

# Replicação de Carvalho et al. 2012

Yure Pereira Campos<sup>1</sup>

<sup>1</sup>Universidade Federal de Campina Grande

yurecampos@copin.ufcg.edu.br

## 1. Estudo original

*Instruções: Descreva o estudo original, deixando claro qual a pergunta de pesquisa e qual a relevância dela. Também descreva em linhas gerais a metodologia utilizada. No processo, explique conceitos que sejam necessários para que seus colegas do curso entendam a pesquisa. Inclua no final da seção um link para o artigo original.*

Título da pesquisa original: "A User-Based Model of Grid Computing Workloads"

A pesquisa faz análise dos modelos de carga de trabalho atuais que avaliam os sistemas em *Grid* e propõe uma solução baseada em usuário.

### 1.1. Objetivo do estudo original

A pesquisa original faz uma análise dos tempos de chegada do trabalho do sistema para carga de trabalho baseado em usuário. O experimento tenta verificar se quando o número de usuários no sistema aumenta, aumenta também a carga do sistema. Propõem um modelo de carga de trabalho em *Grid* baseado em usuário que se apoia no agrupamento de usuários de acordo com seu comportamento no sistema e em suas aplicações.

### 1.2. Pergunta de pesquisa Original

É possível gerar um modelo de carga de trabalho em *Grid* baseado em usuário que seja melhor que os que utilizam uma abordagem de modelagem baseada em sistema?

### 1.3. Relevância da pesquisa original

Na avaliação de soluções para sistemas de grade (por exemplo, planejadores e políticas de gerenciamento de recursos), é desejável analisar como as soluções propostas se comportam em vários cenários representativos do sistema alvo. Normalmente, os diferentes cenários são obtidos variando a carga de trabalho do sistema (demanda) e/ou os recursos do sistema (fornecimento). Entretanto, na avaliação de soluções para computação em *Grid*, não há muito esforço em usar cargas de trabalho realistas para experimentos e, na maioria das vezes, as atividades e aplicativos dos usuários não são bem representados.

### 1.4. Metodologia utilizada na pesquisa original

A metodologia de modelagem de carga de trabalho em *Grid* proposta neste trabalho é dividida em duas partes: A primeira parte cobre a fase de extração do modelo, enquanto a segunda cobre a geração de carga de trabalho sintética que usa o modelo extraído como entrada.

O fluxograma da metodologia para modelagem de carga de trabalho pode ser visto na Figura 1.

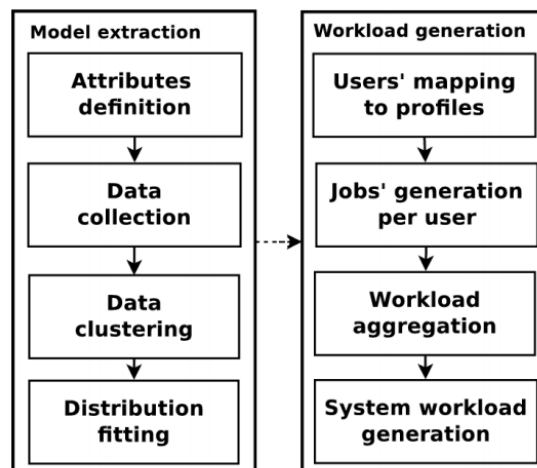


Figure 1. Methodology flow for workload modelling.

Figura 1. Fluxograma da metodologia para modelagem de carga de trabalho

### 1.5. Link do artigo original

<https://ieeexplore.ieee.org/document/6319153>

## 2. Objeto da replicação (ou reanálise)

*Instruções: Deixe claro aqui qual é a parte do estudo original que você reproduzirá/reanalisará. Essa parte é um resultado, então você deve colocar aqui:*

