

# Modelagem da Carga de Trabalho de Grades Computacionais Baseada no Comportamento dos Usuários

Marcus Carvalho, Francisco Brasileiro

<sup>1</sup>Laboratório de Sistemas Distribuídos (LSD)  
Departamento de Sistemas e Computação (DSC)  
Universidade Federal de Campina Grande (UFCG)

marcus@lsd.ufcg.edu.br, fubica@dsc.ufcg.edu.br

**Abstract.** *Grid computing is a large scale federated infrastructure where users execute several types of applications on different submission rates. On the evaluation of solutions for grids, there are not much effort on using realistic workloads for experiments, and most of the time users' behaviours and applications are not well represented. In this work, we propose a grid workload model which is based on clustering its users according to their grid usage and their applications. The results show that according to the metric proposed the model quality increases when using clustering and extracting models for the group of users with similar profiles. There is a reasonable improvement even using a few clusters, reducing the workload model complexity while keeping it realistic.*

**Resumo.** *Grades computacionais são infraestruturas federadas de larga escala, geralmente utilizadas por usuários que possuem diversos tipos de aplicações e apresentam diferentes comportamentos na submissão das mesmas. Para avaliar soluções para estes sistemas, comumente são realizados experimentos alimentados por cargas de trabalho pouco representativas, pois não caracterizam adequadamente o comportamento de seus usuários. Neste trabalho, propõe-se um modelo de carga de trabalho para grades computacionais que se baseia no agrupamento de usuários de acordo com seus perfis de uso e de suas aplicações. Os resultados mostram que há um aumento na qualidade da modelagem quando os usuários são agrupados de acordo com seus perfis e é extraído um modelo para cada grupo. Verifica-se que mesmo com uma quantidade pequena de grupos é possível ter uma modelagem satisfatória, o que reduz a complexidade do modelo e mantém sua representatividade.*

## 1. Introdução

Grades computacionais são infraestruturas de computação de larga escala que permitem o compartilhamento de recursos entre entidades de diferentes domínios administrativos e têm como característica a heterogeneidade dos recursos que as compõem. Além disso, o fato de ser uma infraestrutura federada faz com que existam usuários que possuem diversos tipos de aplicações e diferentes comportamentos na submissão das mesmas.

Para avaliar soluções para estes sistemas, como por exemplo escalonadores, políticas de gerenciamento ou modelos de predição, é preciso analisar como cada solução proposta se comporta em cenários representativos do sistema alvo. A exploração desses

cenários geralmente é feita a partir da variação na carga de trabalho do sistema (demanda) e/ou da variação dos recursos que compõem a grade (oferta).

Em geral, três abordagens são usadas na geração de cenários de carga de trabalho na avaliação de ambientes de grades. A primeira consiste em realizar uma varredura de parâmetros em atributos de carga de trabalho relacionados às características de aplicações e da taxa de submissões ao sistema [Anglano et al. 2006, Bittencourt et al. 2006]. O problema é que, geralmente, as distribuições de probabilidade e os valores dos parâmetros usados não são bem justificados ou validados com dados reais, sem haver então evidências de quais dos cenários gerados são mais realistas. Por exemplo, um erro comum na avaliação de sistemas paralelos é assumir que a taxa de chegada de tarefas segue a distribuição Poisson, o que foi mostrado ser incorreto [Feitelson 2011]. Na segunda, são usados rastros de carga de trabalho extraídos de grades computacionais ou outros sistemas de computação paralela em produção [Ponciano et al. 2010, Carvalho et al. 2010]. O problema em usar a reprodução direta dos rastros é que a avaliação fica limitada a situações históricas de demanda de sistemas específicos, não havendo uma exploração maior de cenários diferentes. A terceira abordagem consiste no uso de cargas de trabalho sintéticas geradas a partir de modelos que descrevem a carga do sistema. O problema dessa abordagem é a dependência de rastros representativos do sistema em questão para que o modelo seja extraído. Porém, uma grande vantagem dessa abordagem é a facilidade em variar a carga de trabalho com a mudança dos valores dos parâmetros do modelo, ainda mantendo as propriedades gerais do sistema, o que ajuda na exploração de diversos cenários representativos para avaliação [Lublin and Feitelson 2003].

