



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Exploratory data analysis
Metodologias Experimentais em Informática

Bruno Sequeira uc2020235721@student.uc.pt

Samuel Machado uc2020219391@student.uc.pt

Tomás Dias uc2020215701@student.uc.pt

PL3

Ano letivo 2023/24

Índice

| | |
|---------------------------------|-----------|
| Índice | 2 |
| Introdução | 3 |
| Análise | 4 |
| HPC2N | 4 |
| NASA | 16 |
| SDSC | 28 |
| Distribuição de Trabalho | 34 |

Introdução

Neste relatório, descrevemos a nossa primeira interação com o processo de Análise Exploratória de Dados, isto é a aplicação sistemática de técnicas estatísticas e lógicas a determinados datasets. O objetivo principal deste processo é extrair informações valiosas, condensando as principais características e permitindo, acima de tudo, uma visualização eficaz destas informações que esses dados nos dão. Além disso, a AED faz uso de métodos visuais diversos, tais como gráficos e tabelas.

Naturalmente, como primeiro passo deste projeto, realizamos uma reunião para selecionar alguns conjuntos de dados adequados. Durante essa seleção, priorizamos conjuntos de dados limpos, ou seja, aqueles cujas informações estavam mais próximas de estar prontas para uma análise confiável. Além disso, buscamos conjuntos de dados heterogêneos, que nos permitissem formular uma variedade de questões. Optamos por quatro conjuntos de dados, mas posteriormente identificamos que uma análise sólida dos três seguintes seria eficaz:

- HPC2N.csv
- NASA.csv
- SDSC96.csv

Após definirmos os datasets ainda foi necessário manipular um pouco os ficheiros com recurso ao python, de forma a estarem aptos a ser inseridos e analisados no Datatab, a ferramenta que optámos por utilizar. Este relatório apresenta questões que procuramos responder, seguidas das conclusões que extraímos desta análise de dados. Em resumo, nosso foco principal é obter uma visão aprofundada dos dados e, assim, contribuir para uma compreensão mais ampla dos sistemas em questão.

Análise

HPC2N

Depois de preparar os dados usando Python para os colocar no DATAtab, descobrimos que alguns dos Data Fields não seriam adequados para tirar conclusões neste conjunto de dados (Status; Executable (Application) Number; Queue Number; Partition Number; Preceding Job Number; Think Time from Preceding Job).

Isso significa que temos apenas 12 campos de dados que são relevantes para nossas análises. Um dos 12 campos continha informações temporais (Run Time), mas não estava disponível de forma a permitir uma análise mais detalhada. Por forma a conseguirmos abordar essa questão, introduzimos novos campos: o dia da semana, o dia e o mês.

Com base nas informações fornecidas sobre o conjunto de dados sabemos que os dados são provenientes do Seth, um cluster Linux com 120-nós e que cada nó possui 1 GB de RAM sendo esta partilhada por dois processadores. O Seth usa o sistema de agendamento Maui, uma ferramenta de programação e gestão de recursos em cluster, principalmente utilizada em ambientes de computação de alto desempenho (HPC). O Maui permite que os administradores definam políticas de programação com base em fatores como prioridades de trabalhos, tempos de execução de trabalhos, requisitos de recursos e muito mais. Os trabalhos recebem valores de prioridade e o programador aloca recursos com base nessas prioridades. Normalmente, os trabalhos de maior prioridade são programados primeiro.

Tendo como ponto de partida estas informações propusemo-nos a analisar as seguintes questões:

Todos os trabalhos foram concluídos com sucesso?

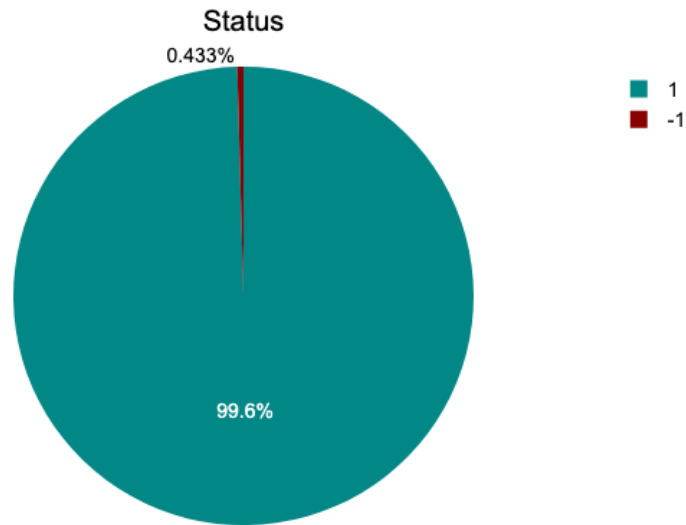


Fig. 1 - Percentagem de trabalhos concluídos com e sem sucesso

Para o field status existem 4 tipos de resultados possíveis: 1 (Concluído), 0 (não correu), 5 (parado) e por último -1. Este último caso, que representa uma reduzida percentagem (0,43%) menos de 1% do total dos casos, foi referido na documentação. Este existe devido ao facto de estes trabalhos ainda no ficheiro inicial estarem num estado 'Runnable', e por isso mesmo não estão com status igual a 1 porque não foram dados por concluídos.

Este gráfico evidencia uma boa performance do sistema tendo 99,6% dos trabalhos terem sido completados, não tendo havido trabalhos por correr ou mesmo parados durante a sua execução.

Quantas partições foram usadas neste sistema?

| Partition Number | % |
|------------------|---------------|
| 1 | 100% |
| 2 | 0%(0.000493%) |
| Total | 100% |

Tabela de distribuição de partições

Com esta tabela, concluímos que o trabalho foi realizado em 2 partições, maioritariamente na partição 1. É visível que a partição 2 praticamente não foi usada e, devido a isso mesmo, surgiu o interesse em entender esta questão.

O que explica existir um número reduzido de dados sobre a partição 2?

| Partition Number 2 | |
|---------------------------|------------|
| User ID | 12 |
| Request Memory | 409,6k |
| Request Time | 1800 ms |
| Req. Number of Processors | 2 |
| Run Time | 9 ms |
| Wait Time | 6351 ms |
| Data | 26/03/2003 |

[Tabela das especificações da partição 2](#)

Analisando todos os dados recolhidos e apresentados na tabela acima podemos verificar que o único user que usou a partição 2 foi o número 12. Tendo inserido todos os trabalhos no mesmo dia, todos com os mesmos requisitos (Memória, Tempo e Número de processadores). Apesar de não existir informação neste dataset acerca da aplicação, com as informações anteriores podemos supor que a aplicação seja a mesma. Pelas informações recolhidas acima estes trabalhos estavam a correr e apenas correram 9 ms. Em suma, esta partição foi usada sempre pelo mesmo user num único dia, em 26 de Março de 2003.

Qual foi a distribuição do trabalho dentro da partição 1?

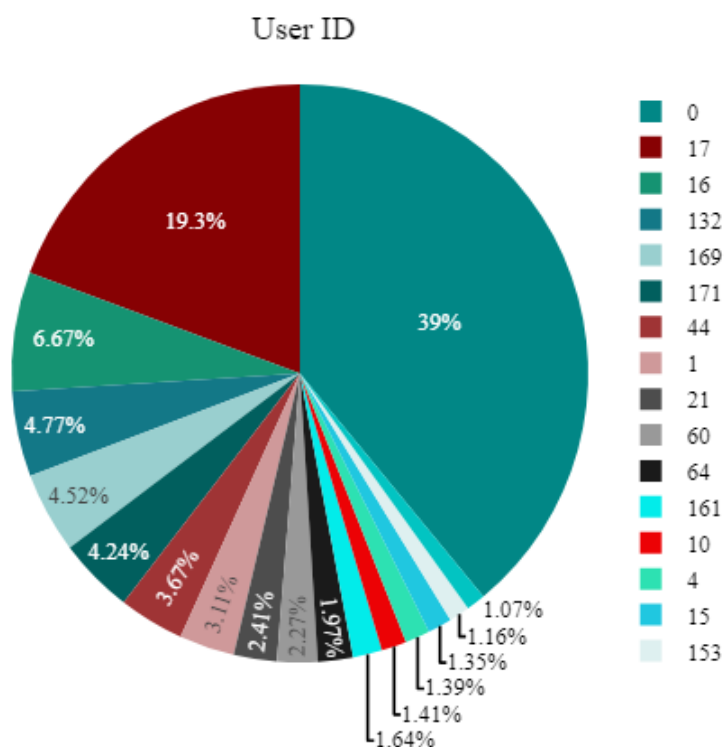


Fig. 2 - Distribuição de trabalho da partição 1

Esta tabela apresenta, individualmente, os users que realizaram pelo menos 1% do trabalho total. Somando a contribuição destes 15 Users, concluímos que, em conjunto, realizaram mais de metade dos trabalhos realizados, aproximadamente 61%. Dentre estes, destaca-se o user com ID igual a 17, tendo realizado cerca de 19,33% do trabalho total desta partição.

A parte User ID igual a 0 corresponde à agregação de todos os restantes, que contribuíram em menos de 1% do trabalho. Verificamos assim que estes, em conjunto, representam 39% do total dos trabalhos, algo pouco significativo tendo em conta que este valor diz respeito a um total de 242 users.

Quais os pedidos requisitados ao sistema?

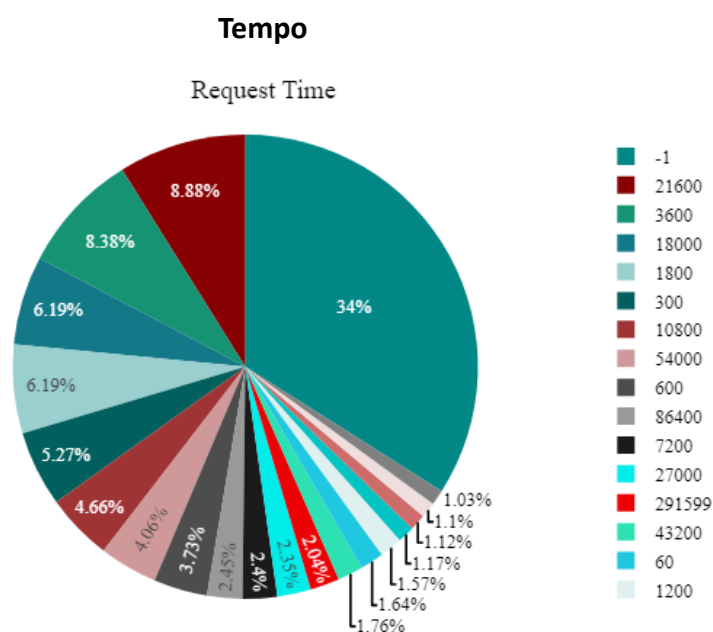


Fig 3 - Gráfico do tempo requisitado pelas aplicações.

O gráfico anterior apresenta os 20 tempos de execução mais requeridos no sistema e todos os restantes representados pelo valor -1.

Verifica-se que estes 20 pedidos estão presentes em 66% dos trabalhos sendo que os 34% englobam bastantes mais tempos pedidos,

Além disso, todos os tempo requeridos, com exceção do tempo 291599 ms, são múltiplos de 60 ms.