### 2.1. Figura com resultado

*Instruções: Uma figura, tabela ou texto que contém o resultado em questão*

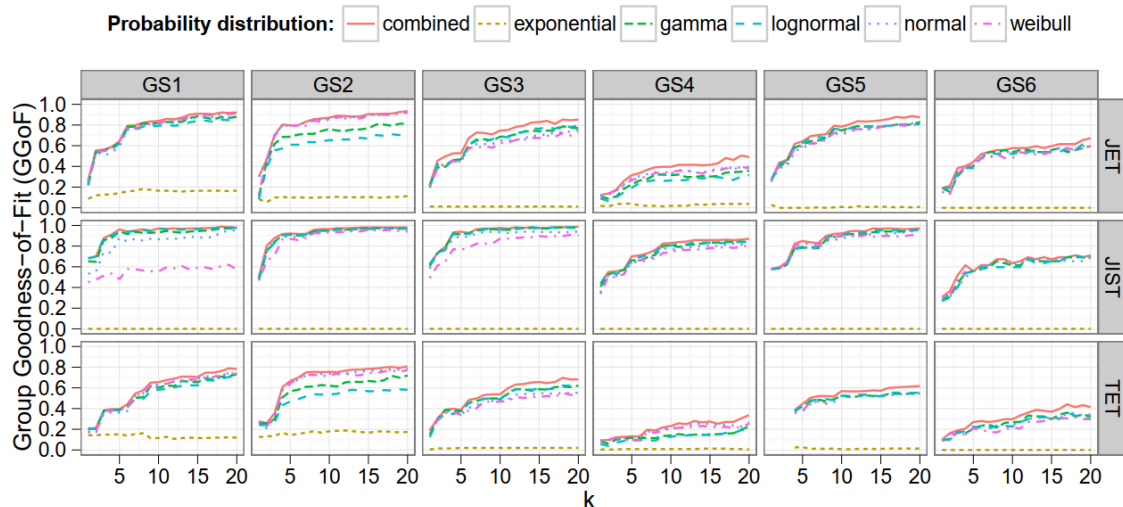


Figure 2. Group Goodness-of-Fit (GGoF) results for different fitting scenarios.

Figura 2. Resultados das avaliações dos modelos

## 2.2. Explicações

*Instruções: Texto com explicações para que consigamos entender o resultado (métricas, nomes, valores, o que for preciso para entendermos a ideia sem ler o artigo original)*

A modelagem foi aplicada a 6 rastreamentos de carga de trabalho (GS1–GS6) e para os 3 atributos de carga de trabalho (JIST, TET e JET). A Figura 2 mostra os resultados da métrica GGoF (Group Goodness-of-Fit) para diferentes valores do número de clusters ( $k$ ) e distribuições de probabilidade candidatas. Cada gráfico apresenta um cenário de rastreamento e atributo da carga de trabalho. As diferentes linhas nos gráficos representam os resultados para cada distribuição candidata listada.

## 2.3. Resultados

*Instruções: Um parágrafo dizendo claramente quais as conclusões que os autores do estudo original tiram do resultado (eles argumentam que há um efeito relevante? Que todas as alternativas do experimento são equivalentes? O que?)*

Pelo gráfico, vê-se que o GGoF tende a aumentar quando o número de clusters  $k$  aumenta. É um comportamento esperado, pois quanto maior o número de clusters, maior o número de distribuições utilizadas no modelo e menor o número de usuários por grupo. Nesse caso, como cada distribuição é ajustada para menos usuários, a probabilidade de obter goodness-of-fit bem-sucedido para todos os usuários é maior. Na maioria dos casos, as curvas GGoF têm uma alta taxa de aumento para os primeiros incrementos de  $k$ , mas em algum ponto as curvas tendem a atingir um platô onde aumentar  $k$  não aumenta o GGoF significativamente. Porém, usar uma combinação de distribuições como na abordagem combinada apresenta sempre os melhores resultados, conforme mostrado no gráfico. A abordagem combinada foi usada na aplicação do modelo para gerar cargas de trabalho sintéticas. Está fora do escopo do artigo propor uma técnica para encontrar o melhor número de clusters  $k$  para cada atributo e traço. Para simplificar, foi escolhido  $k = 5$  como o número de clusters a serem usados no aplicativo de modelo para todos os atributos de carga de trabalho. Como podemos ver na figura não há aumentos significativos no GGoF para o número de clusters maior que 5.

