REVISTA DE EMPREENDEDORISMO E GESTÃO
DE MICRO E PEQUENAS EMPRESAS

EDITORA ENTERPRISING

O uso de *capping* em mainframe dedicados em zCloud: redução de custos em ambiente IBM z/OS

Matheus Porsch, Faculdade Horizonte, Brasil

Ana Cristina Brandão Ribeiro Silva, Faculdade Horizonte, Brasil

RESUMO

A plataforma de computação em nuvem baseada na tecnologia mainframe da IBM, projetada para fornecer serviços de computação em nuvem de nível empresarial para empresas que exigem altos níveis de segurança, disponibilidade e desempenho, alavancou a necessidade de desenvolvimento de um processo que conseguisse compartilhar uma única máquina com diversas empresas garantindo os recursos acordados em contrato. Assim, considerando que a maioria das empresas utilizam a nuvem, tornando-se um assunto recorrente para aqueles que trabalham com a plataforma, é imprescindível um *capping* para delimitar os recursos da máquina para cada ambiente. Nesta perspectiva, realizou-se um estudo de caso com o objetivo de demostrar os impactos de três ambientes com *capping* ativos, tendo em vista que as empresas procuram soluções para reduzir seus custos com as faturas mensais.

.

Palavras-chave: Mainframe; Batch; Db2; Z/os.

1. INTRODUÇÃO

A nova plataforma oferecida pela IBM, o IBM zCloud, é apresentado para o contratante do serviço como um ambiente privativo e único, porém, rodado em uma máquina com recursos compartilhados. Atualmente, mais de três quartos das empresas usam a nuvem de alguma forma, e esse número cresce a cada dia, segundo a própria IBM.

Devido a sua alta escalabilidade, o serviço é ofertado pela capacidade. Uma das métricas é a capacidade de processamento, MIPS ou MSU, que o ambiente consegue entregar. Neste cenário a necessidade de um *capping* para delimitar os recursos da máquina para cada ambiente é essencial.

Porsch, M., Silva, A.C.B.R.; O uso de *capping* em mainframe dedicados em zCloud: redução de custos em ambiente IBM z/OS. Revista de Empreendedorismo e Gestão de Micro e Pequenas Empresas V.9, Nº1, p.82-100, Jan/Abr. 2024. Artigo recebido em 20/03/2024. Última versão recebida em 18/04/2024. Aprovado em 25/04/2024.

Assim, é importante *registrar que capping* é o processo de limitar ou controlar o uso de recursos de computação em um mainframe para garantir que os custos sejam mantidos sob controle. Há diversos tipos de *capping*, cada uma dessas técnicas tem seus próprios benefícios e limitações, e a escolha da técnica de nivelamento apropriada depende das necessidades e requisitos específicos da organização.

O *capping* entra como divisor e o limitador do recurso contratado. Mesmo o mainframe rodando outros clientes, o recurso estará disponível para o seu ambiente, garantindo assim que a parte pertencente ao contratante esteja sempre disponível e outra região não compita pelos recursos.

Sendo assim, hoje é possível ter um mainframe com uma capacidade de processamento altíssima, como por exemplo uma z16 modelo 3931-710 com 200 núcleos de processamento e entrega 215.089 MIPS de capacidade média (IBM Corp, 2023a), e dividir essa capacidade para vários clientes hospedados em uma mesma máquina. Podendo ser ofertado serviços de nuvem de forma mais acessível financeiramente, centralizada e competitiva.

Há uma tendência crescente de clientes migrando para plataformas em nuvem. É uma estratégia atual da IBM para se adaptar à necessidade de cliente de pequeno porte que não possuem recursos financeiros para a comprar uma máquina dedicada e que não querem renunciar aos benefícios da plataforma IBM Z.

Há casos em que o cliente deseja apenas limitar sua própria máquina dedicada para reduzir custos de *softwares* que são cobrados pelo consumo (MLC). Todavia, um *capping* bem definido, limitará os recursos do sistema e não fornecerá mais quando atingido, limitando o consumo, consequentemente o custo.

Sendo assim, o objetivo geral deste estudo de caso é demostrar os impactos de três ambientes com *capping* ativos, dois de uma instituição financeira, outro de uma seguradora, exibindo os resultados em gráficos para melhor compreensão dos impactos que o processo traz para o ambiente, pois, as empresas onde o estudo de caso se baseou, procuravam soluções para economizar na sua fatura mensal.

Além disso, buscou-se abordar e descrever alguns modelos mais usados de *capping* disponíveis atualmente no sistema Z, e mostrar suas vantagens e desvantagens na prática, bem como seus impactos na fatura. Já que é uma tendência crescente de mercado a dominação dessa técnica como diferencial na qualificação do profissional que atua nessa área.

2. ENQUADRAMENTO TEÓRICO

2.1. INVESTIGAÇÃO PRÉVIA

O referido trabalho aborda três casos em ambientes distintos, explicando a situação referente a cada cliente, sendo denominados 1(um), 2(dois), 3 (três), os quais são apresentados a seguir.

Um dos casos, cliente (1), o ambiente é um mainframe Z15 de classe *Enterprise* modelo 8561-735 com capacidade de 47.061 MIPS (IBM Corp, 2023), hospedado em um ambiente zCloud, onde o cliente, uma seguradora, está hospedado desde maio de 2021 com a capacidade contratada de 1.390 MIPS com *Initial capping*. O cliente optou pelo ambiente zCloud para redução de custo financeiro e evitar o *upgrade* de máquinas futuras.

Outro caso, cliente (2), o ambiente é um mainframe z15 de classe *Business* modelo 8562-R05 com capacidade total de 3.310 MIPS (IBM Corp, 2023), dedicado, onde o cliente, uma instituição financeira (conhecida popularmente como uma *Fintech*), possui sua própria máquina e optou pelo *capping* para redução de custos devido a *softwares* que são cobrados pelo consumo. Seu principal produto consumido é o CICS. Ha momentos chaves e estratégicos que o cliente faz um *upgrade* para o modelo 8562-T05 de capacidade total média de 4.140 MIPS.