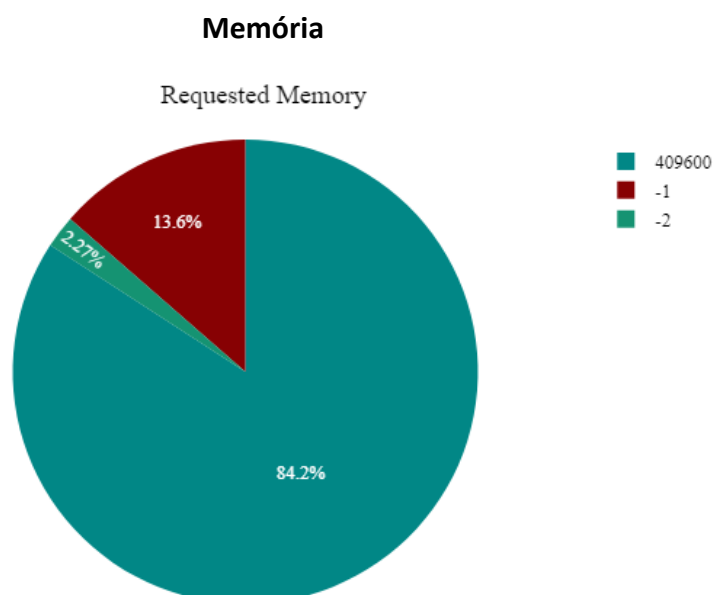


Fig 4 - Gráfico da memória requisitada pelas aplicações.

Este gráfico que apresenta o requisito de memória dos trabalhos não apresenta todos os requisitos, pois o número é elevado e seria pouco visual. Em vez disso, apresenta os requisitos mais comuns dos trabalhos realizados e junta os restantes numa só secção.

É fácil de perceber que a maioria do trabalho (84,2%) requisitou 409,6 kilobytes. Os restantes pedidos de memória (na legenda assinalados como -2) representam apenas 2,27% o que demonstra uma homogeneidade na memória requerida na aplicação executada.

Verifica-se ainda um caso especial, onde a memória requerida está assinalada como -1, que representa 13,6% do bolo total. Segundo os dados fornecidos isto significa que estes requisitaram 0 kilobytes de memória.

Número de processadores

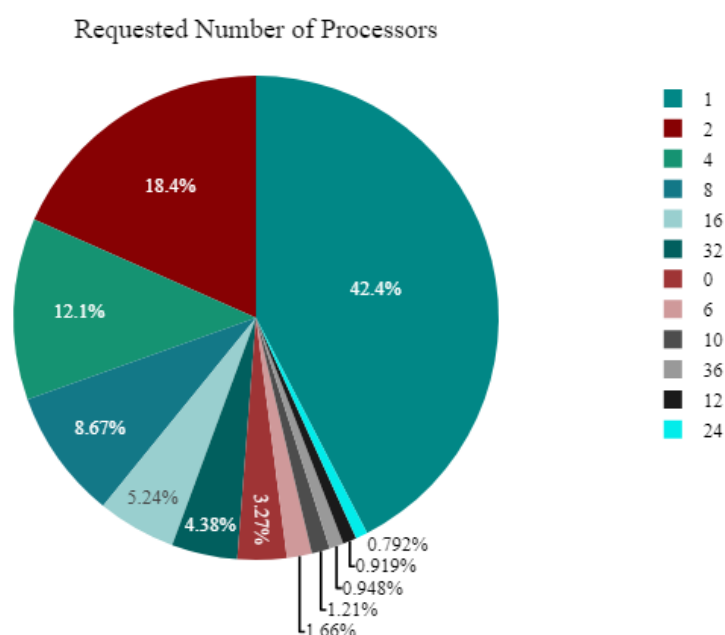


Fig 5 - Gráfico do tempo requisitado pelas aplicações.

Este gráfico permite evidenciar que o número de processadores requisitados variou entre 1 e 32 de um modo mais dominante (cerca de 91,19%). Sendo que o número de processadores mais requerido foi apenas 1 processador (42,4%), um valor considerável em relação aos restantes, apesar da requisição de 2 processadores também representar uma percentagem significativa do total dos trabalhos (18,4%). Além disso, os restantes número de processadores representados pelo valor 0, apresentam apenas 3,27% dos pedidos, evidenciando que as aplicações possivelmente foram desenvolvidas para um número de processadores baixo.

Os pedidos requisitados foram satisfeitos? Foram realísticos?

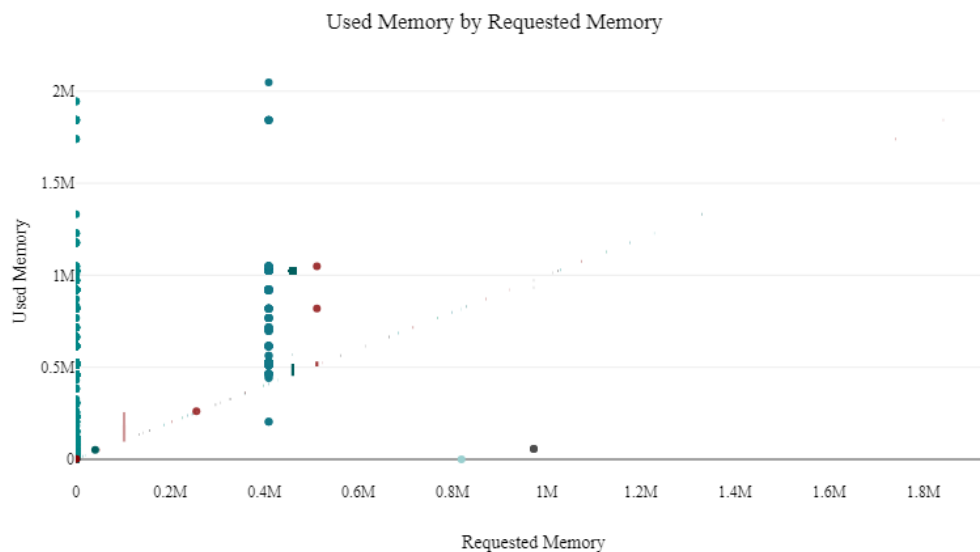


Fig 6 - Tabela que representa a relação entre memória usada com memória requisitada

Apesar de não ser muito perceptível, verificam-se 3 coisas importantes. O gráfico assemelha-se a função $f(x) = x$, ou seja, demonstra que o número de memória requerida na maioria dos casos foi a utilizada, verifica-se ainda uma linha no início do gráfico que corresponde aos trabalhos que não requisitaram memória e ainda outra linha, dos casos em que a memória requerida foi 409,6 kilobytes.

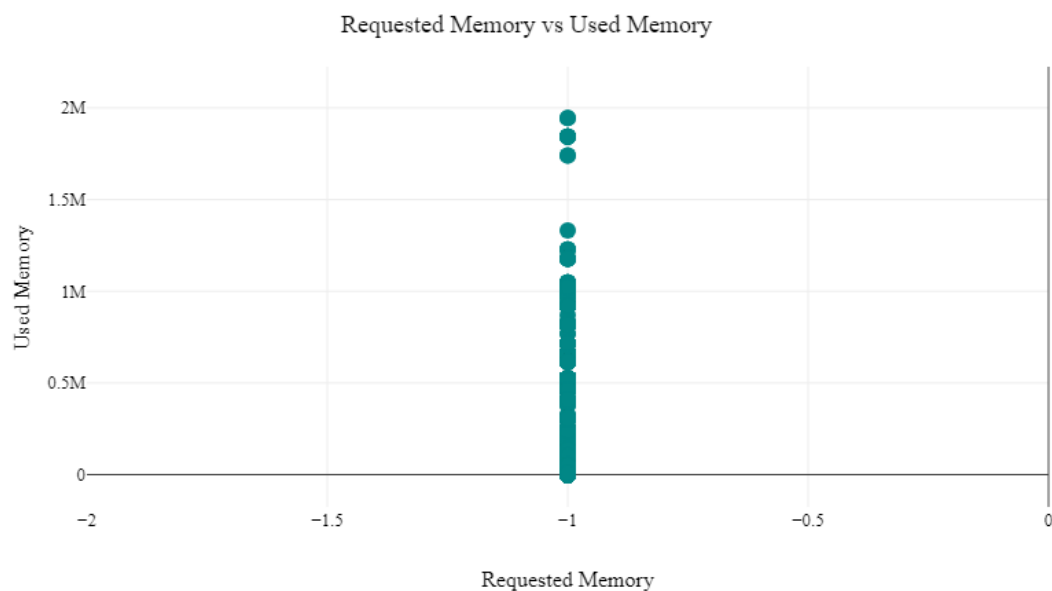


Fig 7 - Aproximação da Figura

Com a ajuda deste gráfico focado na Requested Memory igual a -1, ou seja, os trabalhos que não requereram memória, podemos visualizar que estes usaram principalmente valores entre 0 e 1 GB (1M de kilobytes). Existem valores superiores a 1GB mas estes apresentam-se em número mais reduzido em relação aos restantes. Verifica-se ainda que o valor máximo foi 2 GB.

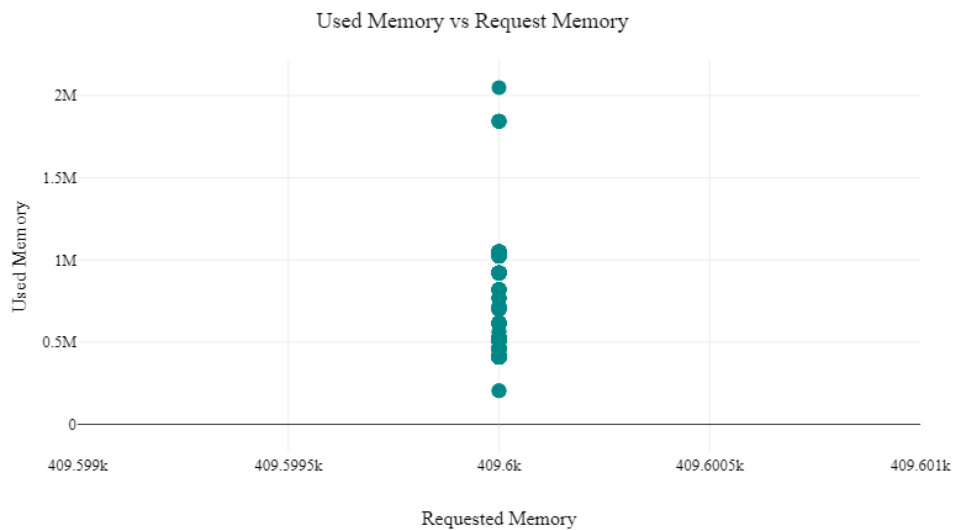


Fig 8 - Aproximação da Figura

Com a ajuda deste gráfico focado na requested Memory igual a 409,6 kilobytes podemos observar que esta memória requerida foi mais vezes excedida do que o contrário. Devido a esta requisição de memória ser a mais comum, podemos supor que seja um standard das aplicações e que vários fatores possam fazer com que existam desvios na memória realmente utilizada, sendo um desses fatores o número de processadores alocados.

