LVI SBPO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
DE 04 ATÉ 07 NOVEMBRO 2024
FORTALEZA | CEARÁ

Assegurando a Confidencialidade de Dados de *Workflows* Executados em Nuvens de Computadores: Abordagens Heurísticas e Exatas

Rodrigo Silva, Yuri Frota e Daniel de Oliveira
Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói – RJ
rodrigo_prado@id.uff.br, yuri@ic.uff.br e danielcmo@ic.uff.br

Esther Pacitti
INRIA, University of Montpellier, CNRS, LIRMM
Montpellier, France
esther@lirmm.fr

RESUMO

As nuvens de computadores fornecem um ambiente sob demanda que permite aos usuários executar seus *workflows* locais em um ambiente elástico e com alta disponibilidade. Diversas aplicações podem ser modeladas como *workflows*, e muitas delas são intensivas em computação e produção de dados. Na nuvem, o local de armazenamento desses dados se torna uma preocupação quando a confidencialidade pode ser comprometida. Usuários maliciosos podem realizar inferências a respeito dos resultados e da própria estrutura dos *workflows*. A dispersão de dados, a criptografia e outros mecanismos podem ser adotados para aprimorar a privacidade dos dados, mas estes não podem ser adotados sem considerar o escalonamento do *workflow*, pois isso arrisca aumentar significativamente o tempo de execução e o custo financeiro. Neste artigo, introduzimos a CYCLOPS, uma abordagem que visa executar *workflows* em nuvens de computadores de forma eficiente levando em consideração as restrições de confidencialidade dos dados produzidos e da estrutura do *workflow*.

PALAVRAS CHAVE. Armazenamento, Confidencialidade, CYCLOPS, Nuvens, Workflows.

ABSTRACT

Cloud computing provides an on-demand environment that allows users to execute their local workflows in an elastic and highly available environment. Various applications can be modeled as workflows, many of which are compute- and data-intensive. In the cloud, the storage location of this data becomes a concern when confidentiality might be compromised. Malicious users can make inferences about the results and the structure of the workflows. Data dispersion, encryption, and other mechanisms can be adopted to enhance data privacy, but these cannot be implemented without considering the workflow scheduling, as this risks significantly increasing execution time and financial cost. In this paper, we introduce CYCLOPS, an approach that aims to execute workflows in cloud computing environments efficiently while considering the confidentiality constraints of the produced data and the workflow structure.

KEYWORDS. Confidentiality, Clouds, CYCLOPS, Dispersion, Workflows.

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

1. Introdução

Um *Workflow* é uma abstração muito usada para modelar processos complexos. Geralmente representados como grafos acíclicos direcionados (DAGs), os *workflows* possuem nós que representam atividades e arestas que indicam as dependências de dados entre elas [de Oliveira et al., 2019]. A execução de uma atividade, chamada de *ativação* desde ponto em diante, consome arquivos e parâmetros de entrada. Embora possam ser implementados como *scripts*, é comum utilizar Sistemas de Gerência de *Workflows* de *Big Data* (SGWBDs), que permitem aos usuários especificar e executar os *workflows*, além de rastrear os dados de proveniência dos *workflows*.

Devido à alta demanda por recursos, muitos *workflows* são executados em nuvens de computadores. Por isso, vários SGWBDs oferecem suporte para ambientes de nuvem, incluindo o Pegasus, o Parsl e o SecDATAVIEW. Escalonar as ativações do *workflow* na nuvem é um problema NP-Difícil. Os SGWBDs procuram oferecer algoritmos de escalonamento otimizados que aproveitam os recursos da nuvem, como elasticidade e modelos de pagamento sob demanda. No entanto, o compartilhamento de recursos e o uso de áreas de armazenamento compartilhadas, conhecidas como *buckets*, expõem os *workflows* a várias classes de ataques [Zamfiroiu et al., 2019]. Por exemplo, Ristenpart et al. [2009] discutem diversas ameaças à confidencialidade dos dados armazenados em máquinas virtuais (MVs) da AWS e destacam a importância de proteger os resultados das execuções dos *workflows* de *big data*.

Uma forma de reduzir o impacto de possíveis ameaças é por meio da criação de um plano de dispersão de dados que determina quais arquivos podem ser armazenados juntos, garantindo a confidencialidade dos dados na nuvem. Esse plano é representado neste artigo por um **grafo de conflitos**, onde cada arco indica um conflito entre os arquivos, sendo crucial conhecer essas restrições para sua geração. Desenvolver um plano eficaz para os dados consumidos e produzidos pelos *workflows* é essencial para aumentar a confidencialidade dos dados em SGWBDs. Entretanto, a dispersão dos dados não pode ser feita desacoplada do escalonamento do *workflow*, pois dependendo de onde os dados são armazenados e processados, pode haver impacto no tempo de execução do *workflow*.

Neste artigo, é apresentada a abordagem denominada `CYCLOPS` , que preserva a confidencialidade dos dados enquanto reduz os custos e o tempo de execução dos *workflows*. Na `CYCLOPS`, as ativações são escalonadas considerando a heterogeneidade das MVs, usando metadados de segurança e o **grafo de conflitos** para dispersar eficientemente os dados entre os dispositivos de armazenamento.Isso minimiza o *makespan* e o risco de acesso malicioso, ao mesmo tempo que pode ser integrado aos SGWBDs existentes.

As contribuições deste trabalho podem ser enumeradas como: a formulação do problema de escalonamento com confidencialidade dos dados como um problema de programação matemática mista (i); o desenvolvimento de uma heurística de escalonamento de *workflows* com restrições de confidencialidade (ii); e uma avaliação experimental da abordagem proposta usando *workflows* de *benchmark* conhecidos (iii). O restante deste documento está organizado da seguinte forma: a Seção 3 aborda a formulação matemática; a Seção 4 apresenta a heurística de escalonamento proposta; a Seção 5 exibe os resultados experimentais; a Seção 2 discute os trabalhos relacionados; e a Seção 6 conclui o trabalho.