Em relação ao cliente (3), o ambiente é um mainframe z14 de classe *Enterprise* modelo 3906-740 com capacidade de 46.300 MIPS (IBM Corp, 2023), hospedado em um ambiente zCloud, é uma instituição financeira, está hospedado com uma capacidade contratada de 6.918 MIPS. O cliente optou pelo serviço de *cloud* para evitar *upgrade* futuro de mainframe e pela alta escalabilidade.

Todos os clientes usam o sistema operacional IBM z/OS 2.04, com fonte de dados exportadas do SMF pelo *software* Tivoli Decision Support for z/OS (TDS) que armazena suas informações de forma estruturada em tabelas de um banco de dados DB2.

A pesquisa tem viés qualitativo com base em Flick (2009) que compreende tratar-se de uma escolha, relação profissional bem como a comunicação do pesquisador em campo com diversas abordagens e métodos. Desta forma, trata-se de um estudo de caso conforme já mencionado anteriormente, com base em Yin (2001), que indica esse tipo de pesquisa para situações ocorridas em ambiente real, ou seja, problemas vivenciados ou simulados em função das rotinas de trabalho em determinada instituição. Além disso, são utilizadas diversas etapas pré-determinadas, o que foi delineado neste trabalho.

Os métodos utilizados para a coleta e geração de gráficos e análise da pesquisa se deu por meio de etapas que serão detalhadas a seguir.

2.1.1 *Softwares*

Para o desenvolvimento dos gráficos foi usado o *software* da Microsoft, o Power BI. A conexão utilizada para fonte de dados foi o tipo IBM DB2 Database pelo drive IBM, com consultas em linguagem SQL direto do repositório hospedado em um mainframe. Também pode ser usado conexão via ODBC (Parziale et al., 2016).

2.1.2 Dados

O trabalho presente não focará no processo de extração dos dados direto do SMF, porém a fonte dos registros, mapas e campos do SMF são mencionados para o leitor saber a fonte dos dados, o propósito do trabalho visa o processo para obtenção dos resultados do estudo, citando de onde os dados vieram e como foram calculados. Os ambientes estudados possuem um *software* já dedicado exclusivamente para essa função de extração, além de um time destinado para o suporte do tal.

Para a demonstração prática dos resultados do *capping* em ação, foi usado dados estruturados do repositório DB2, extraídos do SMF dos sistemas pelo *software* Tivoli Decision Support for z/OS. A tabela usada para monitorar o consumo e os valores de *capping* foram as MVSPM_LPAR_H e a MVSPM_LPAR_MSU_T com granularidade de intervalo de tempo de uma hora, em casos em que havia a disponibilidade de intervalo de 15 minutos, a granularidade foi diminuída. Seus dados são atualizados diariamente, sendo assim, haverá sempre dados disponíveis do dia anterior corrente.

Em ambas as tabelas, os campos DATE e TIME, foram usadas para criar o eixo x das linhas de tempo dos gráficos. E o campo MVS_SYSTEM_ID para filtrar os dados de cada LPAR do sistema, representadas pelas cores nas colunas verticais.

Estas tabelas fornecem estatísticas sobre partições lógicas e atividade do processador em um ambiente PR/SM. Ele contém dados de registros SMF tipo 70, processados por meio de um procedimento de registro (DRL2S070) (IBM Corp., 2009).

A fórmula usada para calcular o consumo médio (em MIPS) oriundas da tabela é:

$$\left(\frac{\left(\frac{TotaldaMediadeMIPSdaBOX}{TotaldeProcessadoresGCP}\right)}{Intervaloemsegundos}\right) Total\ desegundostrabalhado$$

Para calcularmos com os dados armazenados no DB2 extraídos pelo TDS, usamos os seguintes campos na fórmula:

$$\left(\frac{MIPS}{MEASURED_{SEC}}\right) * CPU$$

O resultado trará os valores médios em MIPS do consumo em um intervalo horário em cada data para cada LPAR, para calcular o consumo total do *sysplex*, somamos os valores de todas LPARs. Assim conseguimos criar um gráfico em MIPS Usado vs Data Hora

Para analisar os valores do *capping*, dependera de qual tipo de *capping* está implementado na máquina.

***LPAR Capping* (*soft capping*)**

Capping do tipo *Soft* LPAR, podemos usar dois campos da tabela MVSPM_LPAR_H do TDS, o WLM_CAPPING_PCT e o WLM_CAPPING_ACT.

O campo WLM_CAPPING_PCT, mostra a porcentagem do tempo do intervalo estudado que máquina estava capada, por exemplo, no caso de intervalos de 1 hora de estudo, suponhamos que nos últimos 20 minutos a máquina foi capeada nesse intervalo, o valor exibido no campo será de 33, equivalente a 33% do tempo do intervalo, que nesse caso foi os 20 minutos.

O campo WLM_CAPPING_ACT mostra a porcentagem de tempo que a máquina tentou requisitar recursos e foi impedida devido ao *capping* estar em ação. Por exemplo, suponhamos que a máquina está em *capping*, e um trabalho está requisitando recursos para rodar, devido ao *capping* ele ficou esperando por recurso por 12 minutos, o campo armazenara a porcentagem de tempo que recursos foram negados durante o intervalo de tempo estudado, nesse caso 20.

Os campos WLM_CAPPING_PCT e o WLM_CAPPING_ACT são oriundos do registro SMF do tipo 70, que armazena informações quanto a atividade de CPU (IBM CORP., 2009). A porcentagem de WLM *capping* das partições são registrados no *Condition Name* WCAPPER e usa como fonte os campos do registro SMF SMF70NSW (Número de amostras de diagnóstico em que o WLM considera limitar o conjunto de CPUs lógicas) e SMF70DSA

(Número de amostras de diagnóstico), e o cálculo usado para gerar os valores, é (NSW/DSA) * 100.

***Absolute Capping Limit* (*hardware capping*)**

O campo responsável pelo fracionamento de processadores está presente no mapa SMF70BPD (Seção de Dados do Processador Lógico PR/SM.) do registro 70, que armazena a informação do número absoluto de CPU disponíveis. Sua fonte os dados vêm do campo SMF70HW_ Cap_Limit, com cálculo baseado em HW_Cap_Limit/100. No caso do mesmo *capping* feito em grupo de partições, usando com fonte de dados o campo SMF70HWGr_Cap_Limit do registro 70, com cálculo HWGr_Cap_Limit/100 (IBM CORP., 2009).