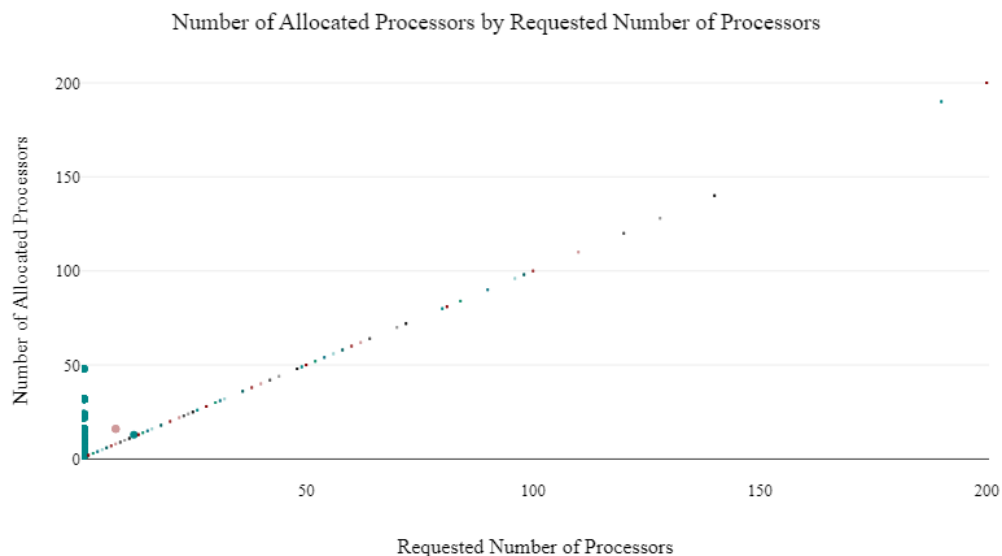


Fig 9 - Número de processadores alocados por número de processadores requisitados

Apesar de não ser muito perceptível, verificam-se 2 coisas importantes. Primeiramente, o gráfico assemelha-se a função $f(x) = x$, ou seja, demonstra que o número de processadores alocados está de acordo com os requeridos, na maioria dos casos.

Verifica-se que quando foi requerido apenas 1 processador em muitos casos foram alocados mais processadores para a realização da tarefa. Além destes casos, verificam-se ainda mais 2 pedidos em que número de processados requisitados foi excedido. Não há

nenhum caso em que foram alocados menos processadores do que os que foram requisitados.

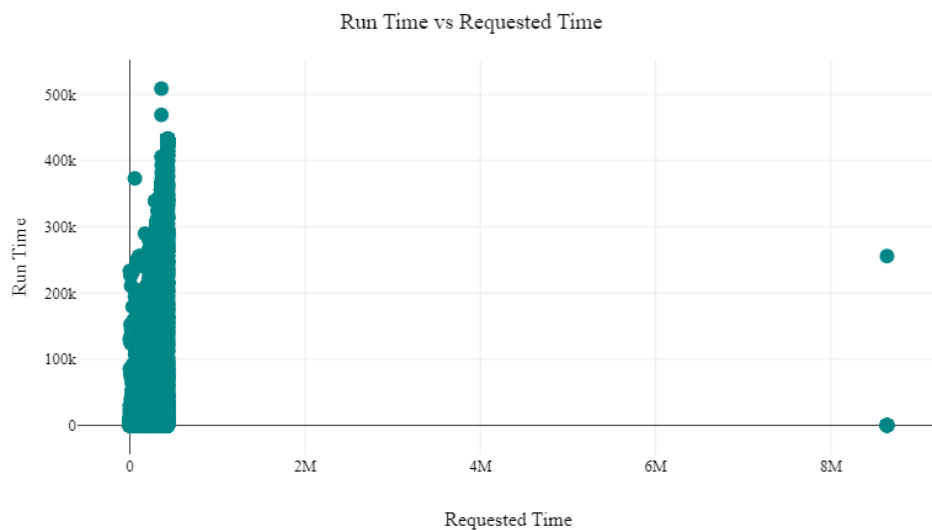


Fig 10 - Run Time relacionado com o Request Time

Este gráfico devido à grande quantidade de dados e à presença de dois outliers não ficou muito visual nem permite conclusões mais precisas, porém é possível verificar que os request time com o valor acima de 8 milhões de milissegundos foram pedidos muito exagerados, visto que o real Run Time foi inferior a 300 mil milissegundos.

Após remover os outliers identificados foi possível uma melhor observação destes campos.

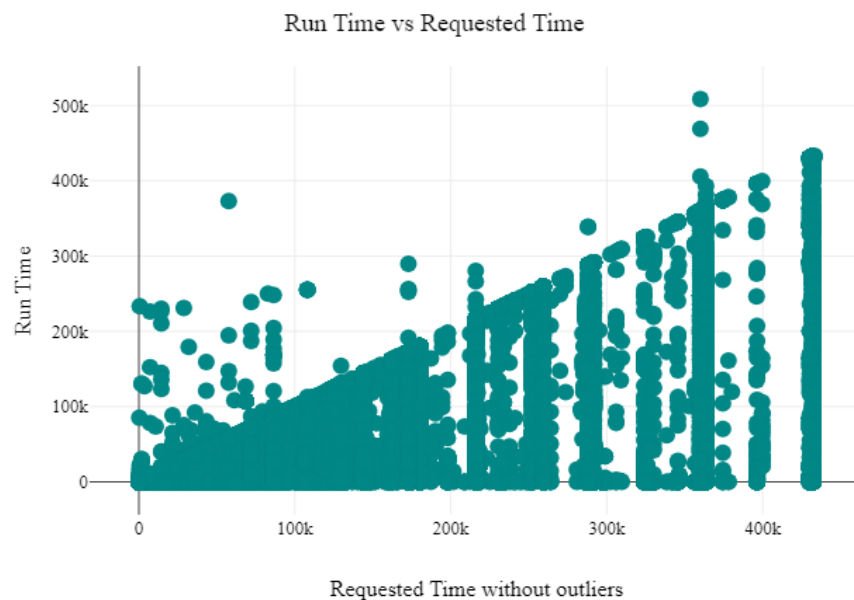


Fig 11 - Run Time relacionado com o Run time sem outliers

Analisando este gráfico, é perceptível verificar uma linha que se assemelha ao gráfico da função $f(x) = x$, como nos casos anteriores. Porém neste caso temos bastantes mais casos em que o tempo requerido foi superior ao tempo de execução. Decidimos focar nos tempos requeridos abaixo de 100k, para melhor análise.

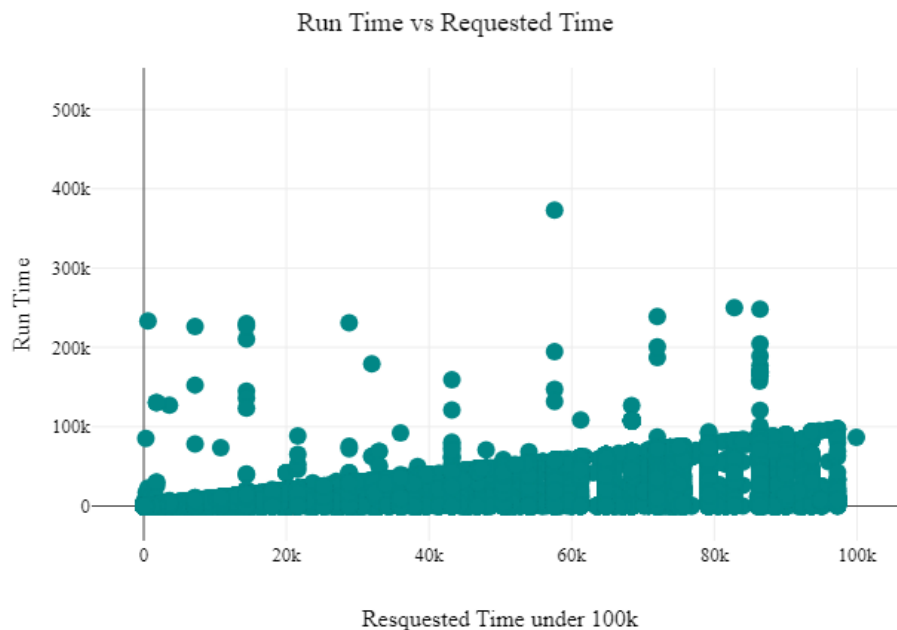


Fig 12 - Run Time relacionado com Request Time com valores inferiores a 100k

Focando na gama entre o 0 e os 100k de tempo requerido podemos evidenciar que foi aí que se verificaram os casos onde o tempo real de execução foi superior ao tempo requerido, o que pode ser explicado por ter sido um pedido demasiado otimista ou o número de processadores requeridos não foi o mais correto, pois o número de processadores alocados nunca foi abaixo de pedido.

Qual o custo das aplicações mais usadas e dos seus pedidos (Memória usada, Run Time e número de processadores alocados) para o sistema ?

| User ID | Used Memory | % |
|------------------|-------------|-------|
| Principais Users | 40960 | 94.45 |

| User ID | Total Run Time | User ID | Total Run Time |
|--|----------------|-----------|----------------|
| 17 | 56.98508M | 60 | 12.83755M |
| 16 | 84.55036M | 64 | 8.358496M |
| 132 | 95.2392M | 161 | 60.8813M |
| 169 | 168.4778M | 10 | 37.30356M |
| 171 | 410.6109M | 4 | 37.30356M |
| 44 | 267.5428M | 15 | 193.6215M |
| 1 | 96.62928M | 153 | 140.9784M |
| 21 | 50.20693M | 0 | 1.357303B |
| Total Run Time em conjunto (excepto 0) | | 2.092869B | |

Tabela relação user ID, total RunTime

| | | Number of Allocated Processors | | | | | | | | | | | | Total |
|---------|-----|--------------------------------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| | | 0 | 1 | 2 | 4 | 6 | 8 | 10 | 12 | 16 | 24 | 32 | 36 | |
| User ID | 0 | 2.27% | 7.45% | 6.78% | 7.05% | 1.63% | 6.4% | 0.59% | 0.62% | 3.97% | 0.21% | 1.94% | 0.11% | 39.02% |
| | 1 | 0.41% | 0.01% | 0.03% | 0.35% | 0.01% | 0.11% | 0.63% | 0.23% | 0.18% | 0.42% | 0.11% | 0.62% | 3.11% |
| | 4 | 0.22% | 0.02% | 0.03% | 0.16% | 0% | 0.07% | 0% | 0.04% | 0.41% | 0.13% | 0.18% | 0.13% | 1.39% |
| | 10 | 0% | 1.36% | 0.03% | 0% | 0% | 0% | 0% | 0% | 0.01% | 0% | 0% | 0% | 1.41% |
| | 15 | 0.02% | 0.03% | 0.11% | 0.25% | 0% | 0.33% | 0% | 0.02% | 0.35% | 0.01% | 0.19% | 0.05% | 1.35% |
| | 16 | 0% | 6.67% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 6.67% |
| | 17 | 0% | 7.34% | 7.02% | 4.04% | 0% | 0.87% | 0% | 0% | 0.05% | 0% | 0% | 0% | 19.33% |
| | 21 | 0% | 0.23% | 0% | 0.01% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 2.41% |
| | 44 | 0% | 3.67% | 0.13% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 3.67% |
| | 56 | 0.36% | 0.13% | 0.18% | 0.16% | 0.02% | 0.06% | 0% | 0.01% | 0.07% | 0% | 0.04% | 0.03% | 1.07% |
| | 60 | 0% | 3.97% | 2.23% | 0% | 0% | 0.01% | 0% | 0% | 0% | 0% | 0% | 0% | 2.27% |
| | 64 | 0% | 0.41% | 0.01% | 0% | 0% | 0.02% | 0% | 0% | 0.03% | 0.03% | 1.88% | 0% | 1.97% |
| | 132 | 0% | 4.77% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 4.77% |
| | 153 | 0% | 1.13% | 0.02% | 0.02% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 1.16% |
| | 161 | 0% | 0.49% | 0.43% | 0.2% | 0% | 0.5% | 0% | 0% | 0.01% | 0% | 0% | 0% | 1.64% |
| | 169 | 0% | 4.52% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 4.52% |
| | 171 | 0% | 2.06% | 1.52% | 0.07% | 0% | 0.35% | 0% | 0% | 0.17% | 0% | 0.05% | 0% | 4.24% |
| Total | | 3.29% | 42.07% | 18.39% | 12.32% | 1.67% | 8.72% | 1.21% | 0.92% | 5.26% | 0.8% | 4.4% | 0.95% | 100% |

Tabela relação user ID com número de processadores alocados

Estas três tabelas permitem evidenciar diferentes factos sobre os 15 principais utilizadores. Ao nível de memória utilizada, verifica-se que 94,45% dos trabalhos realizados pelos 15 principais users usaram 409,6 kilobytes, o valor “standard” do sistema.

