Metodologias Experimentais em Informática

Assignment 1

Exploratory Data Analysis

André Clemêncio – 2013152406
Joel Pires - 2014195242
Pedro Andrade – 2014225147

Universidade de Coimbra
Departamento de Engenharia Informática
Mestrado em Engenharia Informática

Índice

Introdução 3

Análise Experimental 4

UniLu - Gaia 4

CEA-Curie 12

Conclusão 17

Introdução

Este trabalho é feito no âmbito da cadeira de Métodos Experimentais em Informática de Mestrado em Engenharia Informática da Universidade de Coimbra. Pretende-se assim neste trabalho analisar dados exploratórios e fornecer uma visão mais condensada da informação contida nos dados. Para isso vamos fazer recurso de gráficos para conseguir fazer essa visualização mais inteligível dos dados.

Iremos analisar dois registos de atividade de sistemas de produção alojados no repositório disponibilizado pelo Laboratório de Sistemas Experimentais da Universidade Hebraica em Israel. Ao longo deste trabalho iremos procurar extrair o maior número de conclusões possíveis resultantes da análise destes conjuntos massivos de dados.

Análise Experimental

UniLu Gaia

Este log contém 3 meses de dados relativos ao grupo Gaia na *University of Luxemburg*. É utilizada essencialmente por biólogos com problemas de grande escala e também por pessoas de engenharia que trabalham com simulações físicas.

O grupo Gaia é um dos 4 grupos operados pela ULHPC (*University of Luxemburg HPC Center*). Possui 151 *nodes* com um total de 2004 *cores*.

De seguida serão apresentados vários gráficos que ilustram algumas características deste sistema:

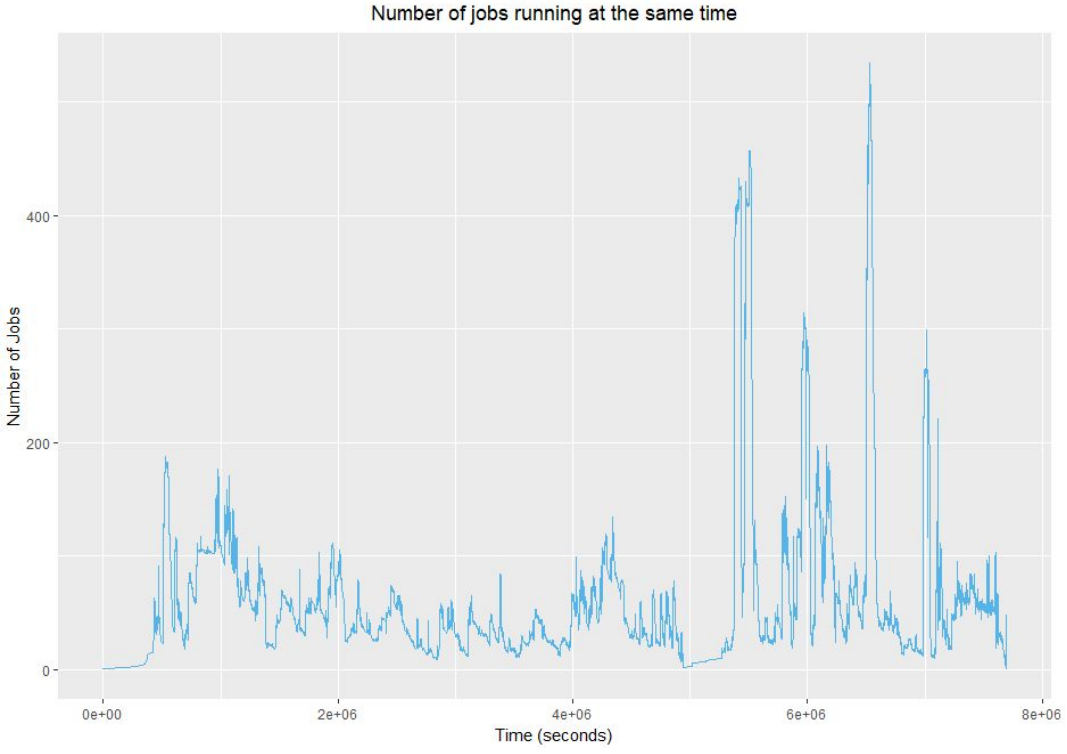

Fig 1. *Gráfico com o número de Jobs que decorreram em simultâneo ao longo do tempo (em segundos)*

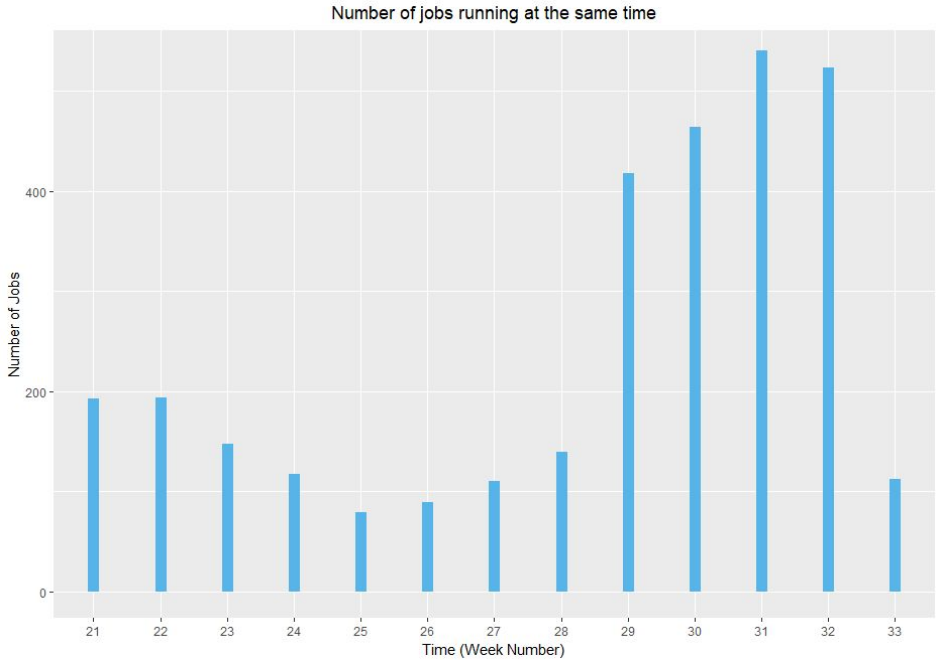

Fig 2. *Gráfico com o número de Jobs que decorreram em simultâneo ao longo das várias semanas.*

Através destes gráficos (**Fig 1 e 2**) conseguimos ter uma noção em que alturas há mais *jobs* a decorrer simultaneamente. Inicialmente verifica-se que o sistema de produção não se encontra muito sobrecarregado, nunca ultrapassando os 200 *jobs* simultâneos, no entanto, a partir da semana 29 verifica-se que o sistema de produção fica bem mais atarefado com picos que chegam a ultrapassar os 500 *jobs* simultâneos.