Para nos mostrar a capacidade da máquina, usamos o seguinte cálculo:

$$(MIPSporCP * n°CPUAbsolutodisponivel)$$

Porém, não temos o valor de CPU Absoluto na tabela MVSPM_LPAR_H, então fazemos o cálculo de forma reversa, pegando o total da capacidade em MIPS disponível na máquina (PROC_CAPACITY_MSU, campo SMF70WLA do registro SMF) e dividindo pela capacidade real dos processadores (MIPS_PER_LOG_CP), assim terá o valor de n° CPU Absoluto.

$$(PROC \quad * MIPS \quad) = n°CPU| \quad |$$

Defined capacity

Para *capping* do tipo DC ou GC, é usado o campo CAPACITY_LIMIT_MSU, que contêm o valor em MSU da capacidade máxima que a máquina deve entregar, seu campo no registro 70 SMF é o, SMF70MSU (IBM Corp, 2009). O valor deve ser convertido em MIPS para uma padronização da entrega dos valores. Para se descobrir o fator de conversão MSU/MIPS, basta pegar o valor da capacidade média da CEC em MIPS e dividir por sua capacidade em MSU. Com o valor do fator de conversão, basta multiplicar o valor do campo pelo fator que terá o valor da capacidade absoluta em MIPS. O campo está também disponível na tabela VSPM_LPAR_MSU_T.

Através dessas três abordagens, podemos controlar a cobrança que o cliente receberá no mês seguinte. Os contratos são baseados no pico da média das quatro horas rolantes (4HRA) no mês (Sharma; Gupta, 2019), porém há casos que o cliente é cobrado pela média do pico da média de consumo horaria de cada dia no mês.

Considerando a análise realizada sobre os ambientes, o cliente (1) é cobrado pelo pico das 4 médias horaria das horas rolantes do mês. Seu ambiente, era usado apenas balanceamento de pesos para o controle de consumo, fazendo com que a o consumo da CEC

ultrapassasse o limite médio definido. O gráfico abaixo mostra o consumo da máquina chegando a 200% do esperado, em um intervalo horário. Como consequência, empurrando a média das 4 horas rolantes para cima e posteriormente a cobrança além do esperado.

A solução nesse caso foi a implementação de um *hardware capping*, limitando a quantidade de GCP disponível. Nesse caso mantendo apenas 1 GCP com entrega média (LSPR) de 1.345 MIPS.

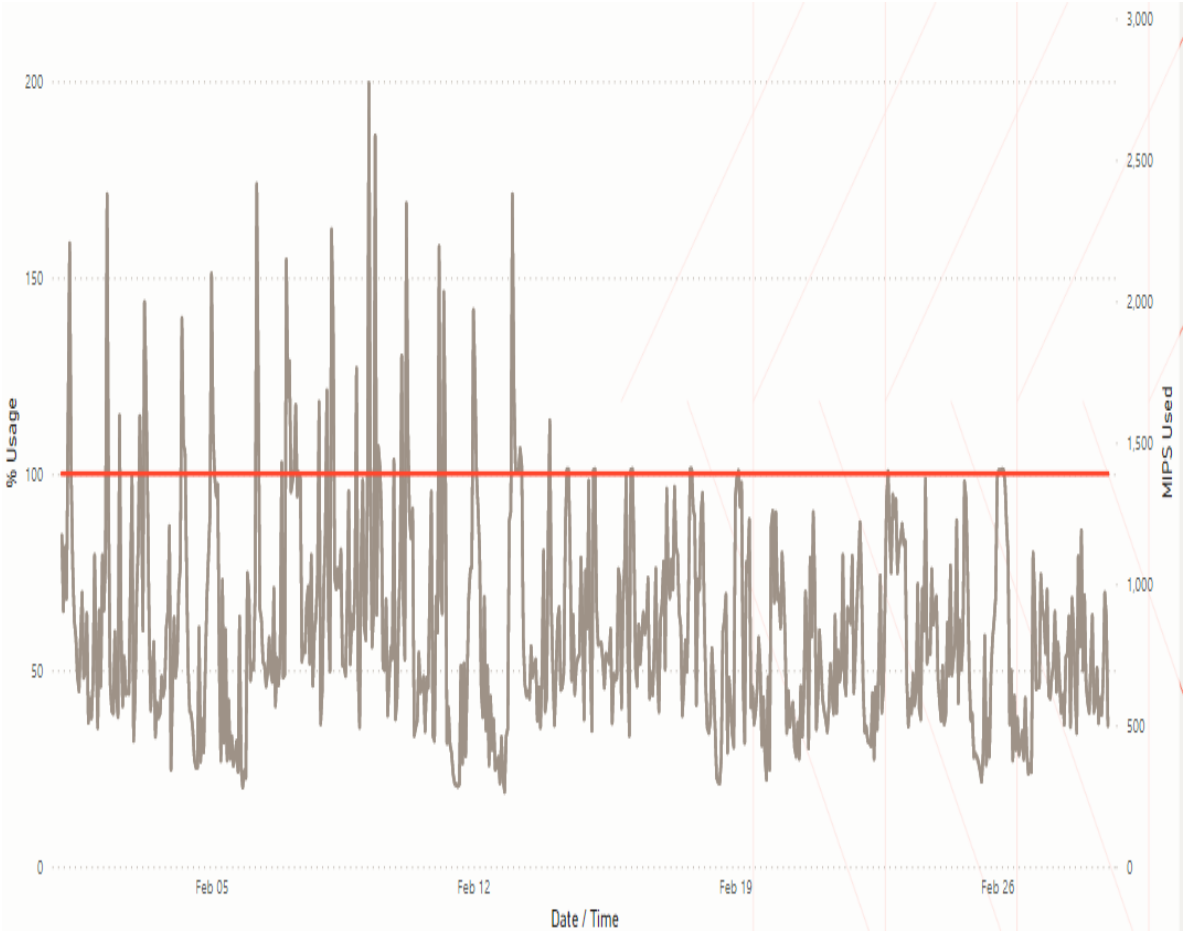
Em alguns momentos a máquina pode passar poucos MIPS da capacidade definida, sua curva de desempenho da capacidade de carga de trabalho, a LSPR, que se altera devido a quatro fatores que alteram seu desempenho, são elas, o comprimento do caminho da instrução, a complexidade da instrução, a hierarquia de memória, e a Intensidade relativa do ninho (RNI).