No tempo de execução total destes users verificou-se um dado curioso: os users que mais trabalhos realizaram por vezes não são os que possuem maior tempo de execução total, como por exemplo o user 17. Isto evidencia que users com menos trabalho possam ter um custo maior para o sistema que users que realizaram mais trabalho, pois este tempo está relacionado com a aplicação executada como também pelos número de processadores alocados.

Apesar da soma do tempo de execução total destes utilizadores ser superior à dos restantes, a diferença entre estes valores é sensivelmente 1000 milhões de milissegundos, ou seja, aproximadamente 12 dias de diferença. Isto evidencia que a diferença é muito reduzida tendo em conta estes dados terem sido recolhidos ao longo de 4 anos.

Ou seja, o custo destes 15 utilizadores ao nível temporal não foi muito superior aos restantes.

Ao nível do número de processadores alocados verificou-se uma tendência do uso de 1 processador (42.07%) e 2 processadores (18,39%). Isto demonstra que ao nível de processadores utilizados estes utilizadores não tiveram um custo elevado ao sistema comparativamente aos restantes.

NASA

Após prepararmos os dados utilizando Python para inseri-los no DATAtab, identificamos que alguns dos campos de dados não seriam apropriados para a análise deste conjunto de dados.

Esses campos incluem: Wait Time, Average CPU Time Used, Used Memory, Requested Number of Processors, Requested Time, Requested Memory, Status, Queue Number, Partition Number, Preceding Job Number e Think Time from Preceding Job.

Isso significa que temos apenas 6 campos de dados que são relevantes para nossas análises. Um dos seis campos continha informações temporais (Run Time), mas não estava disponível para uma análise mais detalhada. Para abordar essa questão, introduzimos novos campos, incluindo o dia da semana, o dia e o mês.

Com base nas informações fornecidas sobre o conjunto de dados e sabendo que os registos são provenientes do supercomputador iPSC/860, desenvolvido pela Intel no final da década de 1980 e capaz de processamento paralelo, sabendo ainda que carga de trabalho no iPSC/860 é uma combinação de tarefas interativas e em lote, que inclui desenvolvimento e produção, principalmente voltada para aplicações de aerociência computacional, podemos formular algumas questões.

Qual a distribuição do trabalho por User? Qual o grupo se apresenta em maior número?

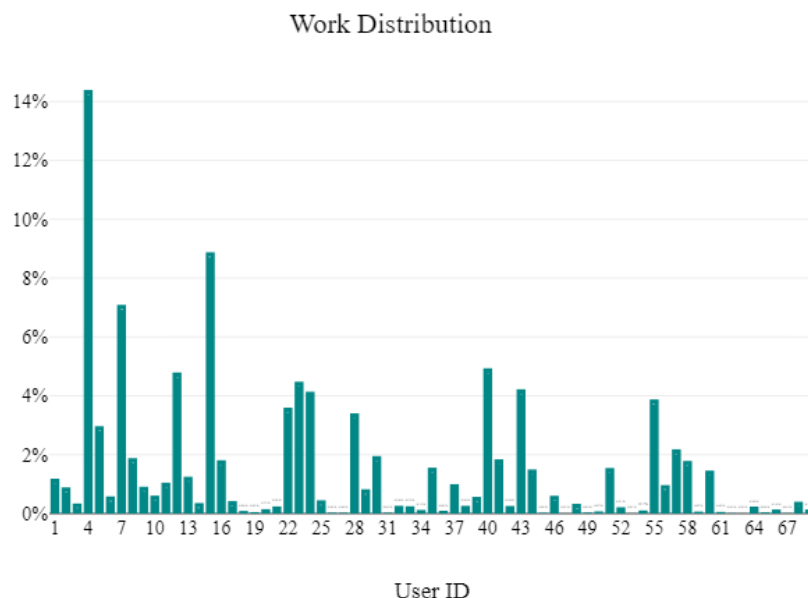


Fig 13 -Tabela com a distribuição do trabalho

Este gráfico destaca que o User ID igual a 4 (14%), 7 (7%), e 15 (9%) foram os mais proeminentes na realização de trabalho. Em conjunto, estes users contribuíram para aproximadamente 30% do trabalho total. Esses dados indicam que um número limitado de users, identificadas pelos User IDs mencionados, desempenharam um papel significativo na execução das tarefas, evidenciando uma distribuição não uniforme na carga de trabalho entre os diferentes users no sistema.

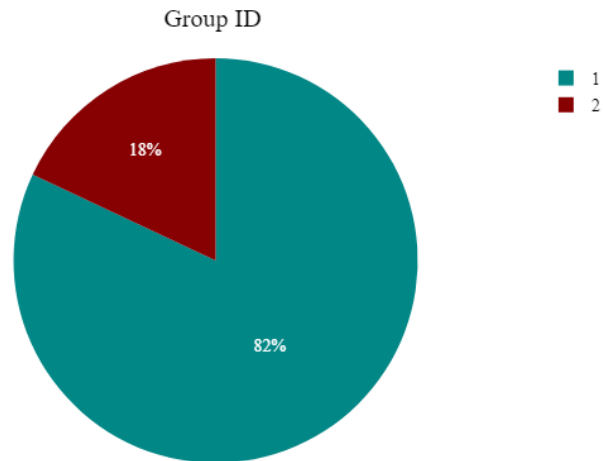


Fig 14 - Gráfico com a probabilidade de pertencer a um grupo.

Além disso, foi observado que aproximadamente 82% das máquinas pertencem ao Grupo 1, enquanto apenas 18% estão atribuídas ao Grupo 2. Essa distribuição evidencia uma disparidade entre a utilização do Grupo 1 e Grupo 2 como também no tempo de execução. Após a visualização do último gráfico surgiu nos uma questão:

Qual a principal diferença entre os dois Grupos?

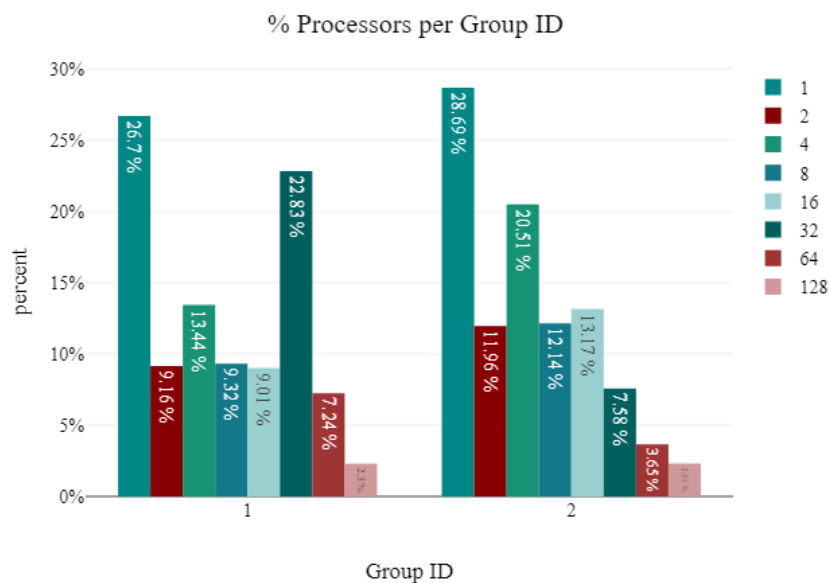


Fig 15 - Gráfico com as percentagens de processadores usados pelos Group ID

Ao analisarmos o gráfico, podemos observar que no Grupo 1, existe uma predominância do uso de 1 (26,7%), 32 (22,83%) e 4 (13,44%) processadores. Essas três categorias representam, em conjunto, cerca de 62,97% deste grupo, o que equivale a aproximadamente 63%. As demais distribuem-se de forma mais equitativa pelas categorias restantes, com exceção das máquinas que usam os 128 processadores, que são bastante raras, representando cerca de 2,3% do total no Grupo 1.

No Grupo 2, também notamos uma predominância do uso de 1 (28,69%) e 4 (20,51%) processadores, as quais compõem aproximadamente 49,2% desse grupo. A principal diferença entre os Grupos 1 e 2 é a proporção do uso de 32 e 64 processadores. No Grupo 2, apenas 7,58% usa 32 processadores, em comparação com os 22,83% do Grupo 1. No entanto, essa discrepância é menos pronunciada quando se trata do uso de 64 processadores, com o Grupo 1 apresentando 7,24% e o Grupo 2, 3,65%. No sentido contrário temos o uso de 4 processadores que representam uma proporção maior no Grupo 2.

Ambos os grupos apresentam duas características comuns. As categorias com maior e menor representatividade em ambos os grupos são as que usam 1 processador e as que usam a totalidade dos processadores 128 processadores, sendo que estas últimas apresentam a mesma proporção em ambos os grupos.

Qual a influência do número de processadores alocado, e qual a gestão deste recurso neste sistema?

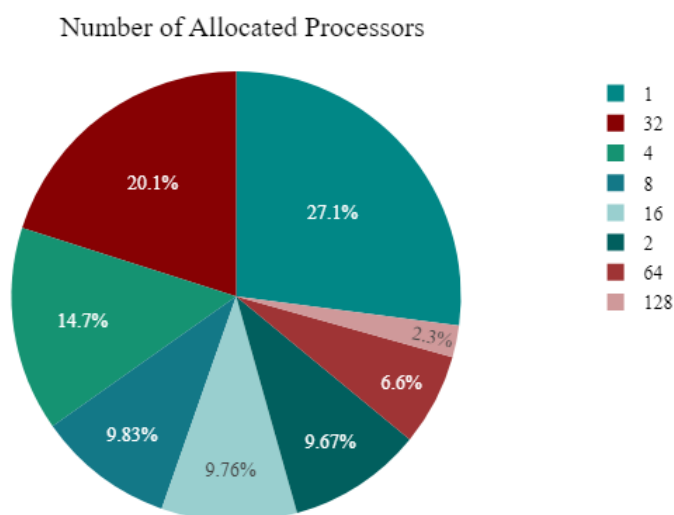


Fig 16 - Gráfico que representa o número de processadores alocados

Após a análise do número de processadores alocados, verificou-se que o número de processadores alocado segue a fórmula 2^N , onde N representa um expoente. O menor número de processadores alocado é 1 (2^0), e o maior número é 128 (2^7).

Além disso, notamos que 27,1% das máquinas utilizavam apenas 1 processador, enquanto 20,1% das máquinas utilizavam 32 processadores, e 14,7% das máquinas utilizavam somente 4 processadores. Em outras palavras, cerca de 61,9% dos trabalhos realizados utilizavam apenas 1, 32 ou 4 processadores, destacando-se a predominância dessas três categorias.

Nos restantes 38,1% das máquinas, houve uma distribuição quase equitativa na quantidade do uso de 8, 16 ou 2 processadores, seguidas de perto pelo uso de 64 processadores. O uso total dos 128 processadores registrou menos atividade, muito provavelmente porque ao usar a totalidade dos processadores da máquina, apesar de existir uma vantagem temporal, retiraria a possibilidade de realizar trabalho em paralelo.