## 3. Metodologia original

### 3.1. Coleta de dados

*Descreva aqui os materiais que o artigo original usa e os métodos para produzir os dados que serão analisados. Exemplos de materiais são datasets, questionários, modelos de classificação, etc. Se seu projeto envolve um modelo de algum tipo, essa etapa abrange criar a classificação ou recomendação, mas não abrange calcular métricas como precision e recall; isso é objeto da próxima etapa.*

Os dados foram coletados de sistemas reais para extrair o modelo de carga de trabalho representativos. Foram processados os logs de sistemas reais, extrair os atributos necessários e armazenar os dados coletados em um formato apropriado. Os rastreamentos de carga de trabalho da grade e outros sistemas paralelos estão disponíveis em repositórios como o Grid Workloads Archive (GWA1) e o Parallel Workload Archive (PWA2). Foram usados os rastreamentos de carga de trabalho de *Grid* disponíveis no repositório GWA. Site: <http://gwa.ewi.tudelft.nl>

### 3.2. Dados gerados

*Inclua aqui uma tabela que mostra quais as colunas que você entende que são geradas na etapa anterior, com 3 linhas de exemplo preenchidas (valores inventados, mas com tipos corretos).*

Tabela com parâmetros do arquivo "wl\_fitdist\_best.txt" com as distribuições de probabilidade para geração das cargas sintéticas:

Param1	Param2	Cluster	distribution	k
Trace	stats.attr	Size	Fraction	
0.102892237715861	NA	8	exponential	10
gwa-t1	bot_user_iat	2	0.0114942528735632	
52.0417132717262	4.45758831824259	14	gamma	16
gwa-t4	bot_runtime_sum	2	0.0122699386503067	
4.48536537671296	8.36867132409201	7	weibull	17
gwa-t5	task_runtime	4	0.0176991150442478	

**Tabela 1. Parâmetros do arquivo de carga com distribuição de probabilidades**

### 3.3. Análise de dados

*Descreva como o artigo original analisa os dados coletados para chegar a um resultado e conclusões. Essa etapa inclui quais as métricas que são usadas (e você também usará), se há modelos de regressão que são criados e se/qual tipo de inferência estatística é feita.*

O artigo original propõe uma nova métrica proposta, chamada de Group Goodness-of-Fit (GGoF), ou Qualidade de Ajuste de Grupo, indica a fração de usuários que obtiveram ajustes de distribuição bem-sucedidos de acordo com um teste de adequação (ou seja, o resultado do teste foi superior a um determinado nível de significância).

## 4. Diferenças metodológicas com o estudo original

*Quais passos do que foi feito no estudo original você fará de maneira idêntica? Quais você fará diferente? Quais não é possível saber? Você deve comentar para cada uma das etapas abaixo se seu estudo é idêntico ou em que ele muda com relação ao original:*

#### *População estudada*

Serão usados os mesmos dados de entrada disponibilizados pelo autor, para tentar realizar a replicação (ou reanálise), pois o custo operacional para realizar estes experimentos, em termos de tempo, é muito alto.

#### *Pergunta de pesquisa*

A mesma inicial, porém, ainda não sabemos se haverá tempo hábil para concluir e obter todos os resultados esperados, por conta do elevado tempo de processamento de cada etapa.

#### *Metodologia de coleta de dados*

A mesma, pois os dados já foram coletados e estão disponíveis.

### ***Dados gerados***

Espera-se gerar dados idênticos e a partir destes fazer uma reanálise, caso sejam congruentes.

### ***Metodologia de análise de dados***

Usaremos a mesma metodologia para geração dos dados e compararemos com os originais. Caso haja alguma divergência, mediremos as diferenças e mostraremos graficamente.

### ***Código da análise de dados***

Serão usados os mesmos códigos disponíveis, mas também outros disponíveis no repositório GitHub.