Para maiores informações e a melhor compreensão do leitor, sugere-se a leitura da explicação no site da IBM sobre como esse recurso funciona em detalhes. Dependendo da memória utilizada, o processador consegue entregar uma capacidade maior, logo poderá ultrapassar a capacidade planejada.

Na figura abaixo, podemos notar que o cliente fez a implementação no dia 15 de fevereiro de 2023, o gráfico mostra que o consumo ultrapassa muito pouco o 100% da capacidade devido a LSPR.

Figura 1

Consumo do cliente 1 em período de um mês



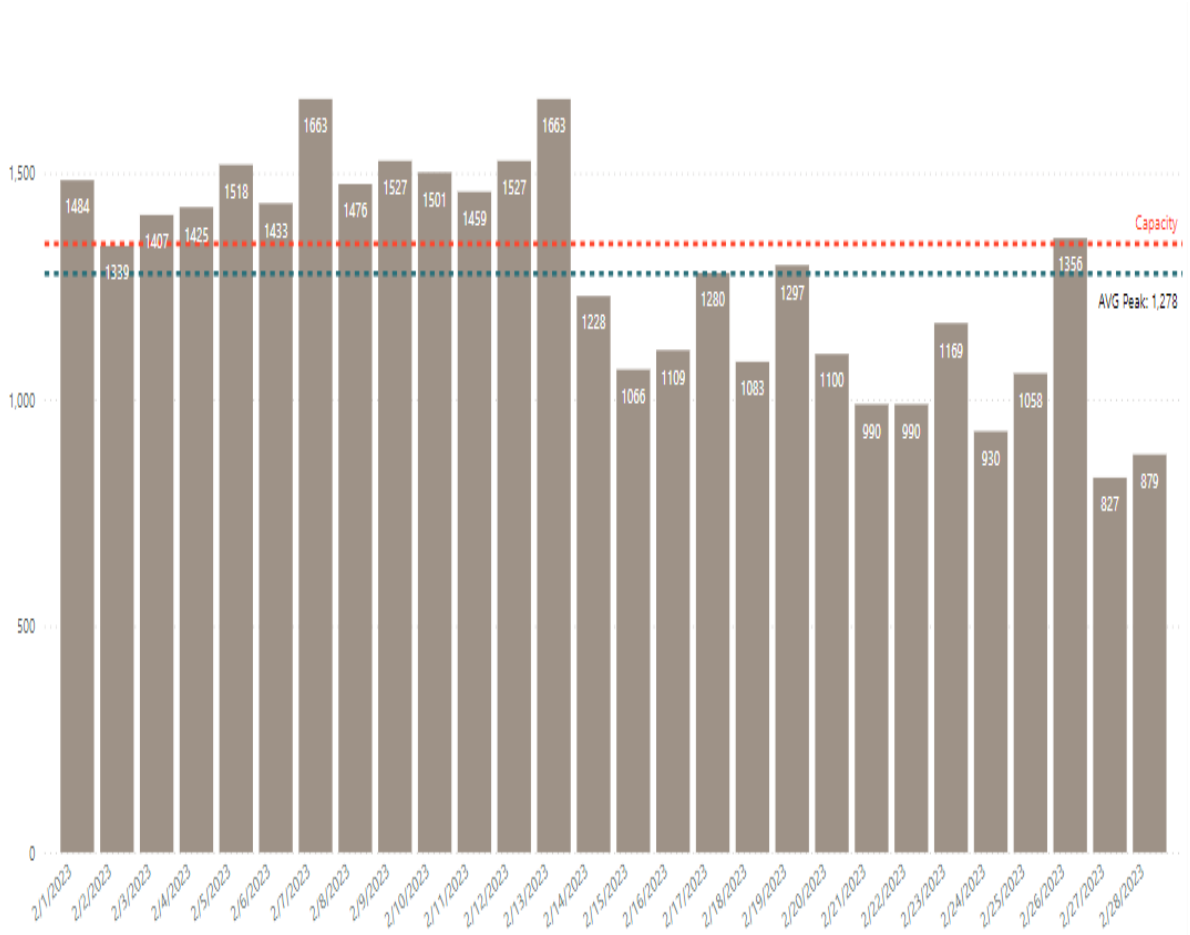
Nota: autoria própria.

Como consequência, os picos da média de consumo das 4 horas rolantes não devem ultrapassar a capacidade, refletindo na cobrança no final do mês do cliente. O gráfico abaixo mostra exatamente esse pico em cada dia dentro da janela de tempo de um mês fiscal.

Pode-se notar que dentro desse mês, sem a implementação do *capping*, dia 15, o cliente arcara com o custo de 1663 MIPS como valor base. Já após o *capping*, o maior pico é o de 1356 MIPS, uma redução de aproximadamente 18,5%, que deve se refletir na cobrança dos meses seguintes.

Figura 2

Valores de picos diários das 4 HRA.



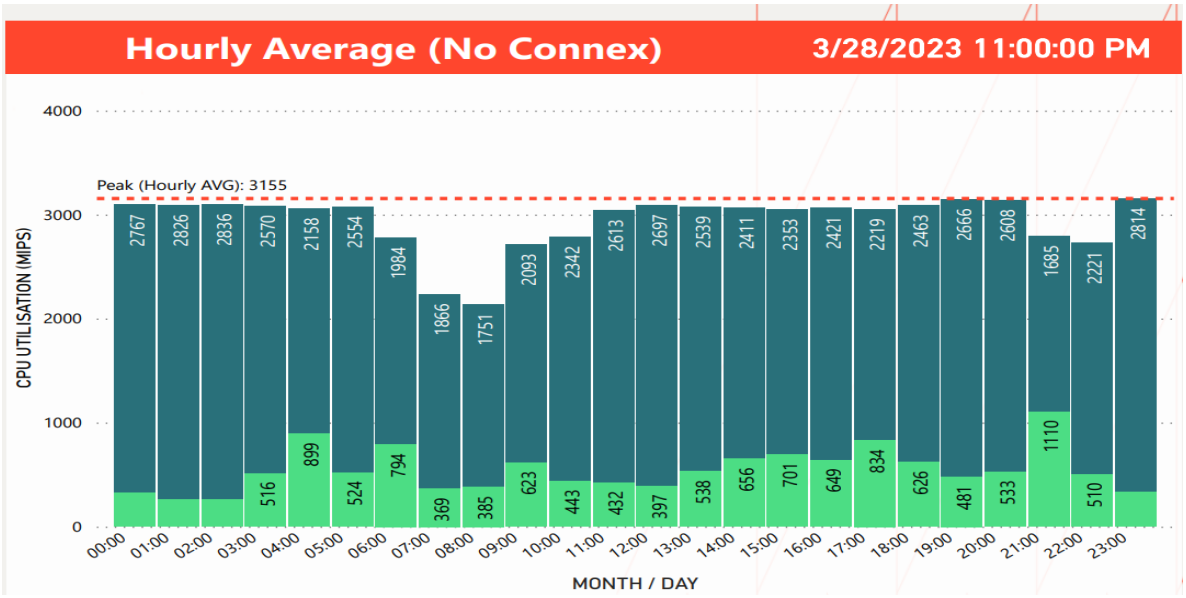
Nota: autoria própria.