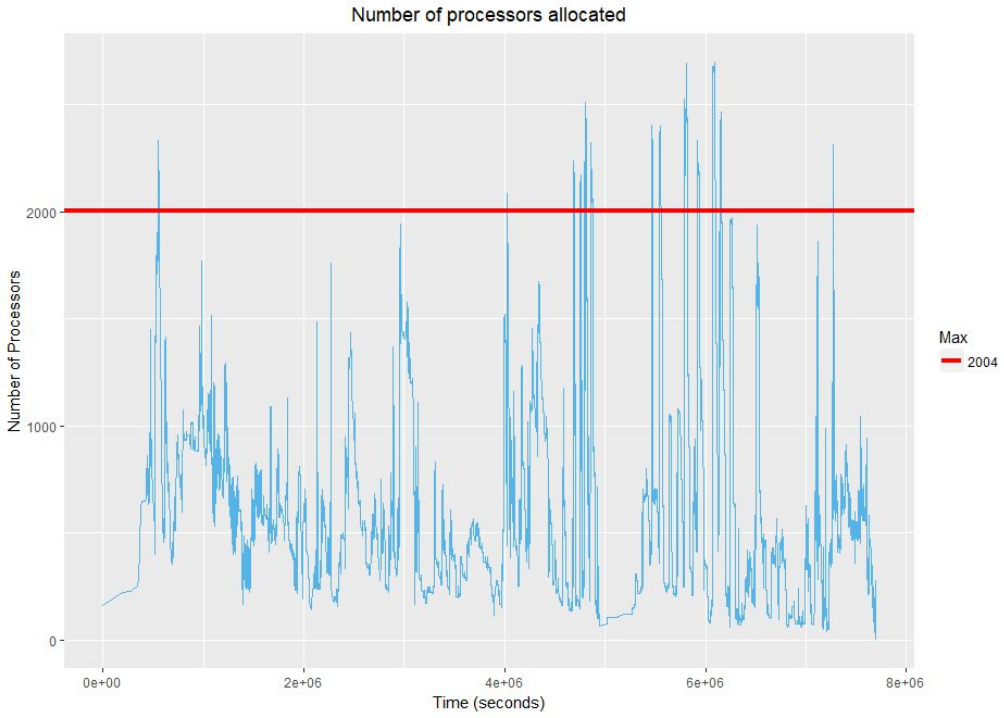

Fig 3. *Gráfico com o número de processadores alocados em cada momento.*

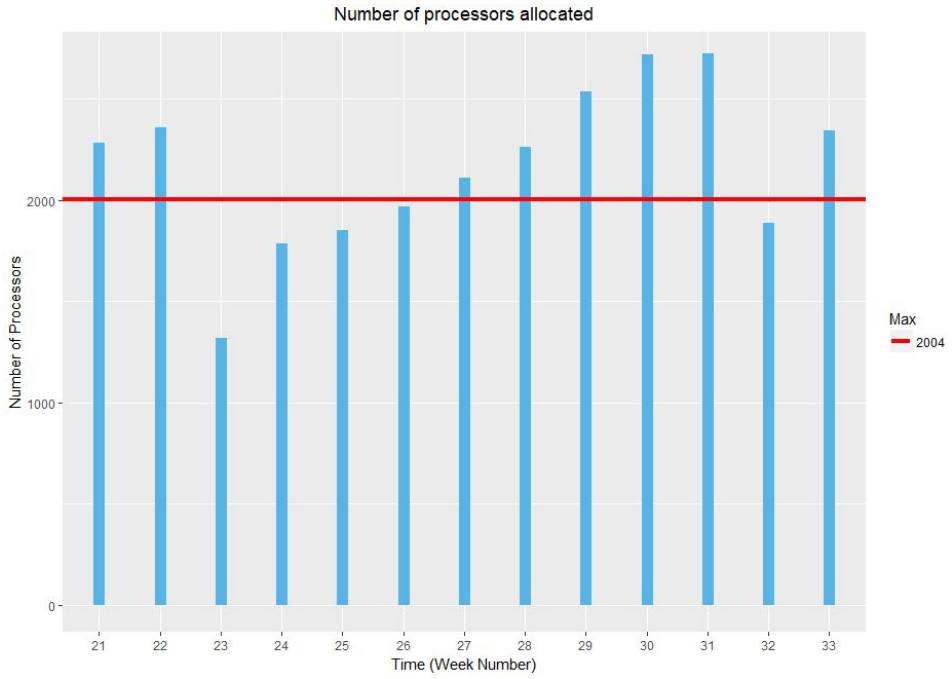

Fig 4. *Gráfico com o número de processadores alocados em cada semana.*

Nos dois gráficos anteriores (**Fig 3 e 4**) podemos ver que o número de processadores alocados ultrapassa em alguns momentos o número máximo de processadores que o sistema possui (que é 2004 e está representado nos gráficos pela linha horizontal vermelha). Se compararmos estes gráficos com os da **Fig 1 e 2**, podemos ver que as semanas com mais processadores alocados correspondem às semanas com mais *Jobs* a serem executados simultaneamente, tal como seria de esperar.

As primeiras duas semanas também ultrapassam o limite máximo de processadores, no entanto, nos gráficos das **Figuras 1 e 2** podemos verificar que não existem muitos *Jobs* a decorrer ao mesmo tempo. Isto pode sugerir que os *Jobs* das primeiras semanas precisaram de um número elevado de processadores para executarem.

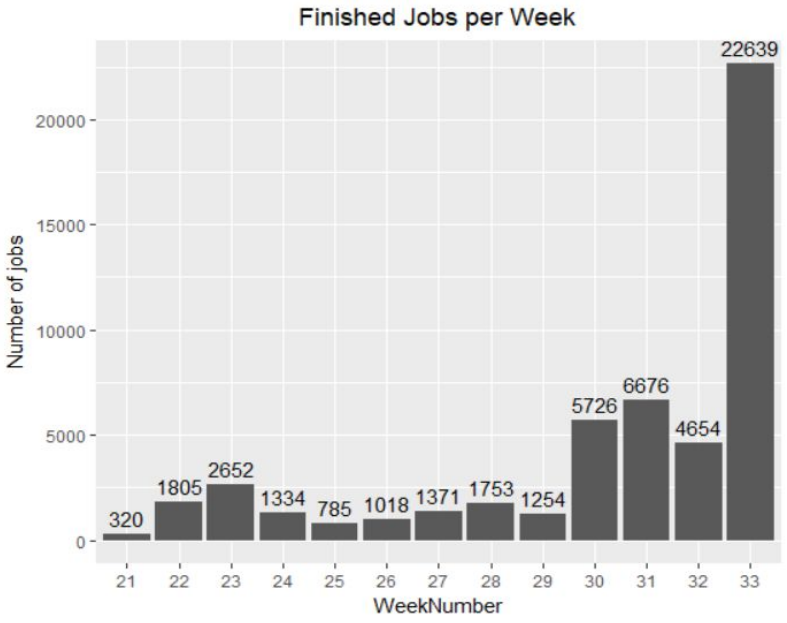

Fig 5. *Gráfico com o número de trabalhos finalizados por semana*

Este gráfico (**Fig. 5**) é importante para termos uma noção da quantidade de jobs que terminaram i.e, cujo running time cessou (podem ter sido concluídos ou não). Daqui podemos concluir em que semanas o nosso sistema de produção se encontrou mais atarefado. Foi na última semana que se verificou a presença de mais jobs a decorrer e, portanto, foi na 33ª semana que o sistema de produção registou mais atividade. Esses resultados são expectáveis; vejamos que os jobs demoram entre 1 a 3 semanas a concluir, logo é expectável que no início se iniciem vários e se concluam poucos, mas no fim se concluam os que cumulativamente se foram iniciando nas semanas anteriores.