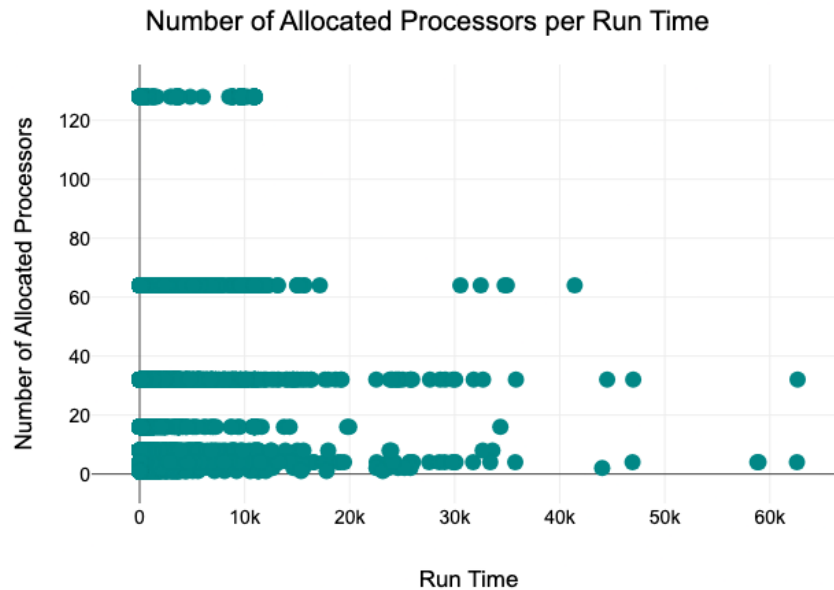


Fig 17 - Gráfico com a relação do número de processadores alocados com Run Time

Este gráfico ilustra que há uma tendência inversamente proporcional entre o número de processadores utilizados e o Run Time, como seria previsto. Ou seja, à medida que o número de processadores aumenta, o tempo de execução diminui. Esse comportamento é consistente com a expectativa geral de que o paralelismo do processamento, através do uso de mais processadores, resulta em uma conclusão mais rápida das tarefas.

Essa análise evidencia que, embora o uso de 128 processadores possa apresentar melhores desempenhos em comparação com as demais em alguns cenários, embora o uso de apenas 1, 4 e 32 processadores também consegue realizar tarefas semelhantes. Isso sugere que a predominância destas últimas categorias pode estar diretamente relacionada à relação custo/benefício. Em outras palavras, a utilização de uma menor quantidade de processadores pode oferecer um desempenho satisfatório, deixando recursos para outras tarefas e permitindo realizar um maior número de trabalhos num intervalo de tempo mais curto.

O trabalho realizado foi homogêneo ao longo das semanas/mês?

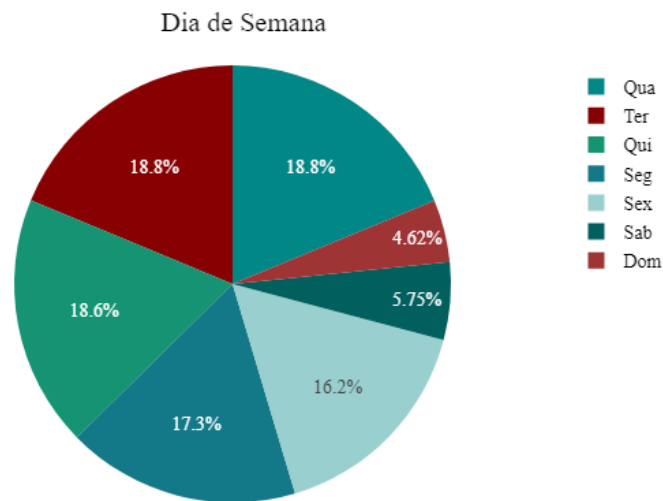


Fig 18 - Gráfico com a distribuição do trabalho pelo dias da semana

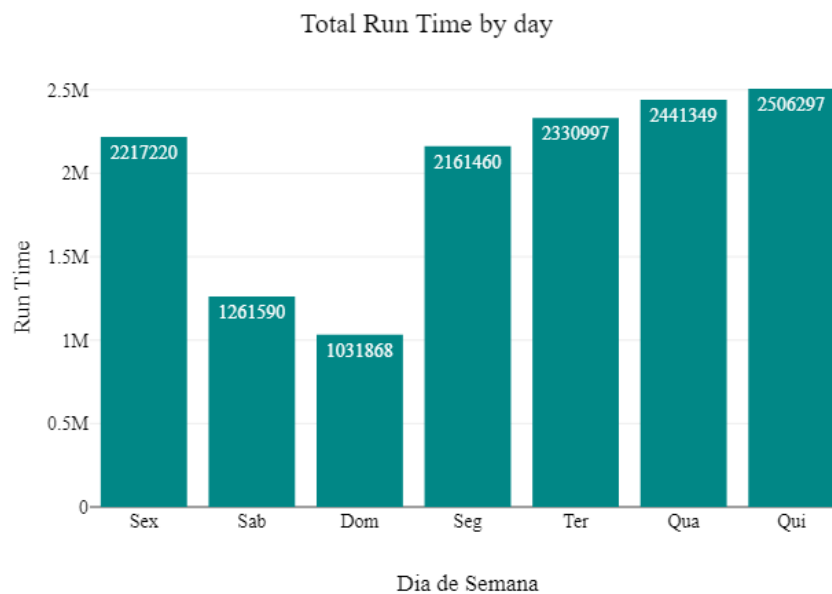


Fig 19 - Gráfico com o total Run Time pelo dias da semana

Após a análise desses dois gráficos, podemos observar que a distribuição do trabalho ao longo da semana foi quase uniforme, com cerca de 18% das tarefas sendo realizadas em cada dia, com exceção de sábado e domingo, nos quais os valores estiveram em torno de 5%.

Uma possível explicação para essa variação pode ser que durante o fim de semana, há menos recursos alocados para as tarefas, mas é importante ressaltar que não podemos chegar a uma conclusão definitiva com base apenas em suposições, uma vez que não dispomos de dados que sustentem essas conjecturas.

Analisando os dados que temos na nossa posse verificamos que o tempo total de execução também foi relativamente uniforme ao longo da semana, com exceção do sábado e domingo, quando houve uma diferença notável. Isso sugere que, durante os finais de semana, o tempo de execução das tarefas pode ser afetado de alguma forma. As únicas informações que podem comprovar as conclusões acima obtidas são que existe uma diferença nas filas de execução dos trabalhos durante "Prime time" que é definido como de segunda a sexta-feira, e o restante tempo onde todas as filas estão em execução e NOS (Network Queuing System) pode encerrar trabalhos interativos para liberar espaço para os trabalhos do NQS (Trabalhos mais longos). No entanto, é importante ressaltar que, assim como na distribuição do trabalho, essa análise é baseada em observações e nos dados à nossa disposição, ou seja, não podemos tirar conclusões definitivas sem dados adicionais que comprovem estas suposições.

A diferença observada no trabalho realizado nos dias da semana também se evidencia nos meses?

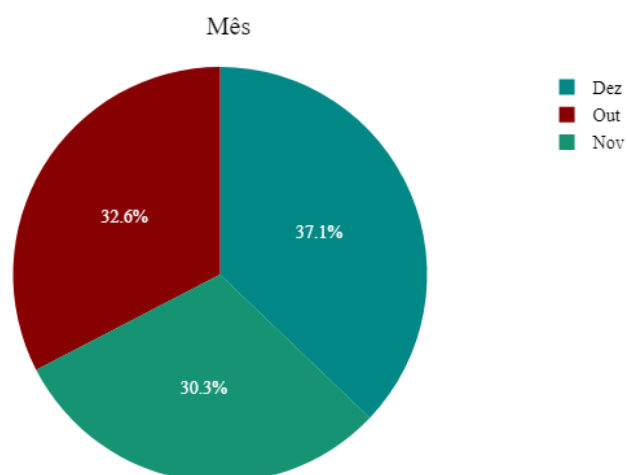


Fig 20 - Gráfico com a distribuição de trabalho por mês

Ao observar o gráfico, podemos concluir que nos dois primeiros meses, houve uma realização quase igual de trabalhos, com 32,6% para o primeiro mês e 30,3% para o segundo mês. No entanto, no último mês, que é dezembro, embora não tenha havido uma grande diferença em relação aos meses anteriores, foi o mês em que se executou o maior número de tarefas (37,1%).

O Runtime ao longo dos meses teve uma tendência crescente ou decrescente?

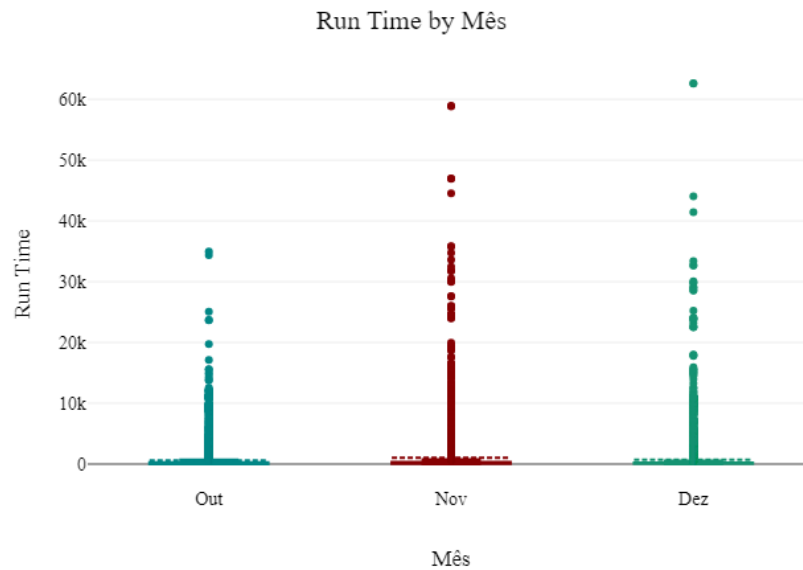


Fig 21 - Gráfico que apresenta a relação entre o Run Time por mês

A partir da análise do gráfico, podemos notar que o tempo de execução (Run Time) não apresentou uma tendência crescente ao longo dos meses, mas foi no mês de dezembro que se registou o tempo de execução mais elevado. No entanto, é interessante observar que quando comparamos o mês de dezembro com novembro, conseguimos observar que, o mês de dezembro apresenta alguns valores mais baixos comparativamente novembro quando nos focamos em valores de tempo de execução compreendido entre 30k a 40k, o que pode ser um ponto de destaque, sugerindo que nem todos os dias do mês de dezembro tiveram um tempo de execução superior aos do mês de novembro.

Que mês teve valores maiores retirando os valores de RunTime superiores a 40k?

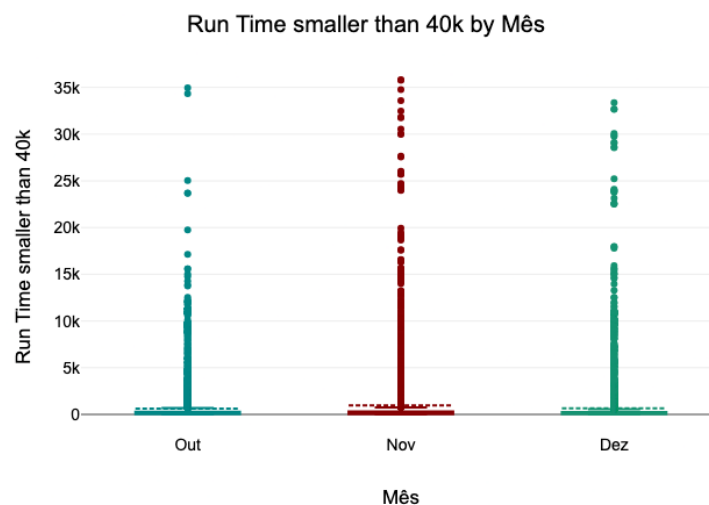


Fig 22 - Gráfico que apresenta a relação entre o Run Time inferior a 40k por mês(outliers)

| | | Mean | Sum | Maximum |
|---------------------------|-----|--------|---------|---------|
| Run Time smaller than 40k | Dez | 658.78 | 4459282 | 33382 |
| | Out | 620.37 | 3687499 | 34962 |
| | Nov | 967.4 | 5337136 | 35829 |

Tabela resultante do gráfico de cima que mostra a média, a soma e o valor máximo do Run time.