## **5. Resultado da replicação/reanálise**

***Descreva aqui os seus resultados. Atenção para a apresentação com visualizações e com clareza e precisão no texto. Comente suas conclusões a partir de seus resultados.***

Para a replicação, construímos um ambiente em Linux para poder rodar as instruções DoMC (de multiprocessamento), pois tais bibliotecas não estão disponíveis para ambiente Windows. No processamento de extração do modelo, foram utilizados os mesmos dados de entrada (gwa-traces-with-botid.zip), indicados e disponíveis em:

<https://github.com/YureCampos/grid-workload-model/tree/main/data/>

O tempo de processamento do código principal foi de 21 horas utilizando uma máquina virtual Linux com 6 processadores e 12GB de memória RAM. Foram gerados diversos arquivos intermediários, disponíveis em:

<https://github.com/YureCampos/grid-workload-model/tree/main/results/>

O arquivo "wl\_fitdist\_all.txt" que contém o cálculo da Qualidade de Ajuste de Grupo - GGoF (Group Goodness-of-Fit), com a seguinte estrutura:

distribution	k	Trace	stats.attr	ratio
exponential	1	gwa-t1	bot_user_iat	0.0114942528735632
combined	5	gwa-t10	bot_runtime_sum	0.8333333333333333
gamma	20	gwa-t2	task_runtime	0.747787610619469
lognormal	11	gwa-t4	bot_user_iat	0.735632183908046
normal	7	gwa-t3	bot_runtime_sum	0.878787878787879
weibull	3	gwa-t11	task_runtime	0.639810426540284

**Tabela 2. Parâmetros do arquivo com o cálculo GGoF**

Percebemos uma diferença nos nomes das variáveis e rótulos usados. Também os nomes dos servidores HPC utilizados estavam com siglas diferentes, por isso, ajustamos os mesmos para similaridade ao exposto no artigo.

Os 6 rastreamentos (traces) de carga de trabalho utilizados são:

Indicação	variável "Trace"	Servidor HPC (local, tipo)
GS1	gwa-t1	DAS-2 (Holanda, acadêmico)
GS2	gwa-t2	Grid'5000 (França, acadêmico)
GS3	gwa-t3	NorduGrid (Europa, acadêmico e produção)
GS4	gwa-t4	AuverGrid (França, produção)
GS5	gwa-t10	SHARCNET (Canadá, produção)
GS6	gwa-t11	EGEE/LCG (Europa, produção)

**Tabela 3. Indicação dos Traces de carga de trabalho utilizados**

Os 3 atributos de carga de trabalho (JET, JIST e TET) são mapeados pela variável "stats.attr" da seguinte forma:

Sigla Atributo	variável "stats.attr"	Significado
JET	bot_runtime_sum	Tempo de Execução do Job
JIST	bot_user_iat	Tempo entre envio de Jobs
TET	task_runtime	Tempo de Execução da Tarefa

**Tabela 4. Siglas, atributos e significados**

Reconstruindo o gráfico a partir da extração do modelo, temos:

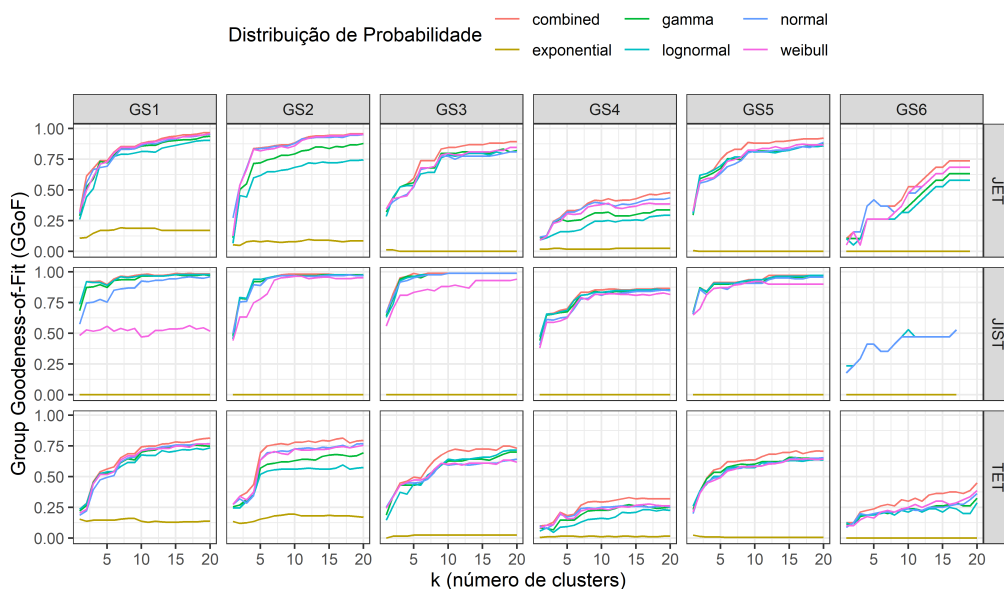


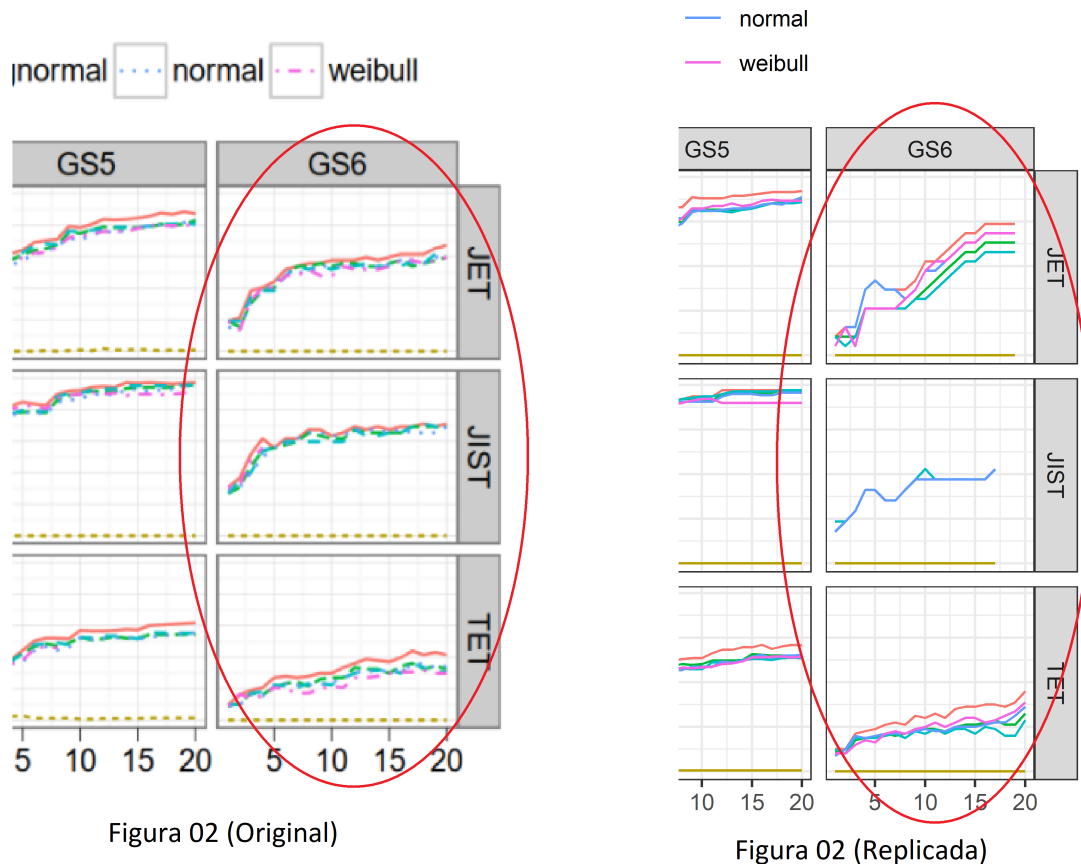
Figura 2: Resultados do GGoF para diferentes cenários de ajuste.