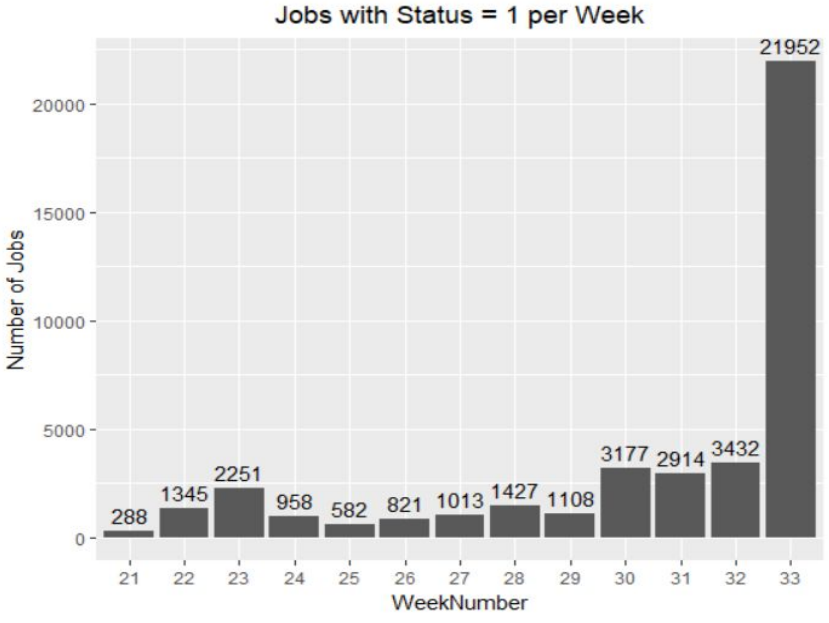

Fig 6. *Gráfico com o número de trabalhos finalizados por semana*

Quando os *jobs* têm status 1 é porque são *jobs* completos e bem sucedidos. Expectavelmente os números apresentados no gráfico (**Fig. 6**) de forma alguma ultrapassam os valores do gráfico anterior (que apresenta os jobs finalizados bem ou mal). Sabemos que foi na 33ª semana que se finalizou um maior número de jobs de forma bem sucedida em virtude também do facto de, tal como foi analisado no gráfico anterior, ter havido também mais jobs a decorrer durante essa semana. Portanto, tal facto não nos dá garantia que o sistema foi mais eficiente nessa semana.

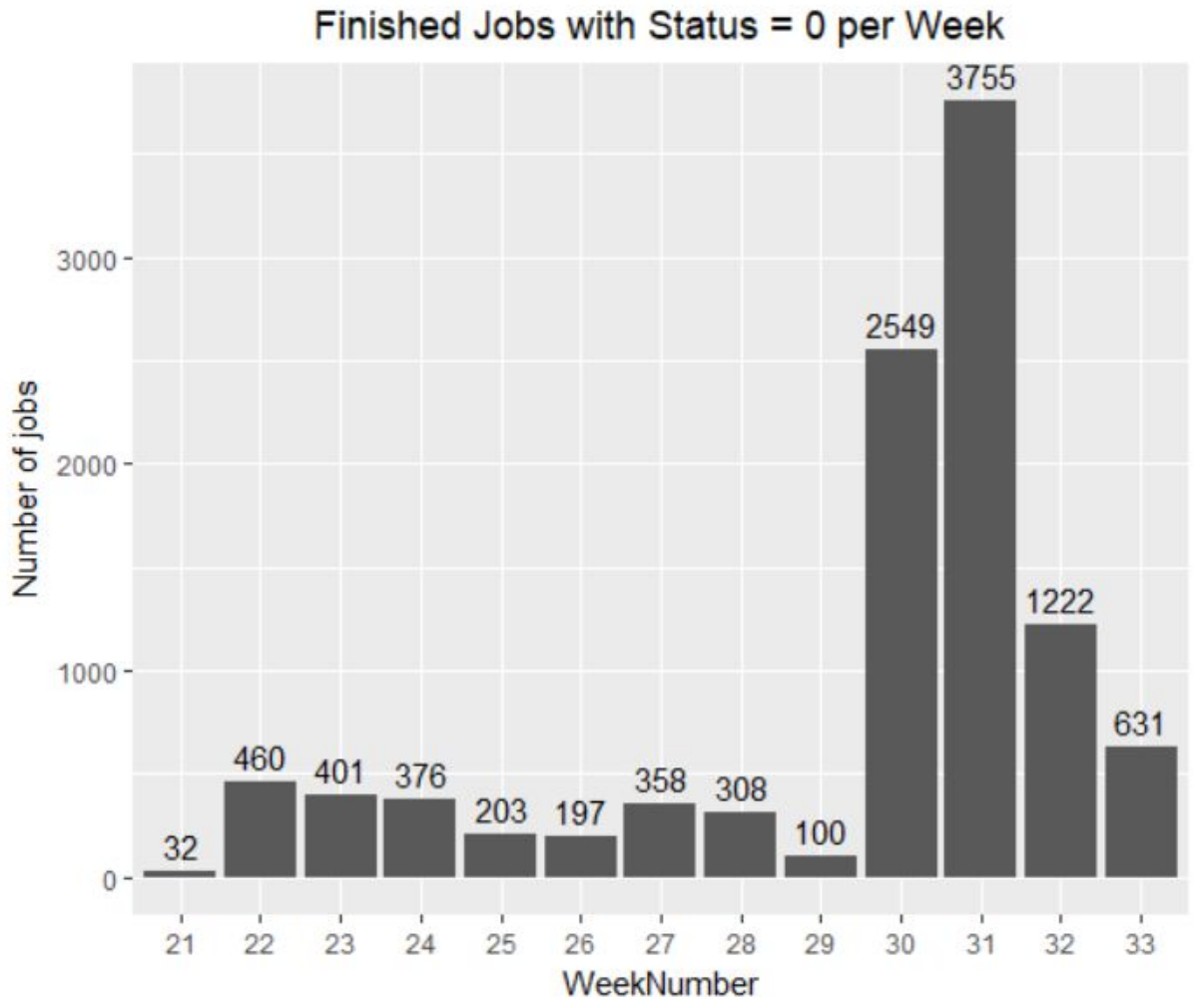

Fig. 7. *Gráfico que relaciona número de jobs falhados com a semana*

Quando os *jobs* têm status 0 é porque são *jobs* que por alguma razão falharam. Previsivelmente, foi na 21ª semana que se registou um menor número de jobs falhados dado ao pequeno número de jobs que estavam em atividade (**Fig. 7**). No entanto, curiosamente, o maior número de falhanços não se dá numa das semana em que o sistema de produção estava mais atarefado, o que pressagia uma extrema ineficiência do sistema durante esta semana. No entanto, analisemos o próximo gráfico (**Fig. 8**) para tirar conclusões mais sólidas sobre a eficiência do sistema.