Analisando o gráfico, como retiramos os outliers acima de 40000 de Runtime podemos comparar com mais facilidade cada mês, sendo que é perceptível a visualização de maiores pontos no mês de Novembro em relação aos outros meses. Apesar disso, apresentamos uma tabela que representa a média dos valores do Runtime, a soma de todos os RunTime's de cada mês e o Runtime máximo de cada mês, sendo que podemos concluir que o mês que houve mais RunTime foi o mês de Novembro, como imaginávamos acima. A implementação deste gráfico ajudou-nos a concluir melhor, visto que em Dezembro tínhamos pontos bastante fora do normal.

Quais foram as aplicações mais executadas?

| Executable (Application) Number | % de execuções | Executable (Application) Number | % de execuções |
|---------------------------------|----------------|---------------------------------|----------------|
| 4 | 5.97% | 11 | 2.35% |
| -1 | 5.72% | 20 | 2.35% |
| 3 | 4.94% | 1 | 2.11% |
| 101 | 3.5% | 18 | 2% |
| 297 | 3.11% | 76 | 1.94% |
| 134 | 3.1% | 324 | 1.69% |
| 10 | 3.08% | 370 | 1.44% |
| 347 | 3.04% | 37 | 1.41% |
| 12 | 3.03% | 2 | 1.3% |
| 9 | 2.92% | 286 | 1.2% |
| 80 | 2.79% | 67 | 1.19% |
| 19 | 2.68% | 379 | 1.14% |
| 28 | 2.67% | - | - |

Fig 23 - 25 Principais aplicações executadas

Neste gráfico que considera apenas 25 aplicações, representativas do maior número de tarefas realizadas, podemos observar que a percentagem de tarefas para as quais a aplicação é desconhecida e a percentagem da aplicação mais executada são praticamente iguais. As três principais aplicações foram a aplicação número 3, a aplicação desconhecida e a aplicação número 4.

À medida que descemos na lista, podemos agrupar as aplicações com base nas suas percentagens. Algumas encontram-se na faixa de 4% a 5%, enquanto outras estão na faixa de 2% a 3%, e assim por diante.

Qual a influência das aplicações no Run Time?

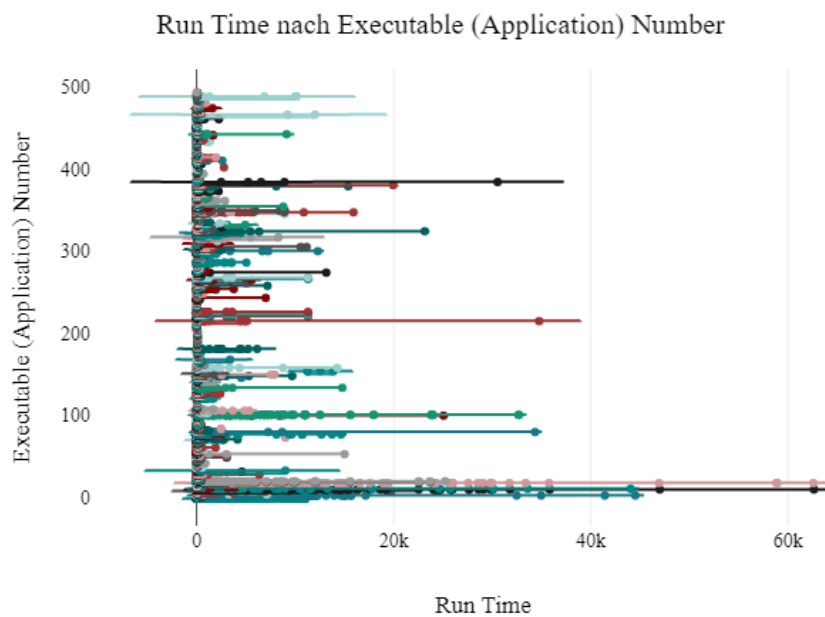


Fig 24- Relação entre Runtime e as aplicações

Este gráfico não nos ajuda a responder a nossa questão, embora sugere que cada aplicação pode possuir diferentes tempos de execução (Run Time).

No entanto, com a ajuda do gráfico é possível evidenciar que o tempo de execução (Run Time) variou de 0 a 60k milissegundos embora a maioria das aplicações obteve tempos abaixo de 40k.

Visto que o gráfico tem bastantes valores decidimos focar nas primeiras 25 aplicações, porque nesta gama estão presentes 12 das 24 aplicações mais executadas neste sistema.

Além disso, aplicações com número entre -1 e 25 representam cerca de 41% do total uso das aplicações, o que é um número relativamente alto visto que existem 493 aplicações.

Para isso iremos realizar mais 4 gráficos para que tenhamos conclusões mais coerentes.