O cliente (2) é cobrado por base na média dos picos das medias horarias dos dias no mês. Sendo assim, *capping* relacionados com as 4HRA não seria interessante, pois os valores poderiam ultrapassar o valor de consumo alvo, já que o sistema com *capping* baseado nas 4HRA pode ultrapassar sua capacidade por horas antes de que seja limitado.

A figura abaixo mostra o pico de consumo baseado na média horário. E a seguinte, mostra o resumo do mês de todos os picos que ocorreram no mês e a média destes.

Figura 3

Consumo médio horário do cliente 2.



Nota: autoria própria.

Figura 4

Picos do consumo média horárias de cada dia, e ao final o valor da média final no mês em cada LPAR.

DATE	PEAK TIME	LPAR1	LPAR2	TOTAL
3/1/2023	12:00 AM	3607	395	4002
3/2/2023	11:00 PM	2945	324	3269
3/3/2023	11:00 PM	2826	361	3187
3/6/2023	11:00 PM	2631	477	3108
3/7/2023	11:00 PM	2715	454	3169
3/8/2023	11:00 PM	2731	380	3111
3/9/2023	01:00 AM	2760	477	3237
3/10/2023	11:00 PM	2826	360	3186
3/14/2023	11:00 PM	2742	431	3173
3/15/2023	11:00 PM	2769	433	3202
3/16/2023	12:00 AM	2761	501	3262
3/17/2023	11:00 PM	2822	365	3187
3/20/2023	01:00 PM	2608	514	3122
3/21/2023	11:00 PM	2732	442	3174
3/22/2023	01:00 AM	2745	366	3111
3/23/2023	01:00 AM	2776	465	3241
3/24/2023	11:00 PM	2831	354	3185
3/27/2023	11:00 PM	2724	378	3102
3/28/2023	11:00 PM	2814	341	3155
3/29/2023	03:00 AM	2791	404	3195
3/30/2023	12:00 AM	2726	497	3223
3/31/2023	12:00 PM	2905	308	3213

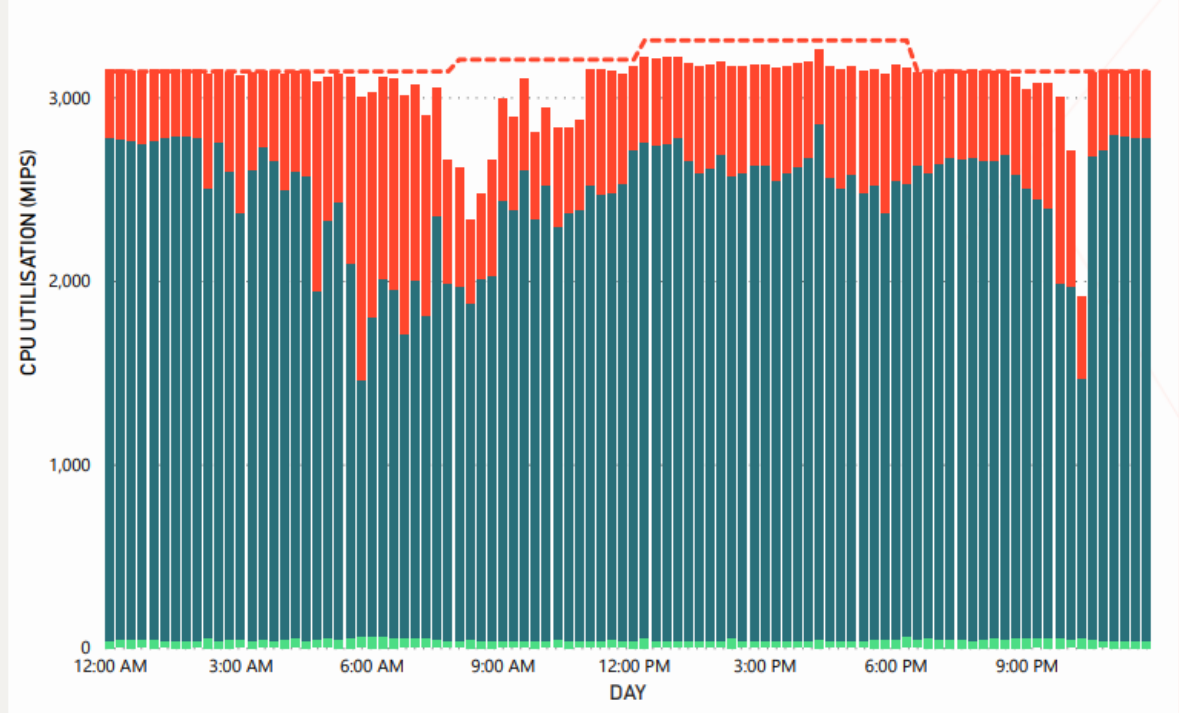
Nota: autoria própria.

A solução para o cliente foi a implementação de um *hardware capping* (*Absolute Capping Limit*). O sistema nesse caso entende que só pode entregar um valor MSU baseado na quantidade de processor definido no HMC, não cedendo quando há uma necessidade maior.