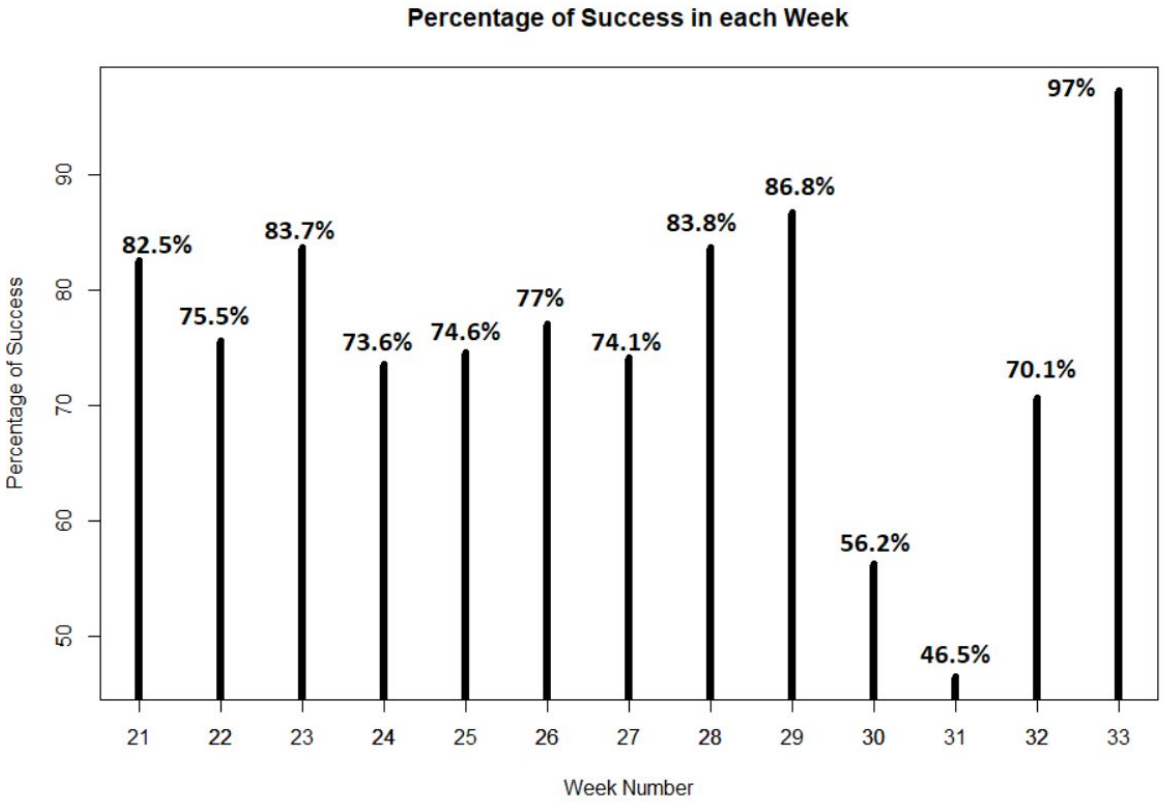

Fig. 8. *Gráfico com a percentagem de sucesso em cada semana*

Se tentarmos dividir o número de trabalhos bem sucedidos pelo total de trabalhos em execução durante determinadas semanas, obtemos o gráfico (**Fig. 8**) que aqui se apresenta. Podemos agora afirmar que o nosso sistema normalmente possui uma eficiência que normalmente não baixa para lá dos 70%. No entanto, na semana 30 e 31, por algum motivo o nosso sistema teve uma baixa performance, executando com sucesso apenas 56.2% e 46.5% dos trabalhos, respetivamente. Em contrapartida, verifica-se uma alta eficiência na última semana.

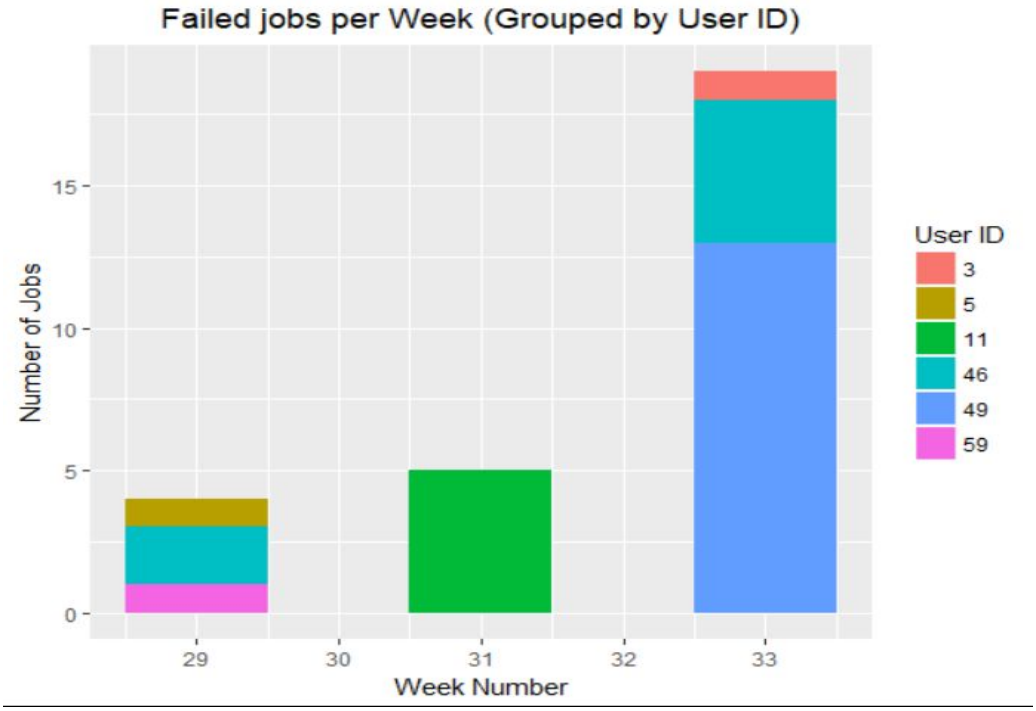

Fig. 9. *Gráfico com o número de jobs falhadas por semana (agrupadas por utilizador)*

Este gráfico (**Fig. 9**) ajuda-nos a ver quais os utilizadores é que estiveram associados a jobs cujo RunTime foi negativo (-1), isto é, nem chegaram a executar. Esses casos aconteceram apenas em 3 semanas (29ª, 31ª e 33ª) e aconteceu mais frequentemente na 33ª semana ao user nº 49.