2. Trabalhos Relacionados

Diversos trabalhos têm como foco executar *workflows* em nuvens minimizando o *makespan* e custo financeiro. Como o enfoque deste trabalho está na segurança, serão apresentados os trabalhos que de alguma forma tratam de questões como segurança da informação e confidencialidade dos dados. Os trabalhos revisados são categorizados em 3 grupos: (i) dispersão dos dados; (ii) segurança por MV; e (iii) segurança por *site*. A primeira compreende os trabalhos que de alguma forma distribuem os dados pelos diversos dispositivos de armazenamentos, a segunda

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

engloba os trabalhos que asseguram as informações associando MVs capazes de atender as demandas de seguranças das ativações e, por fim, a terceira abrange os trabalhos que vinculam diferentes *sites* (pode-se considerar também regiões, *data centers*, nuvens privadas) a diferentes requisitos de segurança, mapeando assim as ativações mais sensíveis aos *sites* mais seguros. Na categoria de dispersão de dados, destaca-se o trabalho do Guerine et al. [2019], que propõem a dispersão dos dados em *buckets* na nuvem. Apesar de considerar tanto a dispersão de dados quanto a criptografia na solução final, a dispersão de dados é desvinculada do escalonamento das ativações do *workflow*, *i.e.*, os dados podem ser armazenados em um dispositivo geograficamente muito distante da MV que irá processar os dados significando em um maior *overhead* na leitura/escrita dos dados. Na categoria de segurança por MV, trabalhos como Shishido et al. [2018] e Tawfeek e AbdulHamed [2018] implementam algoritmos baseados em algoritmos genético para escalonar ativações que produzem dados sensíveis em MVs específicas, ou seja, que possuem níveis ou padrões de segurança que atendam os requisitos de segurança. Similarmente, Sujana et al. [2019] e Abazari et al. [2019] propõem abordagens de escalonamento multicritério que consideram os requisitos de segurança das ativações do *workflow*. Todos esses trabalhos deixam de aproveitar uma maior segurança e confiabilidade que a dispersão dos dados é capaz de proporcionar. Na categoria de segurança por *site*, Sharif et al. [2016] e Wen et al. [2020] propõem algoritmos para preservar a privacidade, associando ativações e dados a nuvens privadas ou *data centers* específicos. Por sua vez, Zhou et al. [2019] definem as capacidades de segurança e privacidade de cada *data center*, distribuídos geograficamente, para a execução das ativações, visando atender aos requisitos de segurança e otimizar tempo e custo. No entanto, essas abordagens não abordam o acesso indevido a dados sensíveis em violações de segurança interna nem criptografia. Por outro lado, a abordagem proposta neste artigo introduz uma barreira a mais através da dispersão dados junto com o cálculo do escalonamento.

3. Formulação Matemática

Workflow	**Descrição**
D_s	Conjunto de dados estáticos (dados já existentes).
D_d	Conjunto de dados dinâmicos (dados produzidos durante a execução do *workflow*).
$D = D_s \cup D_d$	Conjunto de dados.
$O(d)$	Dispositivo que armazena dados estáticos $d \in D_s$.
$W(d)$	Tamanho dos dados $d \in D$.
N	Conjunto de ativações.
B	Conjunto de *buckets* (somente com serviço de armazenamento), com preço de pagamento variável, dependendo de quantos gigabytes são armazenados.
M	Conjunto de máquinas virtuais (com poder de processamento e serviço de armazenamento).
$\overline{M} = (B \cup M)$	Conjunto de dispositivos.
L_j	Conjunto de intervalos de armazenamento $\{0 \ldots \mid L_j \mid\}$ do *bucket* $j \in B$ (intervalo L_0 indica que o *bucket* não é usado).
t_{max}	Tempo máximo de execução para o *workflow*.
T	Conjunto de períodos de tempo viáveis ($T = \{1 \ldots t_{max}\}$).
t_{ij}	Tempo de processamento de ativação $i \in N$ na máquina virtual $j \in M$.
$\overrightarrow{t}_{djp}$	Tempo gasto pela máquina virtual $j \in M$ para ler os dados $d \in D$ armazenados no dispositivo $p \in \overline{M}$.
$\overleftarrow{t}_{djp}$	Tempo gasto pela máquina virtual $j \in M$ para escrever os dados $d \in D_d$ armazenados no dispositivo $p \in \overline{M}$.
$\Delta_{in}(i) \subseteq D$	Conjunto de dados necessários para a execução da ativação $i \in N$.
$\Delta_{out}(i) \subseteq D_d$	Conjunto de dados gerados pela ativação $i \in N$.
cm_j	Capacidade de armazenamento do dispositivo $j \in \overline{M}$.
sm_{jl}	Volume de armazenamento do *bucket* $j \in B$ no intervalo $l \in L_j$, onde $sm_{j0} = 0$.
c_j^M	O custo financeiro da contratação da máquina virtual $j \in M$ por um período de tempo.
c_{jl}^B	O custo financeiro da contratação do *bucket* $j \in B$ no intervalo $l \in L_j$, onde $c_{j0}^B = 0$.
c_{max}	O orçamento máximo financeiro disponível.

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

Workflow	**Descrição**
α_t, α_b and α_s	Os pesos de tempo, custo e segurança que definem a relevância de cada objetivo ($\alpha_t + \alpha_b + \alpha_s = 1$).

Segurança e Privacidade	**Descrição**
$G_c = (D, E_h \cup E_s, \delta)$	Grafo de conflitos.
R	Conjunto de requisitos de segurança de ativações.
l^r_{max}	O valor máximo (nível) do requisito de segurança $r \in R$.
l^{ri}_{task}	O nível do requisito mínimo de segurança $r \in R$ exigido pela ativação $i \in N$.
l^{rj}_{vm}	O nível do requisito de segurança $r \in R$ oferecido pela máquina virtual $j \in M$.
s_{max}	A exposição máxima de segurança e privacidade, definida como $s_{max} = \sum_{r \in R} \lvert N \rvert \cdot l^r_{max} + \sum_{(d_1,d_2) \in E_s} \delta_{d_1 d_2}$.