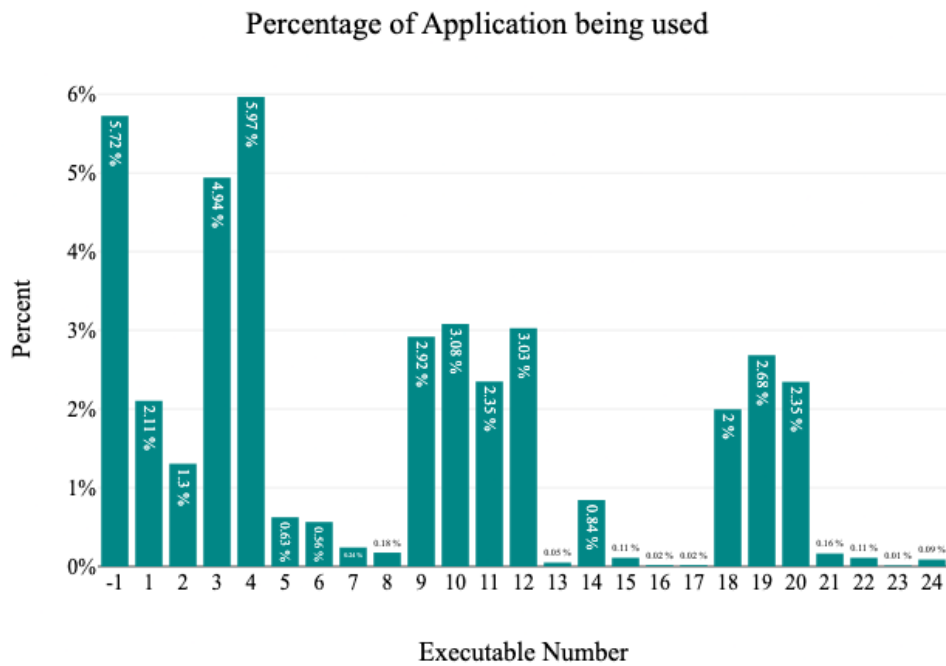


Fig 25- Gráfico que mostra a percentagem de aplicações usadas

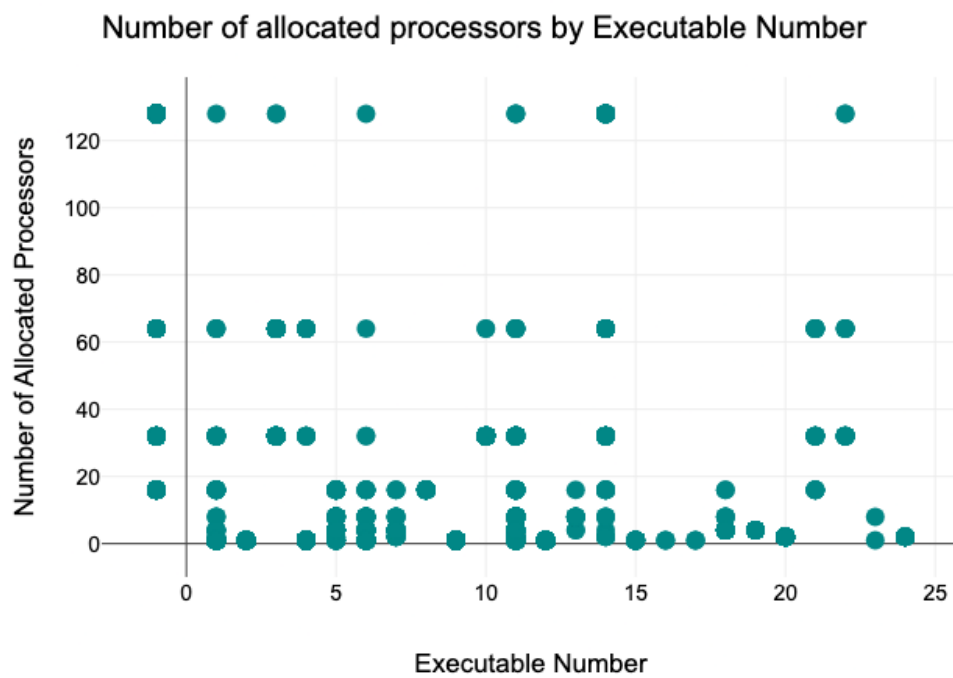


Fig 26 - Gráfico que relaciona o número de processados alocados por aplicação

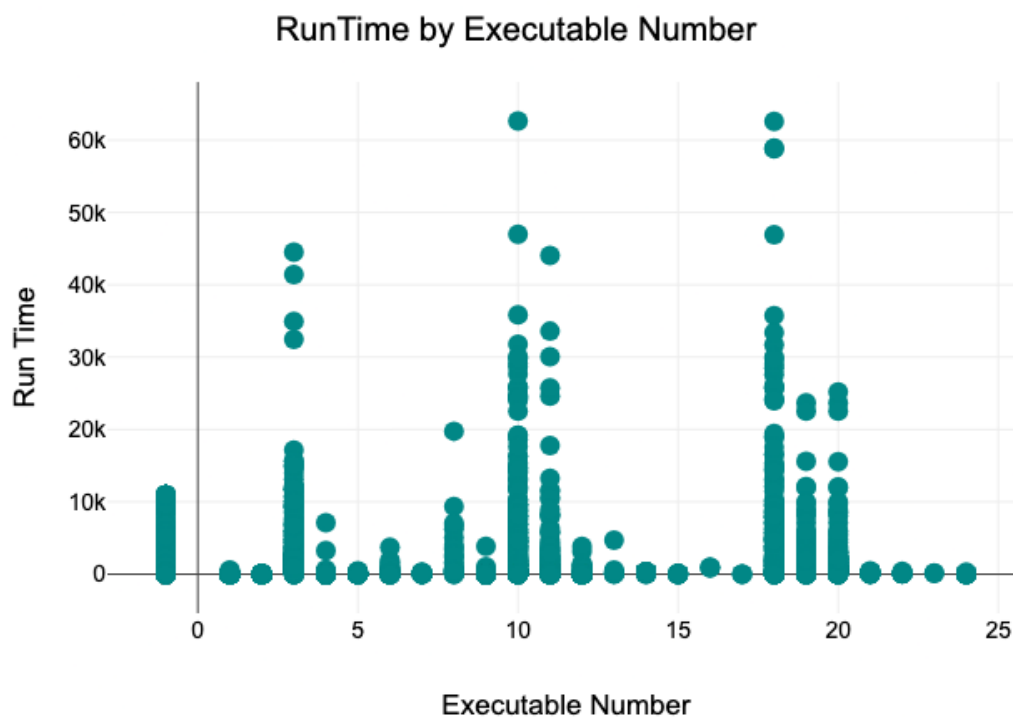


Fig 27 - Gráfico que relaciona o RunTime por aplicação

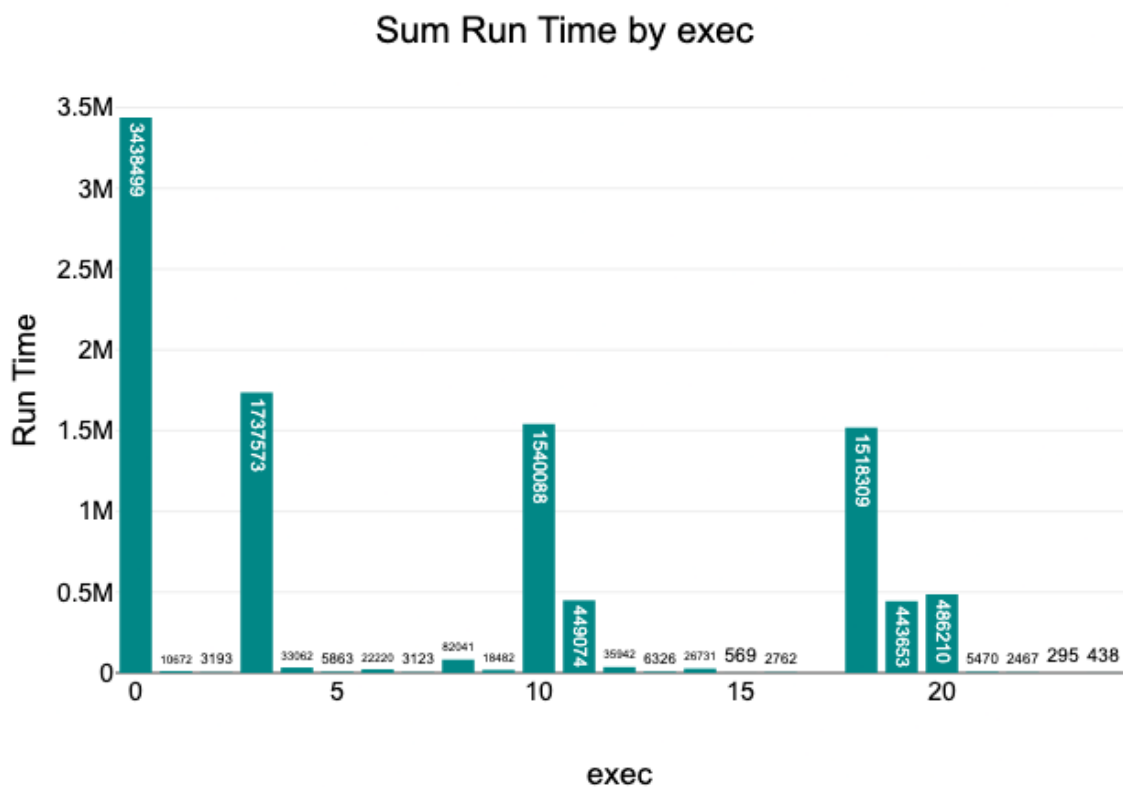


Fig 28 - Gráfico que da soma dos runtimes para cada aplicação(0 representa o -1)

Após visualizar o primeiro gráfico que representa a percentagem de uso das aplicações, decidimos abordar as 3 aplicações mais executadas desta gama mas também no geral, ou seja, as aplicações com número 4,-1,3.

Como podemos visualizar a aplicação mais usada foi a aplicação número 4. Focando na figura número 25 podemos visualizar que a maior parte dos seus jobs têm um RunTime entre 0k a 8k, um valor relativamente baixo em relação às outras aplicações. Estes valores influenciam o RunTime total pois apesar desta aplicação ser a mais executada, fazem com que o tempo total gasto pela aplicação 4 seja um dos mais baixos do sistema. Analisando o número de processadores alocados para esta aplicação verificamos que foram alocados 1,32,64 processadores, sendo na maioria dos jobs usado apenas 1 processador. Para complementar a análise dos processadores verificamos que este trabalho foi apenas executado pelo Grupo 1, e com execução dos 64 processadores alocados, os restantes são os principais valores usados por este grupo.

A aplicação com número -1, representa todos os jobs cuja aplicação é desconhecida, cerca de 5.72% dos trabalhos total. Este valor apenas difere 0.25% da principal aplicação, o que sugere que são um valor considerável para a falta de informação do sistema. O seu RunTime não ultrapassa valores superiores a 12k mas verifica-se a existência de bastantes valores entre os 0 e os 12k o que pode explicar o RunTime total destas aplicações ser o mais elevado. O número de processadores alocados para estes jobs varia entre 16 e 128 processadores, este valor conjuntamente com o RunTime pode ser explicado pela existência de várias aplicações desconhecidas com requisitos diferentes e que apesar de não influenciar o valor máximo de RunTime, influencia o seu total.

A aplicação 3, a terceira aplicação mais executada, tem os valores do seu RunTime num intervalo entre 0k e 45k (o valor mais elevado das 3). Ao analisar o número de processadores alocados verificamos que este variou entre 32 a 128 processadores. Ou seja, teve um custo mais elevado para o sistema que a aplicação mais utilizada. Apesar dos valores do RunTime serem elevados, esta aplicação apenas apresenta o 2 maior valor total de RunTime, o que pode ser explicado pela diferença de 0,78% em relação à aplicação número -1.

Concluindo o custo das aplicações para o sistema varia consoante a percentagem de utilização, o seu RunTime e o número de processadores alocados. Sendo que muitas vezes a aplicação um destes fatores condiciona os restantes, uma boa gestão dos três pode minimizar o custo das aplicações ao sistema e melhorar a gestão de recursos.

SDSC

Depois de preparar a data usando Python para colocá-la no DATAtab, descobrimos que alguns dos Data Fields não seriam adequados para tirar conclusões neste conjunto de dados (Used Memory; Request Number of Processors; Request Time, Request Memory, Group ID; Executable (Application) Number; Preceding Job Number; Think Time from Preceding Job).

Isso significa que temos apenas 9 campos de dados que são relevantes para nossas análises. Um dos seis campos continha informações temporais (Run Time), mas não estava disponível para uma análise mais detalhada. Para abordar essa questão, introduzimos novos campos, incluindo o dia da semana, o dia e o mês.

Com base nas informações fornecidas sobre o conjunto de dados e sabendo que os registros pertencem a um sistema que usa o Intel Paragon, um sistema de supercomputador paralelo desenvolvido pela Intel Corporation no início dos anos 1990, que foi projetado para computação de alto desempenho e normalmente usava uma topologia de hipercubo para sua rede de interconexão podemos formular algumas questões

Qual a percentagem de trabalho concluídos?

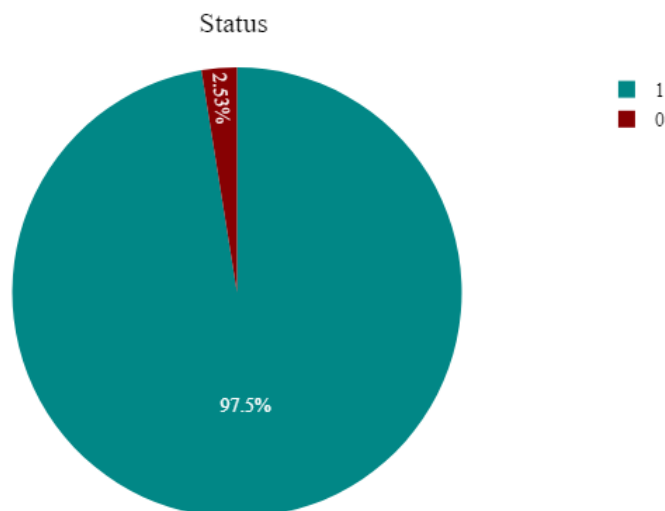


Fig 29 - Relação entre Runtime e as aplicações

Verificou-se que 97,5% dos trabalhos foram concluídos com sucesso e apenas 2,53% não foram concluídos. Isto evidencia uma boa taxa de sucesso.

Qual o maior tempo de espera / existem trabalhos realizados sem terem de esperar?

| | Wait Time |
|---------|-----------|
| Minimum | 0 |
| Maximum | 2865215 |

Fig 30 - Tabela que representa o tempo de espera nos trabalhos realizados.

Verificou-se que o maior tempo de espera obtido foi 2865215, e que além de existirem trabalhos que não tiveram de esperar para serem realizados, ainda existem trabalhos que não apresentam dados sobre este tempo de espera.

Existem dois tipos de partição: partições Interativa e Batch. Os trabalhos interativos são processados na partição Interativa, e os trabalhos de produção de longa duração são executados na partição de Batch.

O agendamento de trabalhos na partição Interativa é suportado por um mecanismo de atribuição sequencial de nós fornecido pelo sistema operacional Paragon. Os trabalhos são agendados com base no critério de "First-Come-First-Serve", podendo um trabalho interativo ser rejeitado se não houver nós suficientes livres para atender à solicitação.

O agendamento de trabalho na partição Batch é gerenciado pelo agendador SDSC batch. Todos os trabalhos são classificados na ordem da "prioridade_do_trabalho", e o agendador seleciona trabalhos para serem executados, começando pelo trabalho de maior prioridade. Se não houver nós livres suficientes para agendar o trabalho de maior prioridade, o próximo trabalho de maior prioridade será considerado e assim por diante.

Os trabalhos grandes têm mais dificuldade em serem agendados do que trabalhos menores. Para garantir que os trabalhos grandes sejam agendados, existe o conceito de bloqueio controlado por uma constante "prioridade_de_bloqueio" que é definida pelo administrador do NQS. Sob o bloqueio, eles são mantidos ociosos intencionalmente até que haja nós livres suficientes para agendar o trabalho de grande porte.

Tendo como base as conclusões das observações podemos fazer algumas suposições. O tempo de espera por ser afetado no caso dos trabalhos interativos pelo número de nós disponíveis, um trabalho pode ter de esperar um longo período caso nunca seja possível responder a sua solicitação.

E no caso dos trabalhos de longa duração o tempo de espera pode ser devido a prioridade dos trabalhos inseridos na queue, ao número de processadores disponíveis e ao tempo que este já esteve em espera.

Partindo destas informações e destes dados tentamos aprofundar mais o nosso conhecimento com ajuda dos dados

Existe uma predominância de uma das partições em relação à outra?

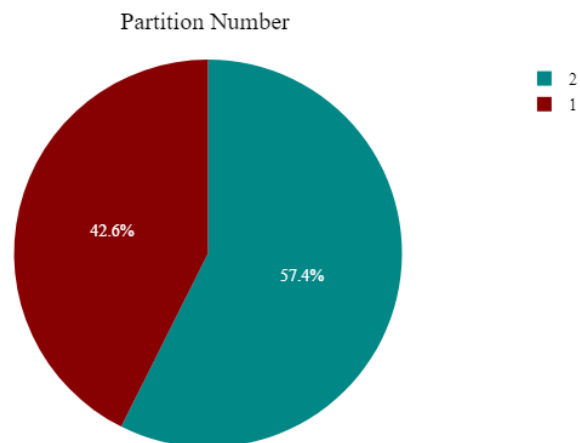


Fig 31 - Gráfico que representa a percentagem de trabalho em relação às partições.

Verificamos que houve um maior número de trabalhos realizados pela partição 2 (Partição Batch) em relação ao número de trabalho realizados pela partição 1 (partição interativa).

Será que conseguimos evidenciar as diferenças entre estas duas partições?

| Partition Number | Wait Time (ms) |
|------------------|-------------------|
| 1 | 0 |
| 2 | 18.21468K |

Tabela que representa a soma do Wait Time para cada partição.