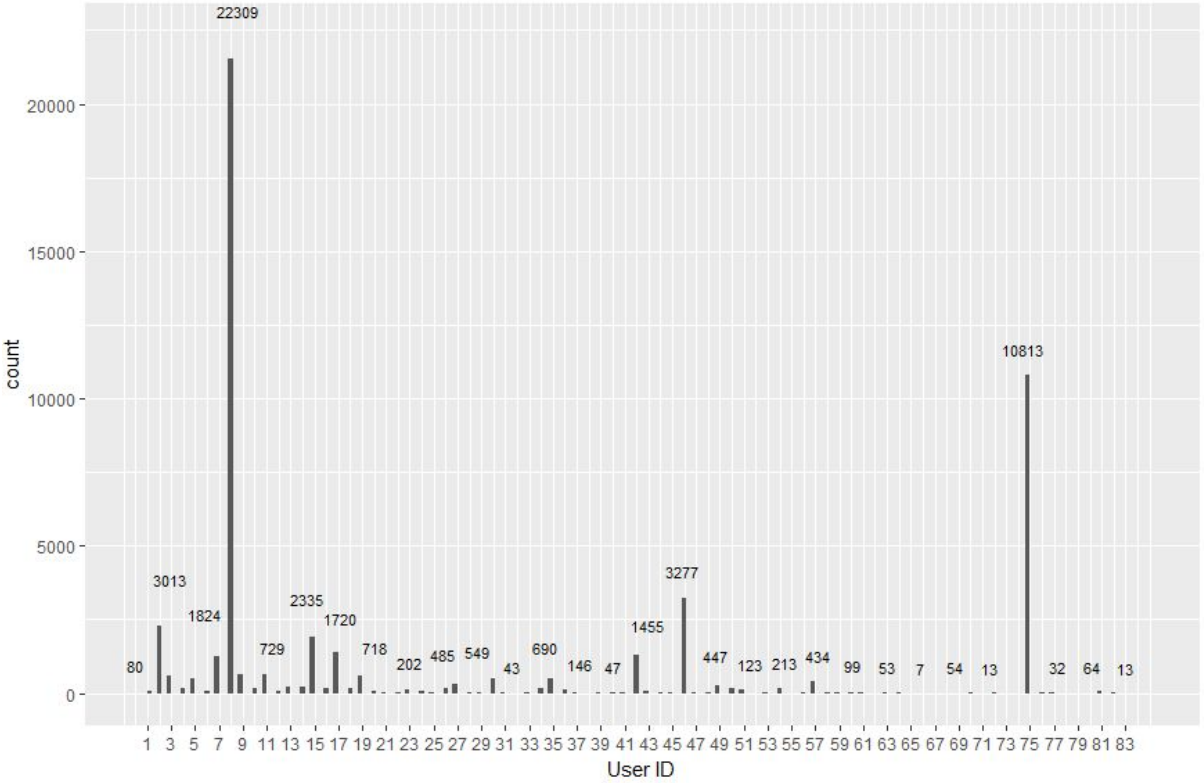

Fig. 10. *Gráfico com o número de jobs por utilizador*

Pela observação das barras deste gráfico (**Fig. 10**) podemos saber quais os utilizadores que estão associados a mais e menos jobs no nosso sistema. Por exemplo, os users que usaram mais o nosso sistema são sem dúvida o user 8 e 75 associados a 22309 e 10813 jobs respetivamente.

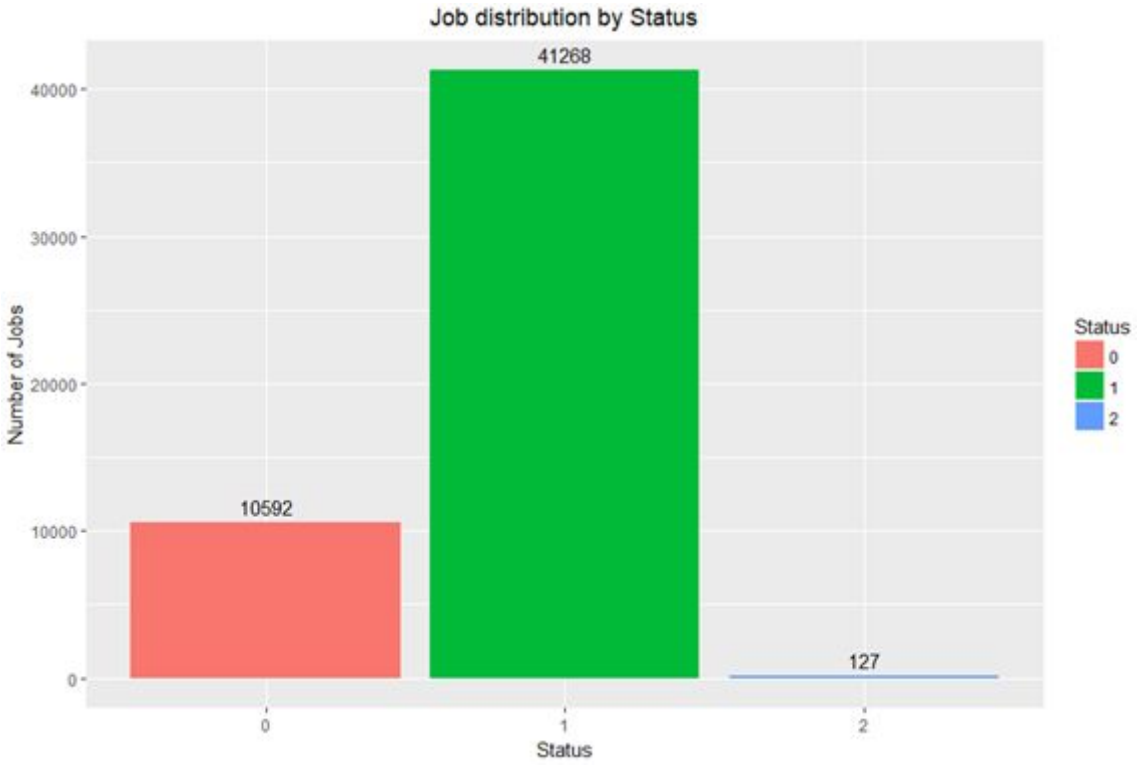

Fig. 11. *Gráfico com a distribuição das jobs pelo status*

Aqui (**Fig. 11**) podemos saber precisamente o número de jobs que terminaram de forma bem sucedida (status = 1), que falharam (status = 0) e que foram suspendidos (status = 2). Conseguimos concluir também que quantidade de trabalhos bem sucedidos foi maior que a de falhados que por sua vez foi maior que a de suspendidos. O número total de jobs foi assim: 41268+10592+127 = 51987 jobs. Se quisermos saber que percentagem estes valores assumem, basta ver que 79% (41268/51987) dos trabalhos foram bem sucedidos, 20% (10592/51987) deles foram mal sucedidos e 1% (127/51987) deles foram suspensos.

CEA-CURIE

Este log contém mais de 20 meses de dados de um supercomputador (“Curie”) que é operado pela CEA (uma organização de investigação tecnológica financiada pelo governo francês).

A informação provém de três partições com 11808 processadores Intel o que dá um total de 93312 cores. No entanto o sistema só esteve com a sua capacidade total nos últimos 10 meses (começou por ter só uma partição e posteriormente duas, o que perfazia uma capacidade a rondar o 1/6 da total).