Variáveis	**Descrição**
x_{ijt}	Variável binária que indica se a ativação $i \in N$ inicia sua execução na MV $j \in M$ no período $t \in T$ ou não.
$\overrightarrow{x}_{idjpt}$	Variável binária que indica se a ativação $i \in N$ executando na MV $j \in M$ começa a ler os dados $d \in \Delta_{in}(i)$ armazenados no dispositivo $p \in \overline{M}$ no período $t \in T$ ou não.
$\overleftarrow{x}_{idjpt}$	Variável binária que indica se a ativação $i \in N$ executando em $j \in M$ começa a escrever dados $d \in \Delta_{out}(i) \subset D_d$ no dispositivo $p \in \overline{M}$ no período $t \in T$ ou não.
y_{djt}	Variável binária que indica se os dados $d \in D$ estão armazenados no dispositivo $j \in \overline{M}$ no período $t \in T$ ou não.
$\overline{y}_{dj}$	Variável binária que indica se os dados $d \in D$ estão armazenados no dispositivo $j \in \overline{M}$ em algum período de tempo.
$w_{d_1 d_2}$	Variável binária que indica se os dados d_1 e d_2 estão armazenados no mesmo local ou não, onde $(d_1, d_2) \in E_s$.
e_{ri}	Variável contínua que indica o nível de exposição da execução da ativação $i \in N$ referente ao requisito de segurança $r \in R$.
b_{jl}	Variável binária que indica se o *bucket* $j \in B$ está sendo usado no intervalo $l \in L_j$ ou não.
q_{jl}	Variável contínua que contém o tamanho total dos dados alocados no *bucket* $j \in B$ no intervalo $l \in L_j$.
v_{jt}	Variável binária que indica se a máquina virtual $j \in M$ é usada (contratada) no momento $t \in T$.
z^T_j	Variável contínua que indica o tempo total de uso da máquina virtual $j \in M$.
z^T	Variável contínua que indica o tempo total para executar o *workflow* (*makespan*).

O escalonamento de ativações do *workflow* realizado pela abordagem `CYCLOPS` é formulado como o problema de programação inteira chamado `CYCLOPS-IP`, conforme descrito a seguir: Um *workflow* pode ser representado como um grafo direcionado acíclico $G = (N \cup D, A)$, onde o conjunto de vértices é composto por ativações (N) e arquivos de dados (D), enquanto que o conjunto de arcos A definem a relação de produção/consumo entre as ativações e os dados. Estes dados podem ser armazenados em MVs $j \in M$ ou em *buckets* $j \in B$, porem apenas MVs possuem poder de processamento, logo, definimos nosso conjunto de dispositivos de $\overline{M} = (B \cup M)$. Além disso, ativações podem requerer requisitos de segurança (*e.g.*, criptografias), definidos pelo conjunto R, que podem ser ofertados pelos dispositivos disponíveis. Cada requisito de segurança $r \in R$ possui l^r_{max} níveis de segurança. Denotamos por l^{rj}_{vm} o nível do requirimento de segurança $r \in R$ oferecido pela MV $j \in M$. Similarmente, denotamos por l^{ri}_{task} o nível do requisito de segurança $r \in R$ requerido pela ativação $i \in N$. Defini-se $D = D_s \cup D_d$ como o conjunto de todos os dados, onde cada dado $d \in D$ possui um tamanho $W(d)$ e pode ser estático (D_s), com um dispositivo de origem $O(d) \in \overline{M}$, ou dinâmico (D_d), gerado durante a execução do *workflow*. Além disso, define-se $\Delta_{in}(i)$ como o conjunto de todos os dados de entrada da ativação $i \in N$, e $\Delta_{out}(i)$ como o conjunto de dados de saída gerados pela ativação. Ademais, t_{max} é definido como o tempo máximo esperado para a execução do *workflow*. Dessa forma, $T = 1...t_{max}$ representa o conjunto de períodos de

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

tempo viáveis. Cada *bucket* $j \in B$ é dividido em intervalos de armazenamento $\{0...|L_j|\}$, cada um com a capacidade sm_{jl}, indicando a faixa de volume utilizado (o intervalo 0 indica que o *bucket* não está em uso). Assim, c^B_{jl} é o custo de contratação do *bucket* $j \in B$ no intervalo $l \in L_j$, e c^M_j é o custo financeiro de contratar uma MV $j \in M$ por um período de tempo (US$/minuto). O **grafo de conflitos** é definido como $G_c = (D, E_h \cup E_s, \delta)$ onde as arestas $(E_h \cup E_s)$ representam os conflitos de confidencialidade. Definimos dois tipos de conflitos: *hard* (E_h) e *soft* (E_s). Os conflitos *hard* definem pares de dados que não podem ser armazenados num mesmo dispositivo, enquanto que os conflitos *soft* definem pares de dados que não deveriam estar juntos mas podem estar perante o pagamento de uma penalidade δ.

As Tabelas *workflow*, *segurança e privacidade*, e *variáveis* resumem os parâmetros e variáveis utilizados na formulação matemática proposta para `CYCLOPS-IP`. A abordagem começa pela definição da função objetivo, que visa minimizar 3 métricas: o tempo de execução do *workflow* (1), o custo financeiro (2), e a exposição à segurança e violação de privacidade (3). Os pesos α_t, α_b e α_s são utilizados para determinar a importância de cada objetivo, e idealmente, sua soma deve ser igual a 1, permitindo um ajuste fino do modelo. Isso possibilita aos usuários escolher se desejam uma execução rápida, de baixo custo ou segura. É relevante observar que todos os 3 objetivos são normalizados usando t_{max}, c_{max} e s_{max}, respectivamente. As restrições (4) asseguram que cada ativação seja executada. Já as restrições (5) e (6) estipulam que todas as operações de leitura e escrita, respectivamente, sejam realizadas.