**Figura 3. Resultados das avaliações dos modelos**

## 6. Replicação vs. estudo original

*Compare seus resultados e os do estudo original. Eles se contradizem? Concordam? Como as suas conclusões se comparam com as do estudo original? Quais são possíveis razões para diferenças encontradas?*

Analisando os gráficos e comparando com o estudo original, encontramos resultados muito similares em quase todos os modelos de distribuição em cada HPC utilizado, porém identificamos uma incongruência no Workload GS6, principalmente no atributo Job Inter-Submission Time (JIST), tempo entre envios de Jobs.



**Figura 4. Destaque das diferenças entre o Gráfico original e o replicado**

Tentamos investigar as causas para os resultados diferentes, mas o arquivo original "wl\_fitdist\_all.txt", objeto da divergência, não foi disponibilizado pelo autor. Este é gerado a partir dos arquivos intermediários listados abaixo:

```
(kali@kali)-[ /Documents/GitHub/grid-workload-model/results]
$ ls -ltrh
total 34M
-rw-r--r-- 1 kali kali 533K Jul 7 00:17 clustdist_gwa-t1_bot_user_iat_.txt
-rw-r--r-- 1 kali kali 1.5M Jul 7 01:33 clustgof_bot_user_iat_gwa-t1_.txt
-rw-r--r-- 1 kali kali 466K Jul 7 01:33 clustdist_gwa-t2_bot_user_iat_.txt
-rw-r--r-- 1 kali kali 1.5M Jul 7 02:38 clustgof_bot_user_iat_gwa-t2_.txt
```