O cliente no caso oscila seu capeamento ao longo do dia. Na figura abaixo, o eixo x cobre um período de 24 horas, podemos notar que a linha tracejada que representa a capacidade da máquina (*capping*) e o seu consumo, as colunas verticais que representa o consumo médio em um intervalo de 15 minutos, em nenhum momento ultrapassa o limite do *capping*.

Figura 5

Consumo médio em intervalo de 15 minutos do cliente 2 com a capacidade sendo exibida na linha tracejada.



Nota: autoria própria.

Com essa técnica é possível o cliente controlar suas medias de consumo e sua fatura do mês.

O cliente (3) é cobrando assim como o cliente (1), por base no pico das 4 HRA do mês. Porém, seu modelo de *capping* é diferente. O cliente optou pelo uso do WLM *Capping*, um capeamento feito via *software* nativo do z/OS.

Um valor de 4 HRA MSU é pré-definido e quando é atingido, os recursos da máquina são limitados até que sua média rolante volte a cair.

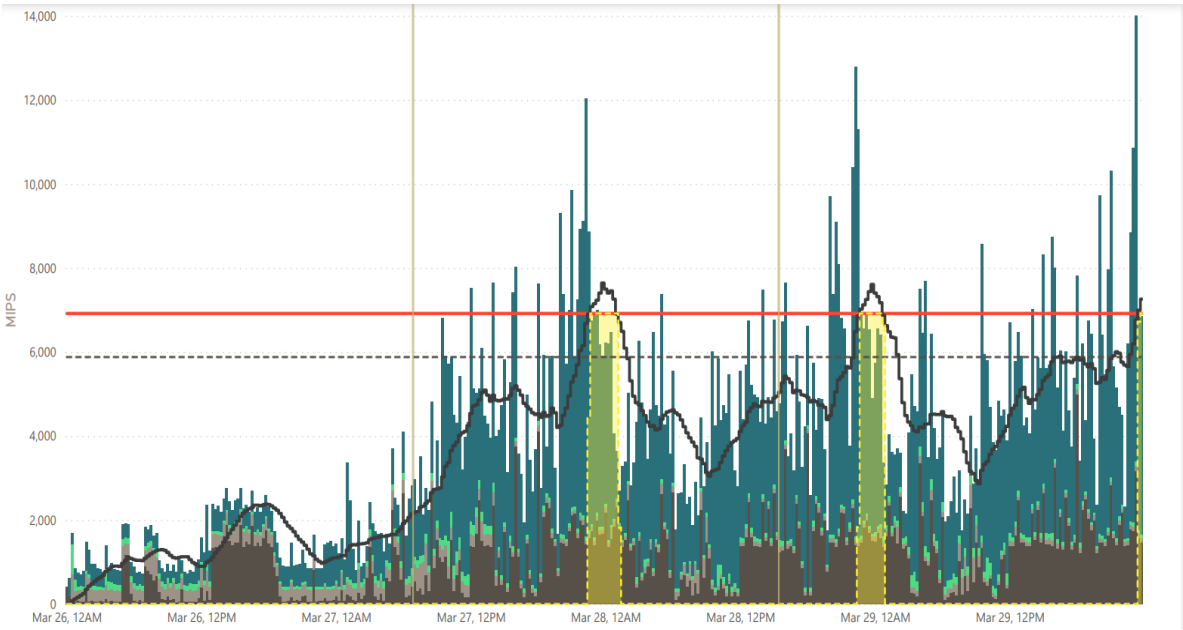
No gráfico abaixo, as colunas verticais representam o consumo médio em um intervalo de 15 minutos, a linha solida laranja horizontal a capacidade (estabelecida pelo WLM

Capping), a linha preta solida mostra as 4 HRA e a sombra amarela demarcada por uma linha tracejada mostra o período que o sistema estava sobe o efeito de *capping*.

Podemos notar que o *capping* fica ativo no exato momento em que as 4 HRA toca na linha de capacidade e só volta a ficar inativo quando as 4 HRA volta a cair abaixo da capacidade.

Figura 6

Consumo médio em intervalos de 15 minutos do cliente 3.



Nota: autoria própria.

Nesse modelo de *capping* o cliente pode sofrer uma pequena cobrança acima do esperado (valor pré-definido pelo *capping*), pois dependendo da agressividade que o consumo da máquina aumenta nas horas anteriores a ativação do *capping*, as 4 HRA pode ultrapassar um pouco a capacidade.

O lado positivo desse modelo de *capping* é que o cliente pode não sofrer com a falta de recursos por picos de consumo atípicos, pois picos isolados não tem forças para limitar o sistema. Porém, neste modelo, os recursos podem levar um pouco mais tempo para serem liberados se o consumo de manter relativamente alto, próximo a capacidade.

O interesse parte quase sempre do cliente na redução de sua fatura, a equipe de Performance e Capacidade pode criar simulações e planejamento de como essa mudança de capacidade da máquina pode interferir no ambiente.

O cliente sempre visa pela maximização dos lucros, partirá sempre do princípio que dá para economizar mais, porém o time de PCM deve evidenciar os riscos quanto a suas escolhas, mostrando que transações podem não ocorrer devido à falta de recurso no momento requisitado.

Ambas as partes travarão batalhas defendendo seus interesses, entre a saúde e a confiabilidade do ambiente e o custo mínimo possível. As partes podem tomar decisões erradas, o cliente por falta de conhecimento técnico, assim como a equipe técnica prefere tomar medidas cautelosas a fim e evitar reclamações futuras.

2.1.3 Propostas de Solução

capping em mainframes é uma técnica usada para controlar a quantidade de recursos (como CPU, memória e I/O) que um aplicativo pode consumir no sistema. A limitação é normalmente usada para gerenciar picos de carga de trabalho e para garantir que os aplicativos não consumam muitos recursos, o que pode causar problemas de desempenho ou interromper outros aplicativos em execução no mesmo sistema.

As técnicas de limitação podem ajudar os clientes a gerenciarem suas cargas de trabalho de mainframe e controlar os custos, e em conjunto com o WLM garantir que os aplicativos críticos recebam os recursos necessários para serem executados com eficiência.

Existem vários tipos de técnicas de *capping* disponíveis no ambiente z/OS, cada uma com suas características.

Initial capping: é uma técnica usada para controlar o número máximo de MSUs pela distribuição de processadores lógicos para cada LPAR baseado em seus pesos. É a técnica mais antiga de *capping* disponível. Exemplo, *Weigth Capping* (Frank Kyne et al., 2001; Ottaviani, 2016).