$$\min \quad \alpha_t \cdot \left(\frac{z^T}{t_{max}}\right) + \tag{1}$$

$$\alpha_b \cdot \left(\sum_{j \in M} \frac{c^M_j z^T_j}{c_{max}} + \sum_{j \in B} \sum_{l \in L_j} \frac{c^B_{jl} q_{jl}}{c_{max}}\right) + \tag{2}$$

$$\alpha_s \cdot \left(\sum_{r \in R} \sum_{i \in N} \frac{e_{ri}}{s_{max}} + \sum_{(d_1,d_2) \in E_s} \frac{\delta_{d_1 d_2} w_{d_1 d_2}}{s_{max}}\right) \tag{3}$$

$$\sum_{j \in M} \sum_{t \in T} x_{ijt} = 1, \qquad \forall i \in N \tag{4}$$

$$\sum_{\substack{j \in M \\ p \in \overline{M}}} \sum_{t \in T} \overrightarrow{x}_{idjpt} = 1, \qquad \forall i \in N, \forall d \in \Delta_{in}(i) \tag{5}$$

$$\sum_{\substack{j \in M \\ p \in \overline{M}}} \sum_{t \in T} \overleftarrow{x}_{idjpt} = 1, \qquad \forall i \in N, \forall d \in \Delta_{out}(i) \tag{6}$$

As desigualdades (7) garantem que os dados $d \in \Delta_{out}(i)$ só podem ser escritos se a ativação i for executada no momento correto. Adicionalmente, as restrições (8) determinam que os dados d não podem ser escritos antes do tempo de processamento da ativação i, que é responsável por sua escrita. É importante observar que ambos os conjuntos de restrições ((7) e (8)) operam em conjunto para garantir um tempo viável para o processo de escrita.

$$\overleftarrow{x}_{idjpt} \leq \sum_{q=1}^{t-t_{ij}} x_{ijq}, \qquad \forall i \in N, \forall d \in \Delta_{out}(i), \forall j \in M, \forall p \in \overline{M}, \forall t = (t_{ij}+1) \cdots T_M \tag{7}$$

$$\overleftarrow{x}_{idjpt} = 0, \qquad \forall i \in N, \forall d \in \Delta_{out}(i), \forall j \in M, \forall d \in \Delta_{out}(i), 1 \leq t \leq t_{ij} \tag{8}$$

As restrições (9) determinam que uma ativação só pode ser executada se todas as leituras necessárias forem concluídas em um tempo viável.

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

$$x_{ijt} \leq \sum_{p \in \overline{M}} \sum_{q=1}^{t-\overrightarrow{t}_{djp}} \overrightarrow{x}_{idjpq}, \qquad \forall i \in N, \forall d \in \Delta_{in}(i), \forall j \in M, \forall t \in T, \text{ such } (t - \overrightarrow{t}_{djp}) \geq 1 \quad (9)$$

As desigualdades (10) garantem que apenas no máximo uma ação (execução, leitura ou escrita) possa ocorrer em um período de uma MV (*e.g.,* uma MV não pode executar uma ativação e escrever dados ao mesmo tempo). Por outro lado, as desigualdades (11) determinam que ações passivas (um dado ser lido ou escrito em uma máquina) podem ser realizadas em paralelo. É importante notar que qualquer ação (ativa ou passiva) realizada em uma MV j no período t terá a variável v_{jt} definida como 1, indicando que a MV está em uso.

$$\sum_{i \in N} \sum_{q=\max(1, t-t_{ij}+1)}^{t} x_{ijq} + \sum_{i \in N} \sum_{d \in \Delta_{out}(i)} \sum_{p \in \overline{M}} \sum_{r=\max(1, t-\overleftarrow{t}_{djp}+1)}^{t} \overleftarrow{x}_{idjpr} + \sum_{i \in N} \sum_{d \in \Delta_{in}(i)} \sum_{p \in \overline{M}} \sum_{r=\max(1, t-\overrightarrow{t}_{djp}+1)}^{t} \overrightarrow{x}_{idjpr} \leq v_{jt}, \qquad \forall j \in M, \forall t \in T \quad (10)$$

$$\sum_{i \in N} \sum_{d \in \Delta_{out}(i)} \sum_{p \in M} \sum_{r=\max(1, t-\overleftarrow{t}_{dpj}+1)}^{t} \overleftarrow{x}_{idpjr} + \sum_{i \in N} \sum_{d \in \Delta_{in}(i)} \sum_{p \in M} \sum_{r=\max(1, t-\overrightarrow{t}_{dpj}+1)}^{t} \overrightarrow{x}_{idpjr} \leq |M|.v_{jt}, \qquad \forall j \in M, \forall t \in T \quad (11)$$

As restrições (12) asseguram que não haja dados dinâmicos no momento inicial da execução do *workflow*. Por outro lado, as restrições (13) e (14) garantem que todos os dados estáticos estejam pré-armazenados em seus dispositivos de origem (*i.e.*, uma MV ou um *bucket*).

$$y_{dj1} = 0, \qquad \forall d \in D_d, \forall j \in \overline{M} \quad (12)$$

$$y_{dj1} = 0, \qquad \forall d \in D_s \mid j \in (\overline{M} \backslash O(d)) \quad (13)$$

$$y_{djt} = 1, \qquad \forall d \in D_s \mid j \in O(d), \forall t \in T \quad (14)$$

As restrições (15) e (16) vinculam a variável de armazenamento y com as variáveis de escrita $\overleftarrow{x}$ e de leitura $\overrightarrow{x}$, garantindo um processo de escrita e leitura viável, respectivamente. Em detalhes, a restrição (15) garante que os dados só estarão armazenados em um dispositivo se já tiverem sido produzidos (escritos) anteriormente. Por outro lado, as restrições (16) garantem que os dados só serão lidos se estiverem previamente armazenados em um dispositivo.