Este gráfico que relaciona o nível total do tempo de espera (Wait Time) das duas partições permite responder melhor à pergunta: Existem trabalhos realizados sem terem de esperar? A resposta é sim, os trabalhos interativos (partição 1) não tiveram de esperar para serem executados, ao contrário dos trabalhos mais longos (Partição 2) tiveram um grande tempo de espera. O número de processadores usados em cada partição poderá ajudar a dissipar algumas dúvidas.

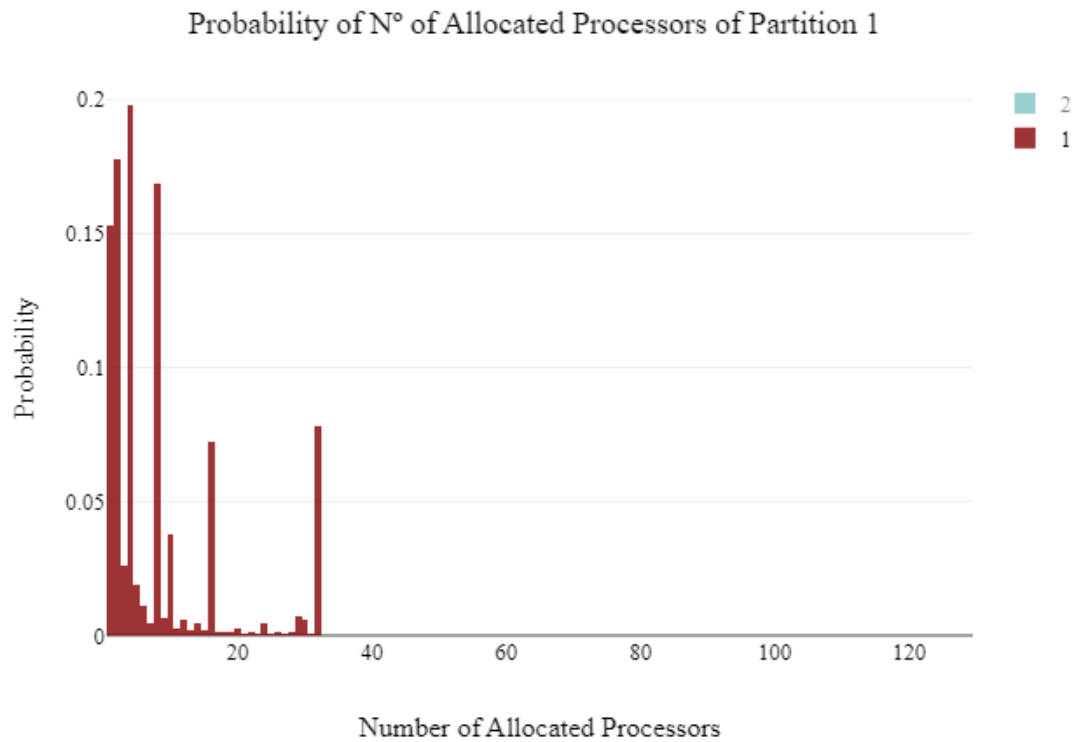
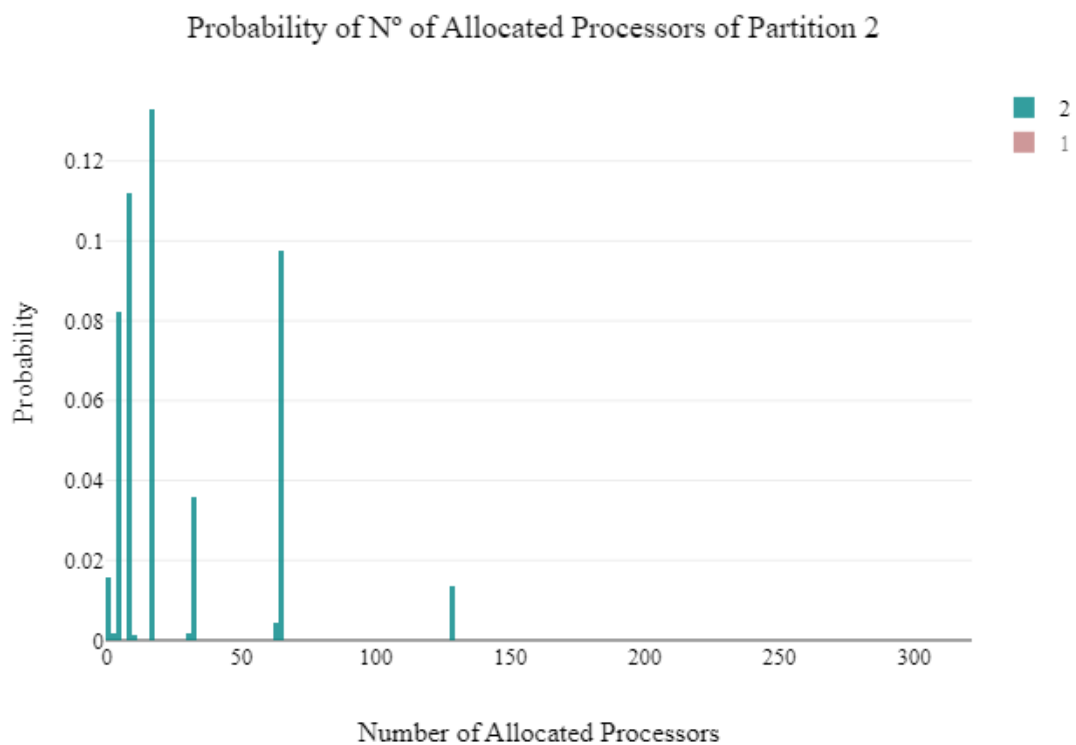


Fig 32 - Gráfico que representa a percentagem de processadores alocados na partição 1.



| Partition | Max Number of Allocated Processes |
|-----------|-----------------------------------|
| 1 | 129 |
| 2 | 320 |

Tabela com o número máximo de processadores alocados para cada partição.

Verificou se que o número de processadores alocados para os trabalhos interativos (partição 1) foi relativamente baixo em relação aos trabalhos longos (partição 2) como seria expectável, mas analisando mais aprofundadamente os valores podemos verificar que o máximo de processadores alocados na partição 1 foi 129 e no caso da partição 2 foi 320. Ou seja, também de acordo com o expectável, mas verificou se que os trabalhos longos muitas vezes usaram poucos processadores e possivelmente o tempo de espera destes trabalhos também possa estar relacionado com este facto, pois trabalhos que necessitam mais processadores teriam de esperar mais tempo em comparação com os que necessitam de menos.

Os trabalhos interativos não necessitam de esperar para ser executados, tendo em conta no máximo puderem ser alocados 129 processadores e a maioria dos trabalhos ter menos de 40 processadores alocados, podemos supor que foram realizados mais trabalhos em paralelo. Para tornar esta suposição em uma evidência focamos a nossa pesquisa no Run Time e a posição na Queue.

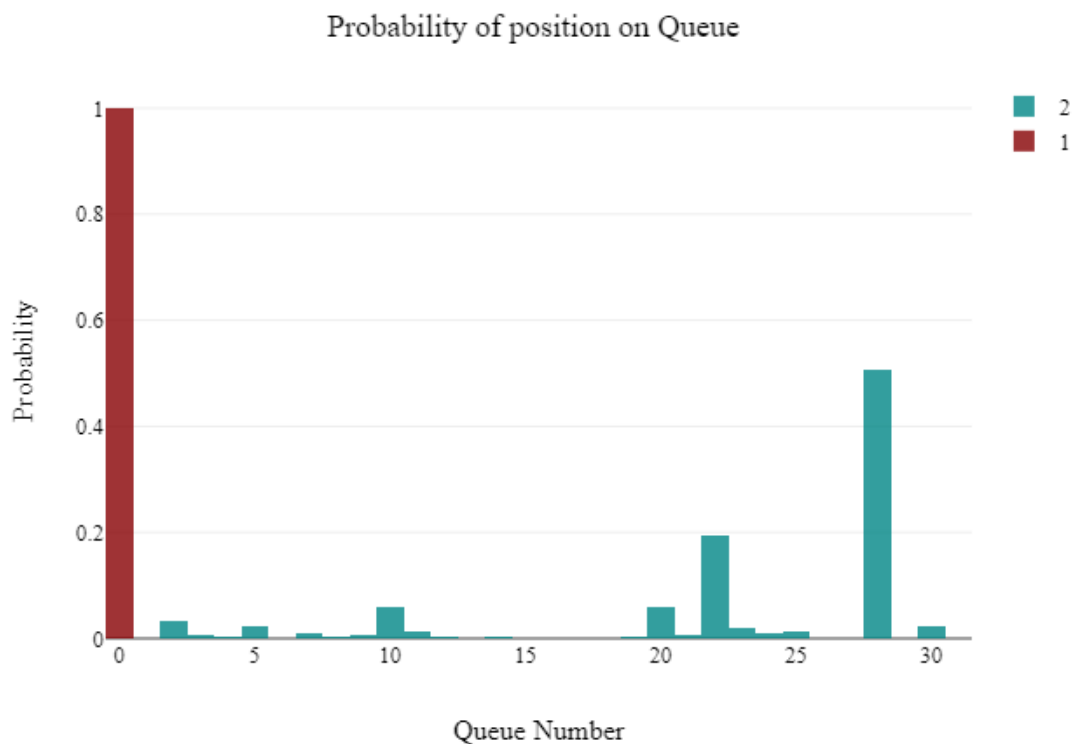


Fig 34 - Gráfico que mostra a percentagem da posição na fila.

Tendo em conta a posição na queue verificamos que os trabalhos interativos entram sempre na posição 0, como seria expectável pois anteriormente vimos que estes não tiveram tempo de espera. Por outro lado, nos trabalhos longos estes tiveram trabalhos a entrar em posições distintas.

| Queue Number | Maximum |
|--------------|---------|
| 1 | 0 |
| 2 | 31 |

Fig 35 - Tabela com a posição máxima em cada partição.

A posição máxima na queue da partição 2 foi 31, mas verifica-se que a maioria dos trabalhos entrou para a posição 20 em diante, o que aumentou consideravelmente o tempo de espera.

| Partition | Maximum Run Time |
|-----------|------------------|
| 1 | 448070 |
| 2 | 432127 |

Fig 36 - Tabela que representa o maior Run time de cada partição.

| Partition | Mean Run Time (ms) | Total Run Time (ms) |
|-----------|----------------------|------------------------|
| 1 | $623.23 \cdot 10^3$ | $14790.37 \cdot 10^3$ |
| 2 | $8534552 \cdot 10^6$ | $272749141 \cdot 10^6$ |

Fig 37 -Tabela que representa a média e o total Run time para cada partição.

Focando agora no tempo de execução (Run Time) podemos verificar uma curiosidade, o trabalho que demorou mais tempo a ser completado foi na partição 1, ou seja, foi um trabalho interativo. Mas tendo em conta tanto o tempo de execução total como o tempo médio de execução apoiam as nossas conclusões anteriores.

Distribuição de Trabalho

Após realização da entrega dos dataset's escolhidos pelo grupo, optamos por dividir os dataset's, foram escolhidos 3 dataset, portanto cada um ficou com um.

Apresentamos agora os responsáveis de cada Dataset:

- Bruno Sequeira - NASA
- Samuel Machado - SDSC 96
- Tomás Dias - HPC2N

NOTA:

Todos os alunos estiveram em contacto uns com os outros, estando dentro dos assuntos de todos os dataset's, o aluno Tomás Dias teve um papel crucial tendo ajudando os outros alunos a passar por vários problemas, ajudando a analisar melhor vários gráficos.