-rw-r--r-- 1 kali kali 124K Jul 7 02:38 clustdist\_gwa-t3\_bot\_user\_iat\_.txt  
-rw-r--r-- 1 kali kali 722K Jul 7 03:14 clustgof\_bot\_user\_iat\_gwa-t3\_.txt  
-rw-r--r-- 1 kali kali 466K Jul 7 03:15 clustdist\_gwa-t4\_bot\_user\_iat\_.txt  
-rw-r--r-- 1 kali kali 1.5M Jul 7 04:30 clustgof\_bot\_user\_iat\_gwa-t4\_.txt  
-rw-r--r-- 1 kali kali 343K Jul 7 04:31 clustdist\_gwa-t10\_bot\_user\_iat\_.txt  
-rw-r--r-- 1 kali kali 1.2M Jul 7 05:28 clustgof\_bot\_user\_iat\_gwa-t10\_.txt  
-rw-r--r-- 1 kali kali 4.8K Jul 7 05:28 clustdist\_gwa-t11\_bot\_user\_iat\_.txt  
-rw-r--r-- 1 kali kali 102K Jul 7 05:34 clustgof\_bot\_user\_iat\_gwa-t11\_.txt  
-rw-r--r-- 1 kali kali 539K Jul 7 05:35 clustdist\_gwa-t1\_bot\_runtime\_sum\_.txt  
-rw-r--r-- 1 kali kali 1.6M Jul 7 07:00 clustgof\_bot\_runtime\_sum\_gwa-t1\_.txt  
-rw-r--r-- 1 kali kali 476K Jul 7 07:01 clustdist\_gwa-t2\_bot\_runtime\_sum\_.txt  
-rw-r--r-- 1 kali kali 1.5M Jul 7 08:14 clustgof\_bot\_runtime\_sum\_gwa-t2\_.txt  
-rw-r--r-- 1 kali kali 123K Jul 7 08:14 clustdist\_gwa-t3\_bot\_runtime\_sum\_.txt  
-rw-r--r-- 1 kali kali 751K Jul 7 08:52 clustgof\_bot\_runtime\_sum\_gwa-t3\_.txt  
-rw-r--r-- 1 kali kali 463K Jul 7 08:53 clustdist\_gwa-t4\_bot\_runtime\_sum\_.txt  
-rw-r--r-- 1 kali kali 1.5M Jul 7 10:08 clustgof\_bot\_runtime\_sum\_gwa-t4\_.txt  
-rw-r--r-- 1 kali kali 351K Jul 7 10:09 clustdist\_gwa-t10\_bot\_runtime\_sum\_.txt  
-rw-r--r-- 1 kali kali 1.3M Jul 7 11:13 clustgof\_bot\_runtime\_sum\_gwa-t10\_.txt  
-rw-r--r-- 1 kali kali 5.9K Jul 7 11:13 clustdist\_gwa-t11\_bot\_runtime\_sum\_.txt  
-rw-r--r-- 1 kali kali 163K Jul 7 11:21 clustgof\_bot\_runtime\_sum\_gwa-t11\_.txt  
-rw-r--r-- 1 kali kali 891K Jul 7 11:24 clustdist\_gwa-t1\_task\_runtime\_.txt  
-rw-r--r-- 1 kali kali 2.0M Jul 7 13:19 clustgof\_task\_runtime\_gwa-t1\_.txt  
-rw-r--r-- 1 kali kali 698K Jul 7 13:21 clustdist\_gwa-t2\_task\_runtime\_.txt  
-rw-r--r-- 1 kali kali 1.8M Jul 7 15:02 clustgof\_task\_runtime\_gwa-t2\_.txt  
-rw-r--r-- 1 kali kali 264K Jul 7 15:03 clustdist\_gwa-t3\_task\_runtime\_.txt  
-rw-r--r-- 1 kali kali 1.1M Jul 7 16:11 clustgof\_task\_runtime\_gwa-t3\_.txt  
-rw-r--r-- 1 kali kali 636K Jul 7 16:12 clustdist\_gwa-t4\_task\_runtime\_.txt  
-rw-r--r-- 1 kali kali 1.7M Jul 7 17:48 clustgof\_task\_runtime\_gwa-t4\_.txt  
-rw-r--r-- 1 kali kali 775K Jul 7 17:51 clustdist\_gwa-t10\_task\_runtime\_.txt  
-rw-r--r-- 1 kali kali 1.9M Jul 7 19:38 clustgof\_task\_runtime\_gwa-t10\_.txt  
-rw-r--r-- 1 kali kali 111K Jul 7 19:38 clustdist\_gwa-t11\_task\_runtime\_.txt  
-rw-r--r-- 1 kali kali 703K Jul 7 20:17 clustgof\_task\_runtime\_gwa-t11\_.txt  
-rw-r--r-- 1 kali kali 4.4M Jul 7 20:17 wl\_fitdist\_best\_peruser.txt  
-rw-r--r-- 1 kali kali 308K Jul 7 20:17 wl\_fitdist\_best.txt  
-rw-r--r-- 1 kali kali 7.6K Jul 7 20:17 wl\_fitdist\_best\_k5.txt  
-rw-r--r-- 1 kali kali 99K Jul 7 20:17 wl\_fitdist\_all.txt

O único disponibilizado pelo autor é o "wl\_fitdist\_best.txt" que é usado para gerar as cargas sintéticas, mas não para análise gráfica dos resultados, como exposto no artigo.

Então, não foi possível comparar os dados originais com os gerados na replicação, mas apenas a comparação visual dos gráficos GGoF.

## 7. Bônus: Material para replicação

***[Link para github com os dados, código e instruções para replicar o seu experimento.](#)***

Os dados, códigos e instruções desta replicação estão disponíveis em:



<https://github.com/YureCampos/grid-workload-model>

O conteúdo de cada subdiretório é o seguinte:

Subdiretórios	Conteúdo
/data	Arquivos de entrada
/paper	Paper original e outros documentos relacionados
/reports	Códigos em R, reports, figuras geradas
/results	Arquivos de saída gerados

**Tabela 5. Conteúdo do repositório**