$$y_{dp(t+1)} = y_{dpt} + \sum_{j \in M} \overleftarrow{x}_{idjp(t-\overleftarrow{t}_{djp}+1)}, \qquad \forall d \in D, \forall p \in \overline{M}, \forall t \in \{1 \ldots t_{max} - 1\}, \text{ such } (t - \overleftarrow{t}_{djp} + 1) \geq 1, d \in \Delta_{out}(i) \quad (15)$$

$$\sum_{j \in M} \overrightarrow{x}_{idjpt} \leq y_{dpt}, \qquad \forall i \in N, \forall d \in \Delta_{in}(i), \forall p \in \overline{M}, \forall t \in T \quad (16)$$

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

As capacidades de armazenamento dos dispositivos (tanto *buckets* quanto discos de MV) são limitadas pelas restrições (17). As restrições (18) relacionam a última operação de escrita com o tempo de execução do *workflow* (*makespan*). Note que, em nosso modelo de aplicação, uma ativação sempre produz dados. As restrições (19) vinculam as variáveis v_{jt} e z_j^T para estabelecer o tempo de alocação (pago) correto da MV j, enquanto as restrições (20) garantem que os custos financeiros não excedam o orçamento máximo definido pelos usuários.

$$\sum_{d \in D} y_{djt} W(d) \leq cm_j, \qquad \forall j \in \overline{M}, \forall t \in T \quad (17)$$

$$\overleftarrow{x}_{idjpt} \cdot (t + \overleftarrow{t}_{djp}) \leq z^T, \qquad \forall i \in N, \forall d \in \Delta_{out}(i), \forall j \in M, \forall p \in \overline{M}, \forall t \in T \quad (18)$$

$$v_{jt} \cdot t \leq z_j^T, \qquad \forall j \in M, \forall t \in T \quad (19)$$

$$\sum_{j \in M} c_j^M z_j^T + \sum_{j \in B} \sum_{l \in L_j} c_{jl}^B q_{jl} \leq c_{max} \quad (20)$$

Além disso, a seguinte restrição operacional (21) deve ser satisfeita: uma ativação i só pode iniciar qualquer processo de leitura se todos os dados $d \in \Delta_{in}(i)$ já estiverem disponíveis (*i.e.*, se todos os dados $d \in (\Delta_{in}(i) \cap D_d)$ estiverem escritos).

$$\sum_{p \in \overline{M}} \overrightarrow{x}_{idjpt} \cdot |\Delta_{in}(i)| \leq \sum_{g \in \Delta_{in}(i)} \sum_{p \in \overline{M}} y_{gpt}, \qquad \forall i \in N, \forall d \in \Delta_{in}(i), \forall j \in M, \forall t \in T \quad (21)$$

As restrições (22) relacionam a variável de armazenamento y (associada ao período de tempo) com a variável de armazenamento $\overline{y}$, que é independente do tempo. A restrição (23) garante que dois dados conflitantes não sejam colocados no mesmo dispositivo (conflito *hard*). Da mesma forma, as desigualdades (24) garantem que se dois dados conflitantes forem armazenados no mesmo dispositivo, a variável de penalidade w equivalente seja atribuída ao valor 1 (conflito *soft*). As restrições (25) medem o nível de exposição, ou seja, o risco de se executar uma ativação ao considerar os requisitos de segurança.

$$y_{djt} \leq \overline{y}_{dj} \quad , \forall d \in D, \forall j \in \overline{M}, \forall t \in T \quad (22)$$

$$\overline{y}_{d_1 j} + \overline{y}_{d_2 j} \leq 1 \quad , \forall (d_1, d_2) \in E_h, \forall j \in \overline{M} \quad (23)$$

$$\overline{y}_{d_1 j} + \overline{y}_{d_2 j} \leq 1 + w_{d_1 d_2} \quad , \forall (d_1, d_2) \in E_s, \forall j \in \overline{M} \quad (24)$$

$$l_{task}^{ri} - \sum_{j \in M} \sum_{t \in T} l_{vm}^{rj} x_{ijt} \leq e_{ri} \quad , \forall i \in N, \forall r \in R \quad (25)$$

$$\sum_{l \in L_j} q_{jl} = \sum_{d \in D} \overline{y}_{dj} W(d) \quad , \forall j \in B \quad (26)$$

$$\sum_{l \in L_j} b_{jl} = 1 \quad , \forall j \in B \quad (27)$$

$$\sum_{d \in D} \overline{y}_{dj} W(d) \leq \sum_{l \in L_j} sm_{jl} b_{jl} \quad , \forall j \in B \quad (28)$$

$$sm_{j(l-1)}.b_{jl} \leq q_{jl} \leq sm_{jl}.b_{jl} \quad , \forall j \in B, \forall l \in L_j \backslash \{0\} \quad (29)$$

$$b_{jl} \leq \sum_{d \in D} \overline{y}_{dj} \quad , \forall j \in B, \forall l \in L_j \backslash \{0\} \quad (30)$$

A equação (26) calcula o volume dos dados armazenados em um mesmo *bucket*. As restrições (27) determinam os intervalos de armazenamentos $l \in L_j$ de cada *bucket* $j \in B$, enquanto que as restrições (28) restringem os tamanhos de armazenamentos utilizados no *buckets*. Por sua

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

vez, as restrições (27) e (29) garantem o uso adequado dos intervalos de armazenamento (b_{jl}) de acordo com o volume total dos dados no *bucket* (q_{jl}). Por fim, as restrições (30) impõem que, se um intervalo $l \in L_j \backslash \{0\}$ do *bucket* $j \in B$ estiver ativo, deve existir pelo menos algum dado associado ao *bucket* j.

4. Heurística Proposta

A execução de *workflows* com restrições de confidencialidade, conforme a abordagem CYCLOPS descrita na Seção 3, pode ser planejada usando métodos exatos apenas para *workflows* de pequena escala, geralmente com menos de 15 ativações. Para *workflows* maiores, como os encontrados no mundo real, métodos exatos tornam-se impraticáveis devido ao alto consumo de recursos computacionais e ao tempo de processamento. Em contrapartida, heurísticas e metaheurísticas oferecem alternativas eficazes e viáveis para resolver problemas complexos de otimização. Nesta seção, apresentamos uma heurística construtiva gulosa-aleatória denominada CYCLOPS-GRCH, projetada para escalonar as ativações de *workflows* com restrições de confidencialidade.