Soft capping: é uma técnica usada para controlar a quantidade máxima de recursos o sistema pode consumir, é sensível as 4 HRA. O *soft capping* permite que um aplicativo exceda o limite máximo se houver recursos não utilizados disponíveis no sistema. Exemplo, LPAR *Capping* (Frank Kyne et al., 2001). Por sua vez Baker (2019) reforça que se trata de uma técnica comum com o objetivo de controlar os cursos de software fundamentado no R4HA. Ottaviani (2016) reforça que diferente pesos podem e normalmente precisam ser especificados para cada conjunto de CP.

Hard capping: é uma técnica usada para definir um limite rígido na quantidade máxima de processador que o sistema pode consumir, não é sensível as 4 HRA, e seu *capping* é instantâneo, sua definição é feita direto no HMC. Se um aplicativo exceder o limite rígido, seu desempenho poderá ser afetado ou poderá ser encerrado. Exemplo, *Absolute Capping Limit*, PR/SM *capping* (Rogers; Salla, 2010).

Absolute MSU Capping: O limite é controlado pelo WLM, não sensível as 4 HRA. É semelhante ao *Initial Capping* controlado por PR/SM. A diferença é que o limite absoluto de MSU é especificado em MSUs (DC – Cada LPAR precisa de sua especificação na PARMLIB IEAOPTxx com a opção AbsMsuCapping = YES), enquanto o *Inicial Capping* é relativo ao peso (Rama, 2023).

Defined Capacity (DC) e *Group Capacity* (GC) *Capping*: é uma nomenclatura usada para definir se o *capping* irá afetar uma única LPAR, no caso da DC, ou *Group Capacity* (GC) para múltiplas LPARs. Essa técnica é frequentemente usada para garantir que um sistema de mainframe não exceda sua capacidade licenciada, é sensível as 4 HRA e ao consumo MSU/h. Exemplo, LPAR *Capping*, *Phantom weight*, *Pattern capping*, *Weigth Capping* (Rogers; Salla, 2010)

Através da técnica de *capping*, é possível assim delimitar a cobrança do ambiente do cliente, tornando sua fatura mais previsível, já que são baseadas na utilização. Dependo do modelo de *capping* escolhido, a previsão poderá ser mais ou menos assertiva, algumas cedem recursos quando o limite é atingido, outros não.

2.1.4 Riscos Iminentes

Conforme Freitas (2002, p.42), o conceito de riscos “Consiste na probabilidade de que um evento indesejado venha a ocorrer associada as consequências desse evento”. Assim, ressalta-se algumas ponderações, tais como: limitar o principal componente de um mainframe, seu CPU, traz grandes riscos. Por isso deve-se desenvolver um bom planejamento com o time de PCM. O time deve ser experiente e detentor de conhecimento de *capping* e simulações, com isso a chance de fracasso, diminuem. Todos os riscos podem ser mitigados, jamais ignorados, alguns dos riscos incluem

- Perda de desempenho: se o *capping* não for implementado corretamente, pode causar uma queda no desempenho em pontos críticos ou causar lentidão no desempenho

geral do sistema. Isso pode ocorrer se os limites máximos definidos forem muito baixos ou se o sistema não conseguir alocar recursos com eficiência.

- Incompatibilidade com programas: em alguns casos, o *capping* pode fazer com que programas se tornem instáveis ou travem. Isso pode acontecer se o aplicativo não for projetado para operar em condições de *capping* ou se os limites de *capping* forem muito baixos para a carga de trabalho do programa.
- Conformidade contratuais: o *capping* pode não estar em conformidade com determinados contratos de licenciamento de *software*, principalmente aqueles que exigem o uso da capacidade total do mainframe. O uso de técnicas de *capping* para limitar o consumo de recursos podem violar esses acordos e expor o cliente a possíveis penalidades legais ou financeiras (Lipovich, 2016).
- Complexidade: A implementação de pode ser complexa e requer planejamento e gerenciamento cuidadosos. Essa complexidade pode levar a erros ou configurações incorretas que podem causar problemas no sistema ou afetar o desempenho (Lipovich, 2016).
- Custo: embora o *capping* ajude a controlar os custos ao limitar o consumo de recursos, também pode incorrer em custos adicionais associados à implementação, gerenciamento e monitoramento.

Sendo assim, ampliar ações que poderão prevenir situações indesejadas que poderão promover prejuízos de diversas naturezas, é imprescindível.

2.1.5 Análise dos resultados

Devido à falta de dados históricos do cliente 2 e 3, e as mudanças de capacidade acontece em uma frequência muito alta, não é possível realizar as métricas de comparação de antes e depois do *capping*, podemos observar apenas como eles vem controlando a capacidade da máquina.

Referente ao cliente 1, nota-se que sem a implementação do *capping*, dia 15, o cliente arcara com o custo de 1663 MIPS como valor base devido ao seu pico de uso. Já após o *capping*, o maior pico é o de 1356 MIPS, uma redução de aproximadamente 18,5%, que deve se refletir na cobrança dos meses seguintes e não ultrapassar o valor estipulado do *capping*.

3 CONSIDERAÇÕES FINAIS

Compreende-se que o estudo alcançou os seus objetivos propostos tendo em vista que demonstrou os impactos existentes em três ambientes com capping ativos, por meio de gráficos, e por sua vez indicou soluções para economia na fatura mensal. Desta forma, cabe destacar que o *capping* é uma técnica útil para gerenciar cargas de trabalho e recursos em sistemas de mainframe.

Todavia, foi possível identificar que ao definir limites para a quantidade de recursos que sistema pode consumir, as organizações devem balancear entre ter um sistema com eficiência, evitando a contenção de recursos e problemas de desempenho, ou uma economia. Existem vários tipos de técnicas de *capping* disponíveis para mainframes, cada uma com suas próprias vantagens e considerações.

A implementação de técnicas de *capping* requer planejamento cuidadoso e consideração dos requisitos de carga de trabalho da organização, padrões de uso de recursos e restrições de custo. Nesta perspectiva, registra-se que com a implementação e gerenciamento adequados, o *capping* pode ajudar as organizações a obterem desempenho e utilização de recursos ideais em seus sistemas de mainframe economizando na sua fatura.

