

## ADAPTAÇÃO DO ESCALONADOR KAIROS PARA ANALISAR REQUISITOS DE COMUNICAÇÃO<sup>1</sup>

Paulo Roberto Albuquerque<sup>2</sup>, Guilherme Piêgas Koslovski<sup>3</sup>.

<sup>1</sup> Vinculado ao projeto “Mecanismos para Alocação de Infraestruturas Virtuais baseados em Aprendizado de Máquina e Acelerados por GPUs”

<sup>2</sup> Acadêmico (a) do Curso de Ciência da Computação – CCT – Bolsista PROBIC

<sup>3</sup> Orientador, Departamento de Ciência da Computação – CCT – guilherme.koslovski@udesc.br

Atribui-se o nome escalonamento ao procedimento de organização de processos, tarefas e/ou trabalhos em sistemas operacionais ou sistemas distribuídos, de tal maneira que se otimize o uso de recursos computacionais tradicionais como CPU e RAM, recursos de rede, e métricas de tempo de execução, latência, etc. Porém, recursos de rede são desconsiderados ou não recebem a devida importância, mesmo considerando que uma significativa porção do tempo de processamento gasto em Data Centers é atribuído ao tráfego de rede. Com este problema em mente, um framework de escalonamento de tarefas chamado Kairos foi previamente proposto e desenvolvido, pretendendo utilizar tecnologias de aprendizado de máquina para tentar alcançar um desempenho satisfatório de escalonamento, considerando tarefas comunicantes.

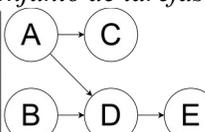
Porém, devido à profunda complexidade do problema, que comumente é chamado de Job-Shop Problem, estes objetivos não conseguiram ser cumpridos totalmente. Com isso em mente, teve-se como objetivo a adaptação desta framework previamente desenvolvida, a fim de acrescentar o processamento de requisitos de comunicação ao escalonador, e aperfeiçoar alguns aspectos da implementação, com o propósito de atingir um desempenho no mesmo nível, ou superior a escalonadores e algoritmos de escalonamento tradicionais e populares usados de fato em Data Centers da indústria.

Na Figura 1 é possível observar um conjunto de tarefas, cada uma com um determinado custo de tempo, e um grafo acíclico direcionado indicando as relações de dependências entre elas. Já na Figura 2, pode-se observar dois possíveis cenários de escalonamento diferentes que podem ser ordenados para as tarefas apresentadas na Figura 1. É válido notar que mesmo os dois cenários seguirem as mesmas restrições, um tem um tempo total de execução 40% maior que o outro pelo fato de organizar de forma mais otimizada a ordem de execução das tarefas, e aproveitar de um processamento paralelizado.

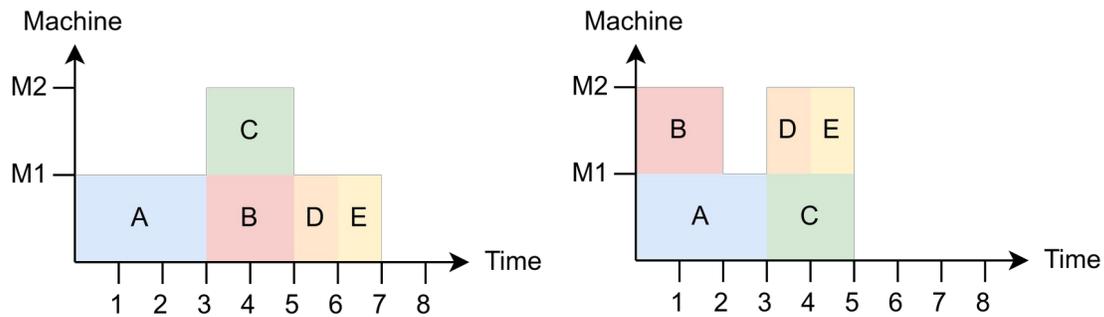
Na Figura 3, esta representado uma visão macro de todo o fluxo de dados da Framework Kairos, desde o pré-processamento do *dataset* de origem, o treinamento da rede neural, até o recolhimento e posterior análise dos resultados coletados ao comparar o algoritmo de aprendizado profundo gerado, com os algoritmos tutores.

**Figura 1.** Custos e relações de dependências de um conjunto de tarefas. (Fonte: Autor)

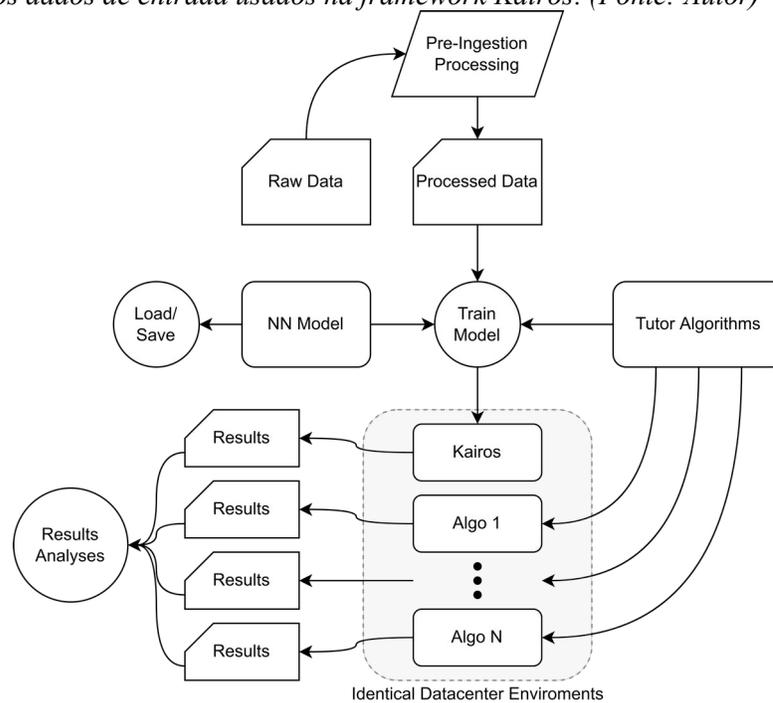
Task	A	B	C	D	E
Time Cost	3	2	2	1	1



**Figura 2.** Dois cenários de escalonamento do conjunto de tarefas observado na Fig 1. (Fonte: Autor)



**Figura 3.** Fluxo dos dados de entrada usados na framework Kairos. (Fonte: Autor)



**Palavras-chave:** Escalonamento. Data Center. Job-Shop.