Algorithm 1: CYCLOPS-GRCH

```
   Data: θ_RCL, β
   Result: schedule S
1  Record {
2      integer act_index, vm_index;
3      List_of_Indexes res;
4      float of_value;
5  } CandList;
6  S ← ∅; N̄ ← N;
7  while N̄ ≠ ∅ do
8      CandList_temp ← new(CandList); CandList ← new(CandList); CandList_restricted ←
         new(CandList);
9      foreach i ∈ N̄ do
10         foreach j ∈ M do
11             if Gen(Δ_in(i)) then
12                 OF_value ← Calc_OF(S, i, j); W ← EstRes(S, i, j, β) ;
13                 CandList_temp.act_index ← i; CandList_temp.vm_index ← j;
                     CandList_temp.res ← W; CandList_temp.of_value ← OF_value;
14                 CandList ← CandList ∪ CandList_temp;
15             end
16         end
17     end
18     sort(CandList);
19     CandList_restricted ← GenRCL(CandList, θ_RCL);
20     (i*, j*, W*) ← draw(CandList_restricted);
21     S ← S ∪ (i*, j*, W*);
22     N̄ ← N̄\{i*};
23 end
```

O Algoritmo 1 apresenta o procedimento CYCLOPS-GRCH. Este algoritmo garante uma solução viável respeitando a precedência entre as ativações, ou seja, uma ativação só é escalonada quando os seus dados de entrada estão disponíveis. O *loop* mais externo (linha 7) itera até que todas as ativações do *workflow* sejam executadas em alguma MV. O algoritmo verifica se os dados necessários estão disponíveis para executar a ativação $i \in \overline{N}$ numa MV $j \in M$, usando a função $Gen()$ (linha 11). Em seguida, calcula o custo de escalonar esta atividade na MV, utilizando a função $Calc_{OF}$. Este custo é definido pelas Equações (1 - 3). Adicionalmente, a heurística estima os dispositivos $W \subseteq \overline{M}$ que irão armazenar os dados de saída da ativação i (função $EstRes()$) da seguinte maneira: para todo dado $d \in \Delta_{out}(i)$, o método seleciona aleatoriamente β dispositivos de $\overline{M}$ e, dentre eles, seleciona aquele com o menor custo na função objetivo que não viole restrições de capacidade e confidencialidade. O algoritmo então gera uma lista de atribuições atividade/MV

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

candidatas ($CandList$) (linha 14), ordenando-as (de forma crescente) com base no valor of_{value} da função objetivo (linha 18) e cria uma Lista de Candidatos Restrita ($CandList_{restricted}$), usando o parâmetro θ_{RCL}, contendo os melhores candidatos (linha 19). Um candidato é escolhido aleatoriamente da $CandList_{restricted}$ e adicionado à solução corrente S (linhas 20 e 21). Este processo é repetido até que todas as ativações do *workflow* estejam escalonadas para execução. Devido à natureza estocástica do `CYCLOPS-GRCH`, diferentes soluções são encontradas para os mesmos dados de entrada, o que justifica a execução do algoritmo 100 vezes para selecionar o melhor resultado entre todas as execuções.

5. Avaliação Experimental Preliminar

Para testar a capacidade da abordagem, realizamos dois conjuntos de testes: (i) uma comparação entre a heurística `CYCLOPS-GRCH` e o método exato usando 6 pequenas instancias sintéticas, e (ii) uma avaliação do desempenho da heurística usando 3 *workflows* do mundo real. A Tabela 1 apresenta as especificações das instâncias utilizadas, onde, #Act. indica o número de ativações e #Arq. o número de arquivos (dados). As especificações de t_{max} e c_{max}, foram definidas empiricamente de forma garantir que ultrapassem os maiores *makespans* e custos financeiros encontrados em testes preliminares. Nesses experimentos, os **grafos de conflitos** foram gerados por um *script* que avaliou os *workflows* e determinou as restrições para os dados usando as seguintes regras: arquivos de entrada e saída de uma mesma ativação não podem ser armazenados juntos (restrição *hard*) (i); arquivos de saída de ativações irmãs (mesmo nível no grafo) não devem ser armazenados juntos (restrição *soft*), e caso isso ocorra, o valor 1 de penalidade é aplicado para cada conflito entre 2 arquivos (ii). Ambos os métodos (heurístico e exato) foram implementados usando GCC 11.4.0 e IBM ILOG CPLEX 22.1.1.0, e executados em um PC Intel i7-10870H 4.80 GHz com 16GB RAM.

Instância Sintética	**#Act.**	**#Arq.**	t_{max}	c_{max}	**Instância Real**	**#Act.**	**#Arq.**	t_{max}	c_{max}
Sintético_005_A	2	3	50	120	Montage_025	25	54	250	125
Sintético_005_B	2	3	100	120	Montage_050	50	107	500	250
Sintético_005_C	1	4	200	80	Montage_100	100	215	1.000	500
Sintético_010_A	4	6	170	168					
Sintético_010_B	3	7	75	198					
Sintético_010_C	4	6	300	440					

Tabela 1: Características das instâncias de *workflows* sintéticas e reais

5.1. Instancias Sintéticas

Considerando a impraticabilidade da solução exata para instâncias reais, foram criadas 6 pequenas instâncias sintéticas de *workflows*, permitindo assim a comparação da heurística com o método exato. Neste primeiro teste, para os parâmetros de entrada, optou-se pelo balanceamento dos objetivos, usando $\alpha_t = 0.3$, $\alpha_b = 0.3$ e $\alpha_s = 0.4$, com uma pequena prevalência para a confidencialidade dos dados. Para os parâmetros da heurística, testes empíricos foram realizados e os valores $\theta_{LRC} = 0.5$ e $\beta = 4$ foram obtidos. Como configuração do cenário de dispositivos para este experimento, foram empregadas 4 MVs, com base na Tabela 2, uma de cada tipo disponível e todas com capacidade de criptografia de nível 1. Além disso, foram utilizados 2 *buckets* como dispositivos unicamente de armazenamento. Os valores *Slowdown* e Rede (Gbps) na Tabela 2 são utilizados para calcular os tempos de execução, escrita e leitura de cada par ativação/máquina como descritos em Teylo et al. [2017].