Importante citar que as limitações encontradas no trabalho foram as já mencionadas na análise dos resultados, como a falta dos históricos dos clientes 2 e 3, contudo, pode-se notar como ambas controlam seus custos pela técnica descrita no estudo. Desta forma não foi possível a visualização de forma gráfica dos resultados destes clientes, devido a limitação imposta pelos mesmos, na ferramenta de gerenciamento dos dados para armazenar informações de curto prazo (necessárias para tal estudo) por apenas 3 meses.

Sendo assim, sugere-se outras pesquisas que possam analisar, como por exemplo, há diversos artigos disponíveis no site Planet Mainframe criado por Baker (2019) que faz referência entre custos e *capping*.

De toda forma, compreende-se que esse trabalho poderá contribuir tanto com as empresas, bem como para os profissionais da área de performance e capacidade, e principalmente, para os clientes que poderão ter redução de custos e um ambiente controlado.

REFERÊNCIAS

Baker, J. (2019) Controlling MLC with Capping: MLC Capping 101. [S. l.]: IntelliMagic, Disponível em: https://www.intellimagic.com/resources/zos/blog/controlling-mlc-with-capping/.(Site).

Frank K; Ferguson, M; Russell, T; Salla, A; Trowell, K. (2001). Z/OS Intelligent Resource Director. 5. ed. USA: IBM, 412 p. ISBN 0738417904. (Livro).

IBM Corp. (2023) IBM Z Large Systems Performance Reference. 6910157. USA. Disponível em: https://www.ibm.com/support/pages/ibm-z-large-systems-performance-reference. [Site].

IBM Corp. (2023) Large Systems Performance Reference. 25. ed. U. S.: IBM. 70 p. v. 10. ISBN SC28-1187-25. [Site]

IBM Corp. (2023) LSPR workload categories. USA. Disponível em: https://www.ibm.com/support/pages/lspr-workload-categories. Acesso em: 23 abr. 2023. [Site]

IBM Corp. (2023). Relative nest intensity. 5.6. USA, 2023. Disponível em: https://www.ibm.com/docs/en/cics-ts/5.6?topic=terminology-relative-nest-intensity. [Site]

IBM Corp. (2009). Tivoli Decision Support for z/OS: System Performance Feature Reference Version 1.8.1. 12. ed. USA: IBM, 2009. 70 p. v. 1. ISBN SH19-6819-12.

IBM Corp. (2023) What is cloud migration?: Case studies. USA, 2023. Disponível em: https://www.ibm.com/topics/cloud-migration. Acesso em: 23 abr. 2023.

International Business Machines Corporation. (2017) IBM Z Decision Support: System Performance Feature Reference. 1.9. ed. USA: IBM, 2017. 1062 p. v. 1. (Livro)

Lipovich, J. (2016) Five Myths of Mainframe Capping. [S. l.], 4 nov. (Blog)

Ottaviani, M. F. (2016) The many effects of the LPAR weight. EPV Technologies. p. 1-10. [Artigo]

Parziale, L; Benke, O; Favero, W; Kumar, R; Lafalce, S; Madera, C; Muszytowski, S. (2016). Enabling Real-time Analytics on IBM z Systems Platform. 5. ed. USA: IBM, 198 p. [Livro].

Rama, H. (2023) Options to Curb Mainframe Capacity Demand. [Blog]

Rogers, P; Salla, A. (2010). ABCs of z/OS System Programming. 2. ed. U. S.: IBM, 252 p. v. 11. [Livro]

Salla, A; Oughton, P. (2018). ABCs of z/OS System Programming. 6. ed. U. S.: IBM, 146 p. v. 10. ISBN 0738443107. [Livro]

Sharma, S; Gupta, S. (2019). Are You Using the Right 4-Hour Rolling Average?. Right 4-Hour Rolling Average, USA, p. 1-6. [Artigo]

Sinram, H. (2015) Capping, Capping, and Capping: A Comparison of Hard and Soft-capping Controls. 16821. ed. Seattle: SHARE. [Slides].

Flick, U. (2009). Introdução à Pesquisa Qualitativa. Porto Alegre/RS. Artmed. [Livro]

Freitas, C. A. S. (2002) Gestão de Riscos: possibilidades de utilização pelo setor público e por entidades de fiscalização superior. Revista TCU. Brasília. v. 33, n.93. pp.42-54. [Artigo].

Yin, R. K. (2001). Estudo de Caso: planejamento e métodos. Porto Alegre/RS. Bookman. [Livro].

O uso de *capping* em mainframe dedicados em zCloud: redução de custos em ambiente IBM z/OS

The use of capping on dedicated mainframes in zCloud: cost reduction in IBM z/OS environment

ABSTRACT

IBM's mainframe technology-based cloud computing platform, designed to provide enterprise-grade cloud computing services for companies that require high levels of security, availability and performance, leveraged the need to develop a process that managed to share a single machine with several companies, guaranteeing the resources agreed in the contract. Therefore, considering that most companies use the cloud, making it a recurring issue for those who work with the platform, capping is essential to delimit the machine's resources for each environment. From this perspective, a case study was carried out with the objective of demonstrating the impacts of three environments with active capping, considering that companies are looking for solutions to reduce their costs with monthly invoices.

Keywords: Mainframe; Batch; Db2; Z/os.

El uso de limitación en mainframes dedicados en zCloud: reducción de costos en el entorno IBM z/OS

ABSTRACTO

La plataforma de computación en la nube basada en tecnología mainframe de IBM, diseñada para brindar servicios de computación en la nube de nivel empresarial para empresas que requieren altos niveles de seguridad, disponibilidad y rendimiento, aprovechó la necesidad de desarrollar un proceso que lograra compartir una sola máquina con varias empresas. , garantizando los recursos pactados en el contrato. Por lo tanto, considerando que la mayoría de las empresas utilizan la nube, lo que la convierte en un problema recurrente para quienes trabajan con la plataforma, la limitación es fundamental para delimitar los recursos de la máquina para cada entorno. Desde esta perspectiva, se realizó un estudio de caso con el objetivo de demostrar los impactos de tres entornos con capping activo, considerando que las empresas buscan soluciones para reducir sus costos con facturas mensuales.

Palabras clave: Computadora central; Lote; Db2; Z/os.