Serão agora apresentados alguns gráficos que irão ilustrar as características do sistema em questão:

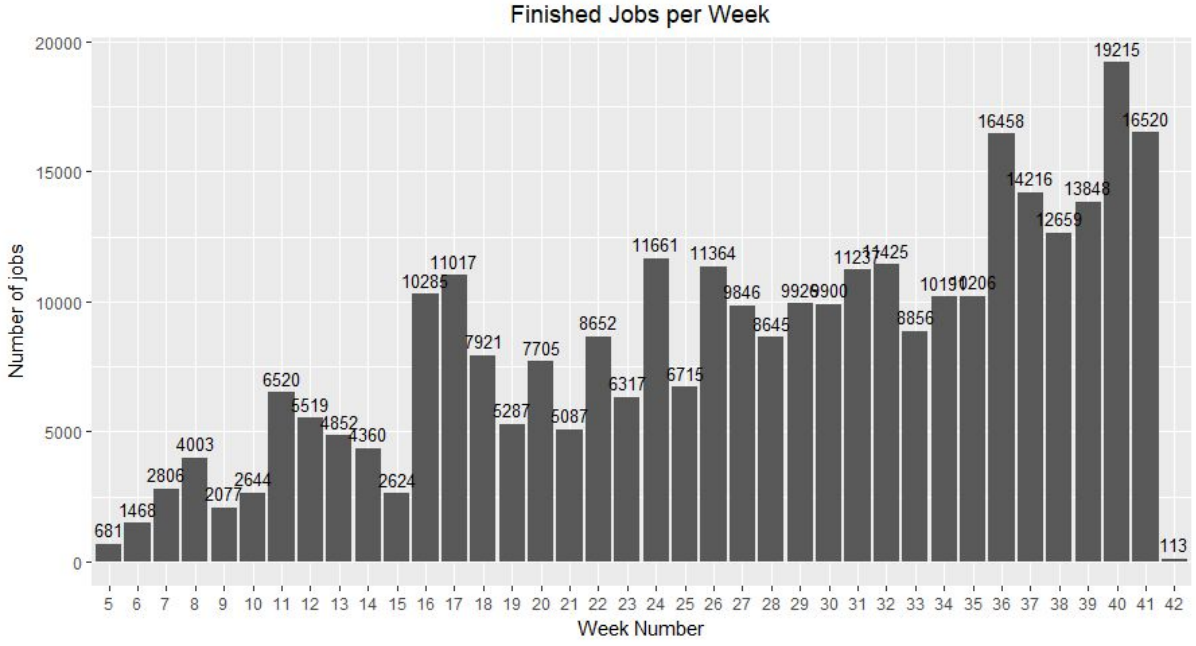

Fig. 12. *Gráfico com o número de jobs terminadas por semana*

Este gráfico (**Fig. 12**) mostra o número de jobs do sistema terminadas em cada semana podendo ser possível observar em que semanas o sistema esteve mais atarefado. Como podemos observar a tendência do gráfico será a de uma subida do número de jobs terminadas à medida que passam as semanas. Estes dados podem ser explicados pelo facto de que como a duração das jobs são de várias semanas, nas primeiras tenham terminado poucas acontecendo depois uma acumulação de jobs terminadas nas semanas seguintes (o que vai provocar o efeito visto no gráfico).

Praticamente todas as jobs terminaram até à semana número 41 restando apenas 113 que acabaram por concluir na semana número 42. A razão para o sucedido poderá ser o facto de nas semanas anteriores se terem iniciado muito poucas jobs que depois terminaram na última semana.

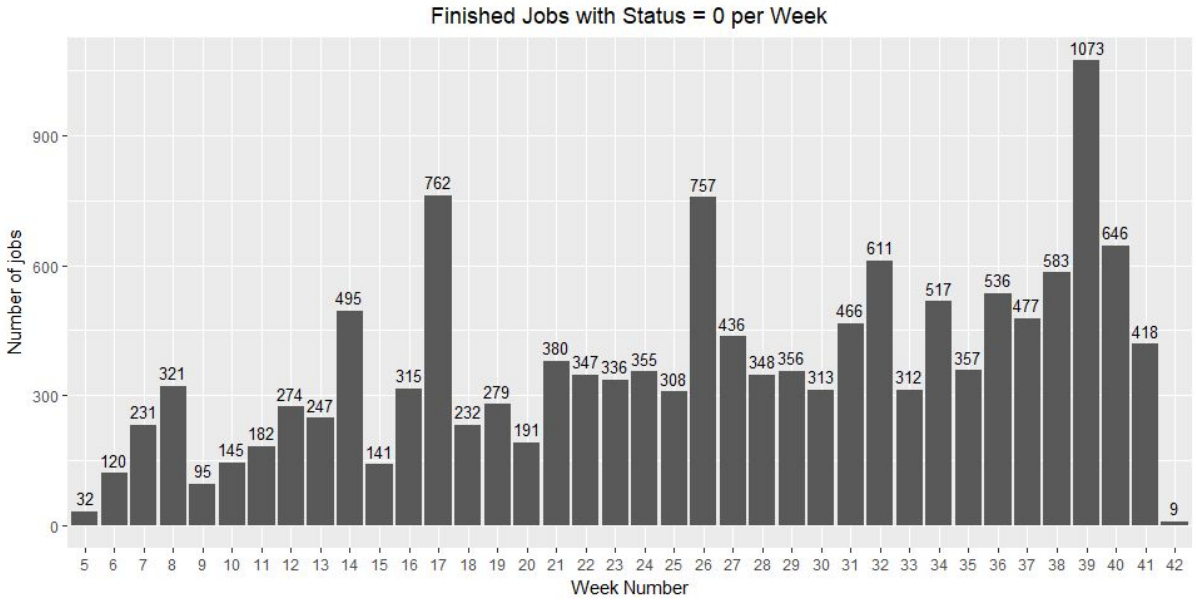

Fig. 13. *Gráfico com o número de jobs com status 0, por semana*

Os jobs que contêm o status com o valor igual a 0, são aqueles que por algum motivo falharam a sua execução.

Com a análise do gráfico (**Fig. 13**) podemos observar que as semanas em que falharam menos jobs foram aquelas em que o sistema teria um menor números de jobs ativas, sendo elas a semana 5 e a semana 42. Nas semanas em que o sistema estaria mais sobrecarregado com jobs em atividade foi quando se notou um maior crescimento de jobs com o status 0. Podemos então concluir que o número de jobs em atividade no sistema influencia o número de jobs que falham (o facto de o primeiro crescer influencia o crescimento do segundo).