Os modelos de carga de trabalho podem ser divididos em duas classes: os descritivos e os hierárquicos. Na primeira classe, a carga é modelada como uma caixa preta e é representada de forma agregada de acordo com as estatísticas de atributos como a taxa de chegada de *jobs* e o tempo de execução no sistema. Na segunda classe, é realizada a modelagem de como a carga é gerada na sua origem por cada usuário do sistema, de modo que a agregação das cargas individuais resulte na carga total do sistema. Entre as vantagens do modelo hierárquico, destacam-se a presença de informações na carga sintética sobre os usuários que a geraram, uma melhor representação do comportamento e das aplicações de cada usuário do sistema, permitindo uma análise de experimentos de acordo com os perfis dos usuários, além de permitir que a carga total do sistema seja facilmente alterada, porém mantendo as características dos seus usuários [Feitelson 2011].

Neste trabalho, é proposto um modelo hierárquico da carga de trabalho de grades computacionais, que descreve o comportamento dos usuários de acordo com seus perfis. As contribuições deste trabalho são as seguintes:

- Apresentação de um novo modelo de carga de trabalho para grades computacionais, baseado no comportamento dos usuários agrupados por seus perfis.
- Apresentação de uma metodologia para a modelagem da carga de trabalho e disponibilização de ferramentas para sua execução;
- Descrição do processo de geração de carga de trabalho sintética que tem como base o modelo proposto e disponibilização de ferramentas para a geração da carga;
- Criação de uma nova métrica para análise da qualidade de ajuste de distribuições de probabilidade realizadas em conjunto com técnicas de agrupamento.

Uma avaliação do modelo proposto é apresentada, além de uma análise de cargas

de trabalho geradas a partir dos modelos para diversos cenários de grades computacionais. A nova métrica proposta indica qual a proporção de usuários que obtiveram resultados para o teste de qualidade de ajuste acima do nível de significância determinado. Os resultados mostram que, de acordo com a métrica proposta, ao agrupar usuários pelos seus perfis e extrair um modelo para cada grupo de usuários, aumenta-se a qualidade da modelagem em relação à abordagem de um modelo único para todos os usuários (sem agrupamento). Verifica-se que mesmo com uma quantidade pequena de grupos é possível obter bons resultados para a métrica, o que reduz a complexidade do modelo e mantém sua representatividade. Mostra-se que é possível avaliar ambientes com diferentes cargas ao variar a quantidade de usuários no sistema, além de permitir que seja feita uma análise de soluções de acordo com informações dos usuários que compõem a grade e seus perfis.

A organização do artigo é a seguinte: na Seção 2 são apresentados trabalhos relacionados. A metodologia de modelagem da carga de trabalho é proposta na Seção 3 e a avaliação do modelo proposto é realizada na Seção 4. Por fim, as conclusões são apresentadas na Seção 5.

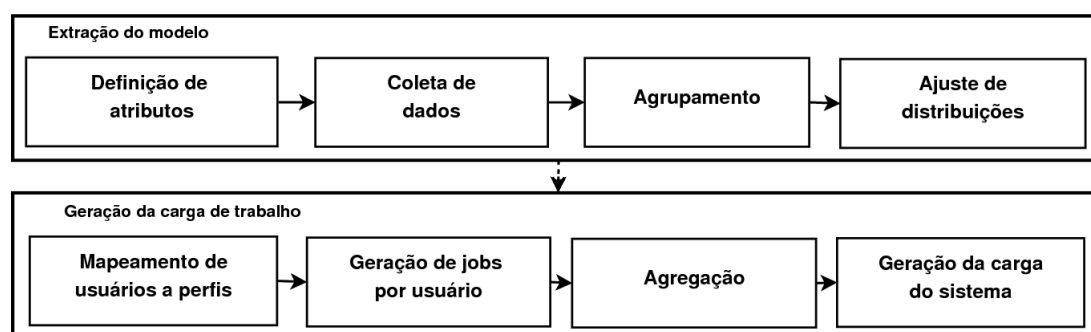
## 2. Trabalhos relacionados

Uma característica presente nos trabalhos de modelagem de carga de trabalho existentes para ambientes de computação paralela é que a demanda que chega no sistema é proveniente de um único modelo. Geralmente, cada atributo é modelado usando apenas uma distribuição de probabilidade e seus parâmetros associados.