A Tabela 3 mostra os resultados para as instancias sintéticas, onde F.O. representa a Função Objetivo, *Mks* o *Makespan* em segundos, US$ o custo em dólares, *Conf.* a penalidade

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

Tipo	Disco (GB)	*Slowdown*	Rede (Gbps)	Custo (US$)	Criptografia
MV1	80	1.53	4	0.02	1 ou 0
MV2	120	0.77	9	0.09	1 ou 0
MV3	160	0.38	10	0.16	1 ou 0
MV4	200	0.19	10	0.33	1 ou 0
Bucket	51200	-	25	0.023	-

Tabela 2: Características das máquinas virtuais e dos *Buckets* utilizados na nuvem

	Heurística (CYCLOPS-GCH)					**Solução Exata**				
Workflow	F. O.	*Mks* (s)	US$	Conf.	T (s)	F. O.	*Mks* (s)	US$	Conf.	T (s)
Sintético_005_A	0,110	10,00	$< 0,01$	0,13	$< 0,01$	0,108	18,00	$< 0,01$	0,00	35,27
Sintético_005_B	0,140	17,00	$< 0,01$	0,22	$< 0,01$	0,140	17,00	$< 0,01$	0,22	568,45
Sintético_005_C	0,107	27,00	$< 0,01$	0,17	$< 0,01$	0,107	27,00	$< 0,01$	0,17	73,16
Sintético_010_A	0,097	40,00	$< 0,01$	0,07	$< 0,01$	-	-	-	-	3.772,37
Sintético_010_B	0,156	19,00	$< 0,01$	0,20	$< 0,01$	0,156	19,00	< 0.01	0.20	805,87
Sintético_010_C	0,126	59,00	$< 0,01$	0,17	$< 0,01$	-	-	-	-	3.870,39

Tabela 3: Avaliação Experimental - **CYCLOPS-GCH** *Versus* **Solução Exata**

de confidencialidade dos dados e T o tempo em segundos que o método levou para gerar a solução. Vale destacar que a coluna *Conf.* foi normalizada para facilitar a compreensão dos dados. O modelo matemático obteve soluções garantidamente ótimas para a maioria dos *workflows* sintéticos, excetuando a última e antepenúltima, devido ao tempo limite de 1 hora imposto ao CPLEX. Enquanto isso, a heurística produziu soluções satisfatórias, desempenhando 0,004% a mais de função objetivo na média em relação ao método exato. A diferença de desempenho entre as abordagem se devem, possivelmente, devido à execução atômica das ativações (leitura, processamento e escrita) da heurística em comparação com o método exato. Este último permite que leituras e escritas de dados ocorram desacopladas do processamento, otimizando assim as alocações das MVs. Porém, as conclusões mais importantes são os tempos de execuções do CYCLOPS-GRCH, que foram significativamente menores do que os do CPLEX, e os valores de confidencialidade baixos, demonstrando assim a eficácia da abordagem proposta.

5.2. Instâncias do Mundo Real

O CYCLOPS-GRCH foi aplicado ao *workflow* do mundo real denominado Montage [Juve et al., 2013], utilizado para criar imagens mosaico a partir de fotografias astronômicas. Foram consideradas 3 variantes do Montage, onde cada uma possui diferentes quantidades de ativações. Neste experimento, vários cenários foram criados ajustando-se os pesos da função objetivo, são eles: Cenário 1 - Ênfase na redução do *makespan* - $\alpha_t = 1$, $\alpha_b = 0$, $\alpha_s = 0$ (i); Cenário 2 - Ênfase na redução dos custos financeiros - $\alpha_t = 0$, $\alpha_b = 1$, $\alpha_s = 0$ (ii); Cenário 3 - Ênfase na redução das penalidades de confidencialidade - $\alpha_t = 0$, $\alpha_b = 0$, $\alpha_s = 1$ (iii); Cenário 4 - Objetivos balanceados - $\alpha_t = 0.33$, $\alpha_b = 0.33$, $\alpha_s = 0.33$ (iv). Com base na Tabela 2, foram empregados 1 *site* com 4 MVs, uma de cada tipo, sendo as duas primeiras com capacidade de criptografia nível 1 e as demais sem criptografia (nível 0). Como todas as ativações foram definidas como tendo necessidade por criptografia, uma penalidade de 1 foi atribuída para cada ativação associada às MVs sem criptografia. Além disso, foram utilizados 16 *buckets* como dispositivos de apenas armazenamento.

Diferentemente do método exato, a heurística conseguiu encontrar soluções viáveis para todas as instâncias de *workflows* do mundo real em menos de 20 segundos. Optou-se por usar

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

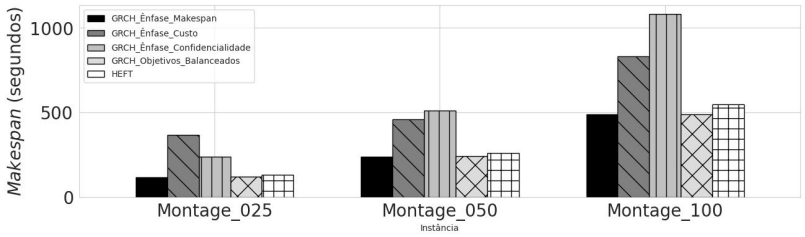

Figura 1: GRCH Vs HEFT (Makespan)