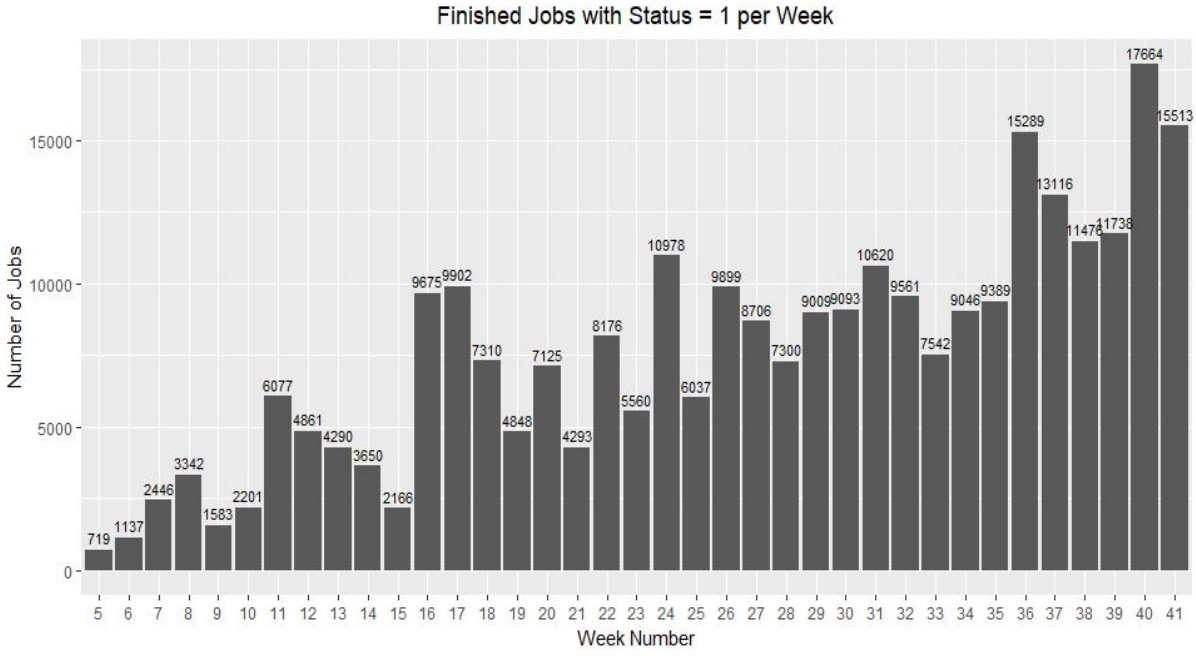

Fig. 14. *Gráfico com o número de jobs com status 1, por semana*

O gráfico (**Fig. 14**) em questão mostra todas as jobs com o status 1, ou seja, jobs que terminaram corretamente a sua execução. Podemos deduzir do gráfico que o número de jobs terminadas com sucesso tem uma tendência crescente o que tem a ver com o facto de a cada semana que passa, estarem mais jobs em atividade no sistema.

A semana com menos jobs terminadas com sucesso foi a semana 5 (menos jobs em atividades) e as semanas com mais jobs com status 1 serão as duas últimas (coincide com o maior número de jobs no sistema).

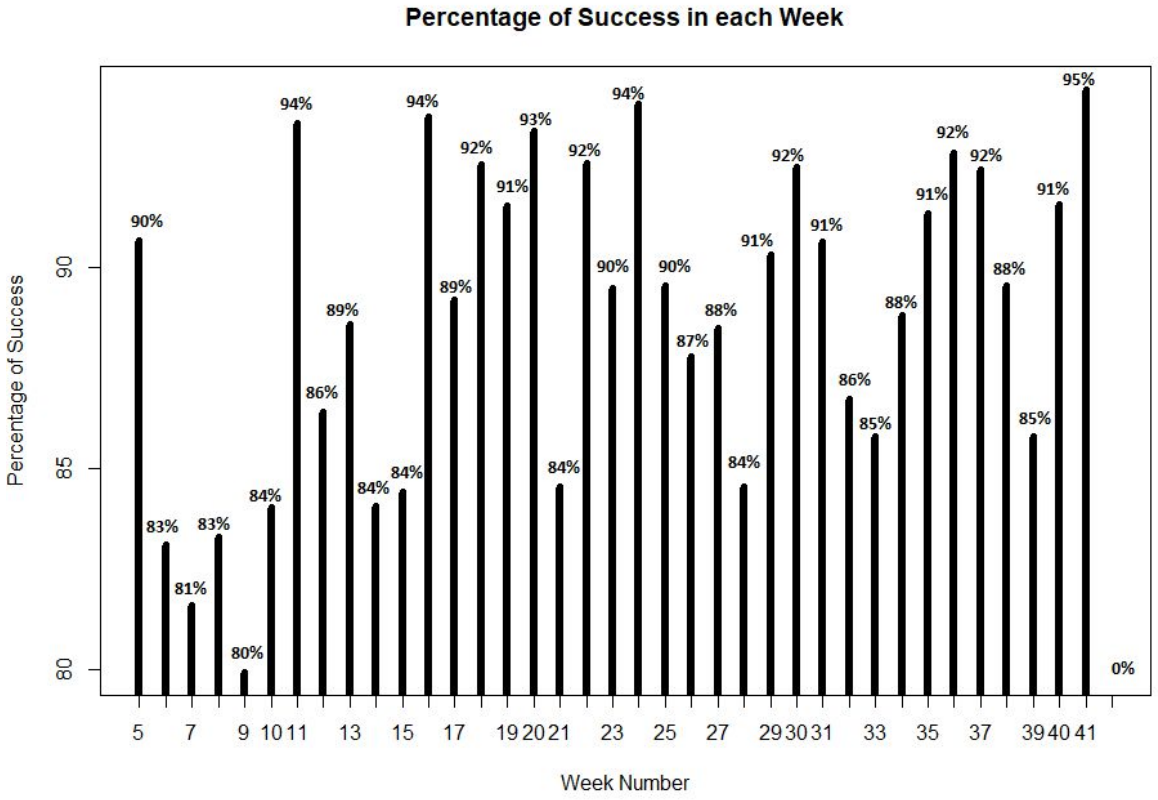

Fig. 15. *Gráfico com a percentagem de sucesso em cada semana*

Com este gráfico (**Fig.15**) obtemos a eficiência do sistema em cada semana (calculada através da divisão dos jobs com status igual a 1 por todos os jobs). Verificámos então que o sistema apresentou uma eficiência relativamente alta em praticamente todas as semanas, atingindo um pico de 95% na semana 41 e, além da última semana, o mais baixo que desceu foi para os 80% na semana número 9. Por algum motivo na semana 42 a eficiência foi de 0% (jobs terminadas nessa semana ou tinham o status 0 ou 5, não terminando nenhuma com o status 1).

