível de indisponibilidade na ferramenta	Somente uma máquina operando, processando e disponibilizando o serviço
4. Estouro no tamanho de armazenamento de <i>logs</i>	Má configuração na <i>crontab</i> do sistema operacional que não está comprimindo os <i>logs</i> .
5. Problemas de processamento em concorrência	Má implementação da infraestrutura nos primórdios do projeto
6. Pagamentos que desaparecem e ficam travados e são reportadas pelos usuários posteriormente	Baixa monitoração do movimento de pagamentos com as diversas entidades de pagamentos no exterior.

Como resposta ao incidente 1, foi aumentado o número de conexões do *datasource* no *weblogic* [4]. Também foram implementados *restarts* automáticos duas vezes na semana no período noturno para zerar o *datasource* sem afetar o usuário final [4][12]. Para os incidentes 2, 3 e 5 foi necessário um estudo juntamente com o fornecedor, onde os parâmetros de configuração da aplicação foram alterados no *backend* permitindo a implantação da máquina 2. Isto possibilitou que elas pudessem trabalhar concorrentemente em *cluster* [6][7], com as 4 instâncias. Também foi configurado para que a aplicação fosse manipulada via interface do *Weblogic* para facilitar intervenções manuais [4][5]. Para o incidente 4, foi criada uma rotina diária na *crontab* do *Linux*, que comprime os *logs* diários, faz o *backup* e apaga os *logs* antigos. Também houve um *upgrade* de disco nas duas máquinas. Para o incidente 6 foi realizada uma melhoria do monitoramento por meio de painéis que mostram o volume de dados transitando em tempo real e de softwares que monitoram os *logs* da aplicação também em tempo real.

Após todas as melhorias realizadas na infraestrutura, a aplicação realizou 15.000 pagamentos de diferente fontes em um só dia sem nenhum tipo de transtorno ou indisponibilidade para o usuário final sem prejuízos para a instituição financeira.

Também foram realizados testes de *compliance* com o CHIPS. Este simulou uma remessa de pagamentos para a ferramenta e ela se comportou muito bem. Com isso a instituição financeira obteve certificação e autorização para continuar operando conforme as normas solicitadas pelo mesmo.

V. CONCLUSÕES

Através do uso da otimização da infraestrutura aplicando conceitos de sistemas distribuídos como concorrência e uma arquitetura *middleware* fez com que fosse possível corrigir as limitações da aplicação e a instituição financeira não tivesse mais incidentes de grandes proporções. Soluções automatizadas fazem com que diminua a ocorrência de intervenções humanas mitigando assim riscos de aplicação de procedimentos errôneos. Uma monitoração bem eficiente e eficaz faz com que seja possível capturar incidentes em tempo real. O aplicativo aumentou a disponibilidade com o trabalho em concorrência conseguindo processar uma grande remessa de pagamentos diariamente e também em datas específicas.

AGRADECIMENTOS

CAPES (grants 23038.007604/2014-69 FORTE and 88887.144009/2017-00 PROBRAL), CNPq (grants 312180/2019-5 PQ-2, BRICS2017-591 LargEWiN, e 465741/2014-2 INCT in Cybersecurity) e FAP-DF (grants 0193.001366/2016 UIoT and 0193.001365/2016 SSDDC), assim como os projetos de cooperação com o Conselho Administrativo de Defesa Econômica (grant CADE 08700.000047/2019-14).

REFERÊNCIAS BIBLIOGRÁFICA

- [1] C. E. C. Fonseca.; F. S. Meirelles; E. H. Diniz.; “Tecnologia bancária no Brasil”. São Paulo: FGV, ERA, SÔS. 2010.
- [2] Federação Brasileira de Bancos “Pesquisa FEBRABAN de Tecnologia Bancária 2019” 2019, unpublished
- [3] B. Albert , T. Tullis, “Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics”. Elsevier, Waltham, 2013
- [4] S. R. Alpati “Oracle WebLogic Server 11g Administration Handbook” 1st. McGraw-Hill Osborne Media ,2011
- [5] Y. Wen, H. Jingsha “The Application of Tuxedo Middleware in the Banking System” 2011 in press
- [6] G. Coulouris, J. Dollimore, T. Kindberg, G. Blair “Sistemas Distribuídos - Conceitos e Projeto” ,5ª edição, 2013
- [7] A. S. Tanenbaum - Sistemas Distribuídos: Princípios e Paradigmas 2013
- [8] R. Descartes. Discurso do Método. Lisboa. Textos Filosóficos. Edições 70. 1979. 118 p.
- [9] D. Filho, O. “As sete ferramentas do planejamento da qualidade”, Belo Horizonte: Fundação Cristiano Ottoni, 1996, 183 p
- [10] L. C. S. Duarte, “Abordagem Comparativa Entre Métodos De Solução De Problemas: O Miasp E O Método Das 6 Etapas” in press
- [11] S. M Azzam; C. Cipolla “Utilização Das Ferramentas De Design Thinking Nas Atividades De Redesenho De Processos” unpublished
- [12] Identifying bank frauds using CRISP-DM and decision trees. BC da Rocha, RT de Sousa Junior. International Journal of Computer Science and Information Technology 2 (5), 2010.