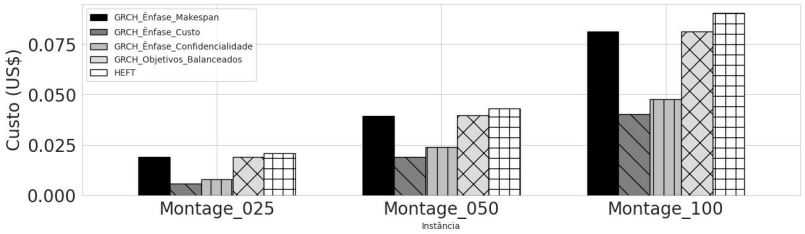

Figura 2: GRCH Vs HEFT (Custo)

o HEFT [Topcuoglu et al., 2002] como referência para comparar os resultados obtidos com o CYCLOPS-GRCH. Embora o HEFT não aborde confidencialidade, é amplamente utilizado como algoritmo de escalonamento devido à sua capacidade de gerar soluções eficazes para *workflows* complexos em ambientes heterogêneos. A Figura 1 apresenta o *makespan* das heurísticas. Pode-se observar que o método CYCLOPS-GRCH atinge melhores resultados que o HEFT quando o *makespan* foi priorizado ou quando os objetivos estão balanceados. Como esperado, ao priorizar custo ou confidencialidade, o *makespan* aumentou. A Figura 2 apresenta o *custo* das heurísticas. Pode-se observar que em todos os casos a CYCLOPS-GRCH obteve melhores resultados em relação ao HEFT, uma vez que esta heurística foca exclusivamente no *makespan*. A Figura 3 apresenta as penalidades de confidencialidade das heurísticas. Observa-se que em todos os casos a CYCLOPS-GRCH obteve melhores resultados em relação ao HEFT. Vale notar que a ênfase dada na confidencialidade foi capaz de zerar as penalidades em todas as instâncias deste experimento. De modo geral, a ênfase no *makespan* teve resultados bem parecidos que os dos objetivos balanceados, o motivo para isso se dá pela maior variabilidade dos valores de *makespan* das soluções, portanto, esta característica acaba se sobressaindo quando igualamos a importância dos objetivos.

6. Conclusões

Muitas aplicações usam *workflows* de *big data*, onde dados sensíveis são gerados e armazenados em *buckets* na nuvem, aumentando o risco de comprometer a confidencialidade. Este artigo aborda a execução eficiente de *workflows* com restrições de confidencialidade, propondo abordagens exatas e heurísticas, que integram o escalonamento das ativações com as restrições de confidencialidade e o plano de dispersão de dados. Avaliamos nossa abordagem usando instâncias de *workflows* sintéticos e reais, obtendo resultados promissores para garantir a confidencialidade e minimizar custos. Como próximo passo, planejamos evoluir a heurística proposta com a inclusão de métodos de perturbações e buscas locais, abordando vários outros *workflows* do mundo real.

Referências

Abazari, F., Analoui, M., Takabi, H., e Fu, S. (2019). Mows: multi-objective workflow scheduling in cloud computing based on heuristic algorithm. *Simulation Modelling Practice and Theory*, 93: 119–132.

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br

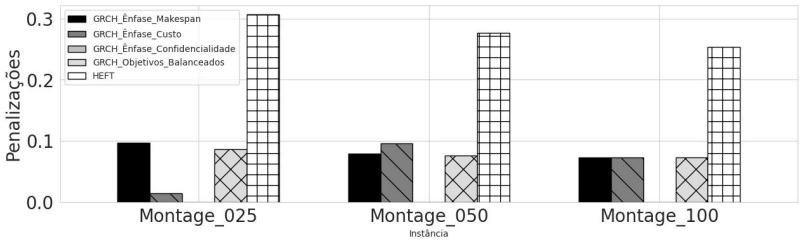

Figura 3: GRCH Vs HEFT (Confidencialidade)

de Oliveira, D., Liu, J., e Pacitti, E. (2019). *Data-Intensive Workflow Management: For Clouds and Data-Intensive and Scalable Computing Environments*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

Guerine, M., Stockinger, M. B., et al. (2019). A provenance-based heuristic for preserving results confidentiality in cloud-based scientific workflows. *Fut. Gen. Comp. Sys.*, 97:697 – 713.

Juve, G., Chervenak, A. L., Deelman, E., Bharathi, S., Mehta, G., e Vahi, K. (2013). Characterizing and profiling scientific workflows. *FGCS*, 29(3):682–692.

Ristenpart, T., Tromer, E., Shacham, H., e Savage, S. (2009). Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. In *Proc. of the CCS '09*, p. 199–212.

Sharif, S., Watson, P., et al. (2016). Privacy-aware scheduling saas in high performance computing environments. *IEEE TPDS.*, 28(4):1176–1188.

Shishido, H., Estrella, J. C., et al. (2018). Multi-objective optimization for workflow scheduling under task selection policies in clouds. In *CEC*, p. 1–8.

Sujana, J. A. J., Revathi, T., Priya, T. S., e Muneeswaran, K. (2019). Smart pso-based secured scheduling approaches for scientific workflows in cloud computing. *Soft. Comp.*, 23(5):1745–1765.

Tawfeek, M. A. e AbdulHamed, A. A. (2018). Service flow management with multi-objective constraints in heterogeneous computing. In *ICCES*, p. 258–263.

Teylo, L., de Paula Junior, U., et al. (2017). A hybrid evolutionary algorithm for task scheduling and data assignment of data-intensive scientific workflows on clouds. *FGCS*, 76:1–17.

Topcuoglu, H., Hariri, S., e Wu, M. (2002). Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE TPDS*, 13(3):260–274.

Wen, Y., Liu, J., et al. (2020). Scheduling workflows with privacy protection constraints for big data applications on cloud. *FGCS*, 108:1084–1091.

Zamfiroiu, A., Petre, I., e Boncea, R. (2019). Cloud computing vulnerabilities analysis. In *Proc. of the CCIOT '19*, p. 48–53.

Zhou, A. C., Xiao, Y., Gong, Y., He, B., Zhai, J., e Mao, R. (2019). Privacy regulation aware process mapping in geo-distributed cloud data centers. *IEEE TPDS*, 30(8):1872–1888.

DOI: 10.59254/sbpo-2024-193619
https://proceedings.science/p/193619?lang=pt-br