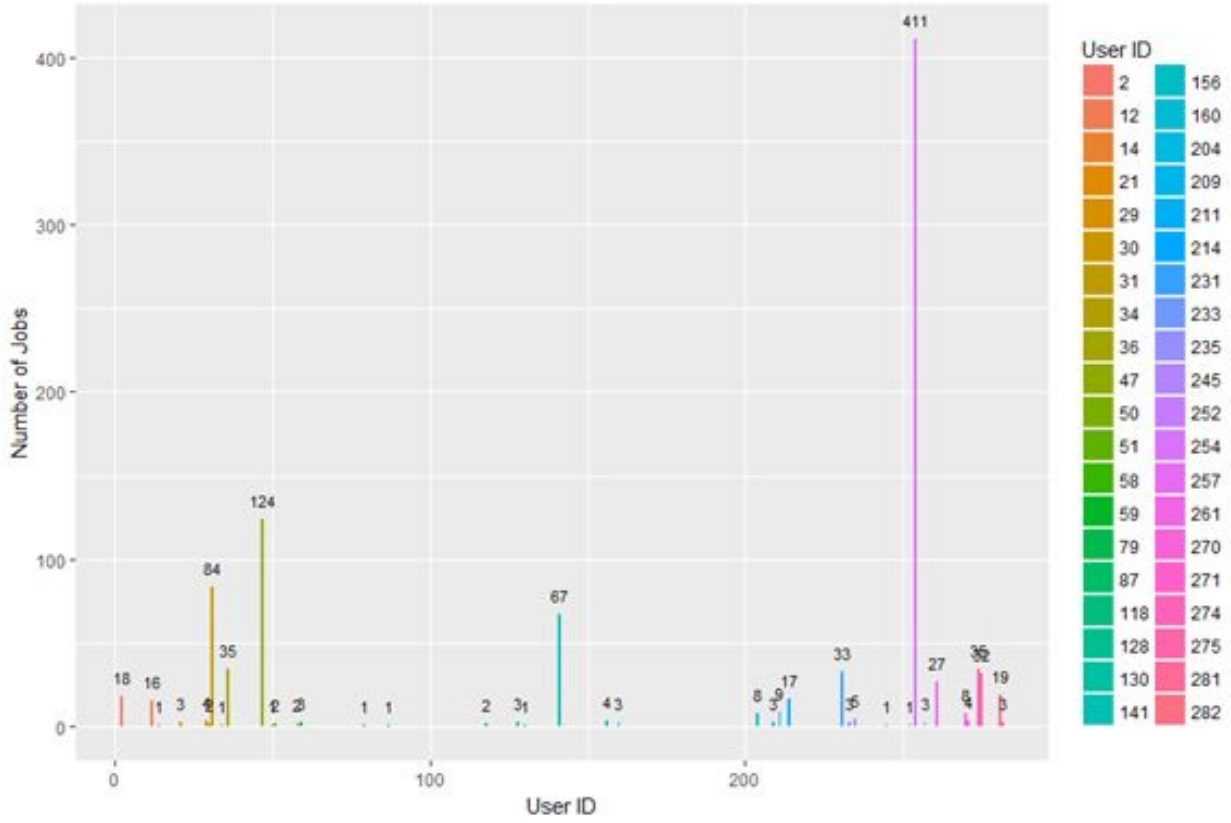

Fig. 16. *Gráfico com o número de jobs por cada utilizador*

Com a observação do presente gráfico (**Fig. 16**) podemos ver a distribuição do número de jobs por todos os utilizadores que utilizaram o sistema. O utilizador que, por larga margem, teve mais jobs no sistema foi o utilizador com o user id número 254. Existiu depois uma distribuição parecida pela maior parte dos restantes utilizadores sendo que alguns ainda se destacaram com maior número de jobs.

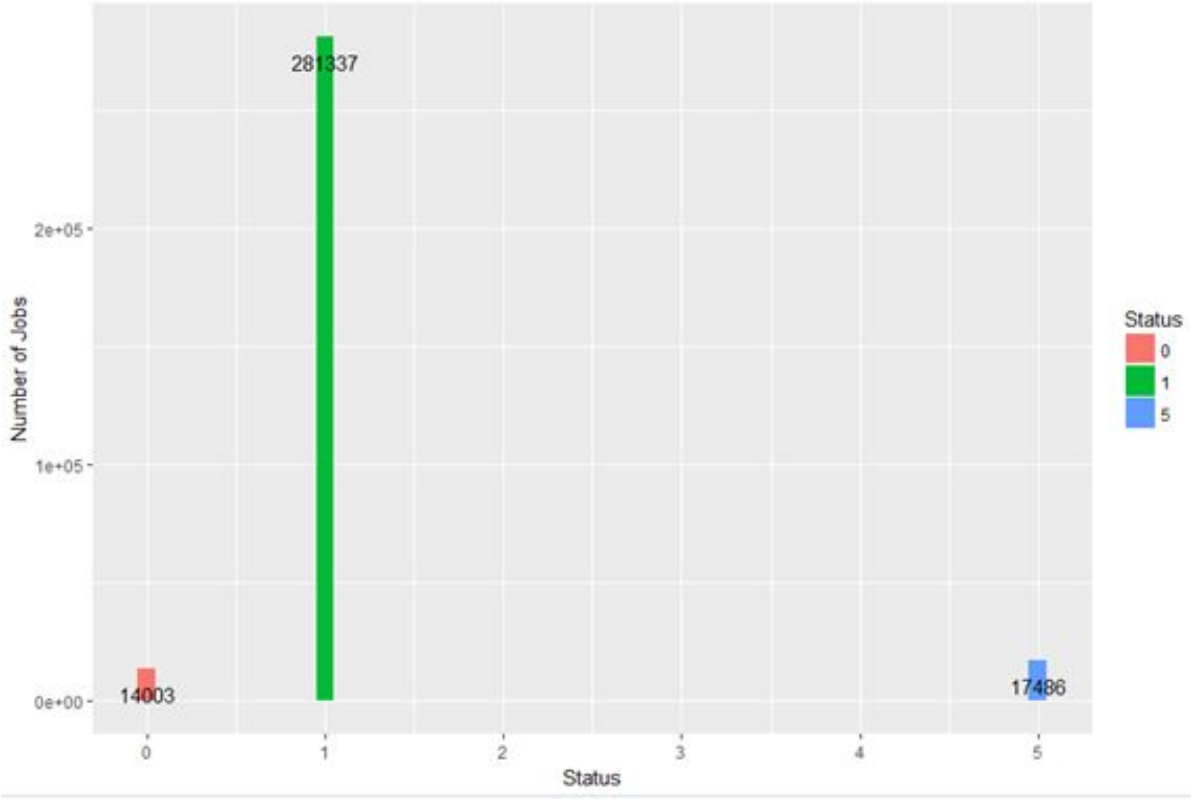

Fig. 17. *Gráfico com a distribuição das jobs pelo status*

Neste gráfico (**Fig. 17**) pode-se observar a distribuição de todos os jobs pelos status existentes. Como já foi explicado, jobs com status igual a 0 são jobs que falharam por alguma razão, jobs com status igual a 1 são aquelas que terminaram com sucesso e por último as que têm o status igual a 5 são as que foram canceladas (antes mesmo de começarem ou estando elas a correr).

Vemos então que a grande parte das jobs foram concluídas com sucesso sendo elas um total de 281337 (percentagem de aproximadamente 90%) o que poderá ser um bom indicativo da eficiência do sistema. Por sua vez 14003 jobs falharam (aproximadamente 4,5%) enquanto que 17486 foram canceladas (aproximadamente 5,5%).

Conclusão

Por vezes deparamo-nos com uma quantidade de dados massiva sobre determinado sistema. A mera observação não nos permite de todo concluir relações significativas. Aprendemos que o uso de recursos visuais como por exemplo gráficos é essencial para concluir tendências e reconhecer padrões nos nossos dados.

Neste trabalho, através do tratamento de variáveis como tempo de execução, número de processadores usados, entre outras, podemos tirar conclusões sobre a eficiência do nosso sistema, sobre taxas de utilização, distribuição do esforço do nosso sistema por utilizador, entre outras relações.

Deparamo-nos também muitas vezes em situações em que processávamos matricialmente informações de forma a obter a relação entre duas variáveis e esse processamento acabava por nos permitir voltar a analisar os dados de forma a estabelecer relações mais significativas entre os dados. Assim concluímos que, para estabelecer relações profundas e reconhecer padrões complexos entre os dados é necessário realizar processar os dados de forma a estabelecer primeiro relações menos significativas.