Lublin e Feitelson propuseram um modelo de carga de trabalho para computação de alto desempenho em que a demanda é considerada como uma caixa-preta, não havendo uma associação da carga do sistema com os usuários que a geram [Lublin and Feitelson 2003]. Porém, em alguns sistemas, é necessário que a carga de trabalho esteja associada a usuários que a geram, pois esta informação é usada para tomar decisões como por exemplo em escalonadores ou em políticas de compartilhamento. Além disso, defende-se que, ao invés de utilizar um único modelo para gerar a carga do sistema, é mais adequado utilizar modelos hierárquicos baseados no comportamento dos usuários do sistema [Shmueli and Feitelson 2009], como é feito na metodologia proposta no presente trabalho. A ideia é que a carga de trabalho do sistema seja gerada a partir da agregação da demanda de diversos usuários do sistema, modelados separadamente de acordo com seus perfis.

O modelo de carga de trabalho em grades computacionais proposto por Iosup et al. é feito para aplicações do tipo *Bag-of-Tasks* (BoT), onde um *job* é composto por um conjunto de tarefas independentes [Iosup et al. 2008b]. O modelo é extraído a partir de rastros de grades computacionais de serviço, que estão disponíveis no Grid Workloads Archive [Iosup et al. 2008a]. Um problema presente neste modelo é que a carga de trabalho não é alterada à medida que novos usuários são adicionados ao sistema, pois assume-se que a carga gerada pelo sistema é a mesma independente do número de usuários existentes, sendo então essa carga dividida pela quantidade de usuários especificada. Além disso, o comportamento dos usuários e características das aplicações são modelados a partir de uma única função de probabilidade para cada rastro, o que impossibilita uma análise de acordo com os diferentes perfis de usuários e aplicações que cada sistema possui.

A modelagem da carga de trabalho também é estudada em outros campos de sis-



**Figura 1. Etapas da metodologia de modelagem da carga de trabalho.**

temas distribuídos, como em sistemas P2P. O trabalho proposto por Marques-Neto et al. [Marques-Neto et al. 2009] apresenta uma caracterização hierárquica do comportamento dos usuários em um sistema P2P, de acordo com o tipo de sessão de cada usuário. Porém, a caracterização proposta não se aplica diretamente a sistemas de grades computacionais, pois os atributos de carga de trabalho dos dois sistemas são diferentes, além da divisão de usuários em grupos ser específica para o ambiente estudado.

O presente trabalho, por outro lado, apresenta um modelo hierárquico da carga de trabalho de grades computacionais, em que perfis de usuários são modelados de acordo com seus comportamentos e aplicações. Os grupos de usuários são formados a partir da similaridade das distribuições de probabilidade obtida dos dados de cada indivíduo. A carga sintética gerada é aumentada à medida que mais usuários são considerados no modelo e cada perfil de usuários com características semelhantes possui seu próprio modelo.

### 3. Metodologia de Modelagem da Carga de Trabalho

Nesta seção, é apresentada a metodologia para a modelagem da carga de trabalho de grades computacionais proposta neste trabalho. Ela é dividida em duas fases: a extração do modelo e a geração da carga de trabalho sintética. Os modelos obtidos na primeira fase servem de entrada para a segunda, que terá como saída a carga de trabalho sintética. A Figura 1 apresenta cada fase da metodologia e suas fases, que são detalhadas a seguir.

#### 3.1. Extração do Modelo

A primeira fase consiste em extrair o modelo a partir de rastros históricos de utilização de sistemas em produção. Para isto, propõe-se como partes do processo: identificar quais atributos serão utilizados na geração da carga de trabalho sintética e definir para cada atributo quais rastros serão usados para sua extração; coletar os dados dos rastros para cada atributo; agrupar usuários em perfis de acordo com a similaridade para cada atributo; e fazer ajustes de funções de distribuição de probabilidade aos dados para cada perfil de usuários. As etapas da extração do modelo são descritas nas seções seguintes.

##### 3.1.1. Definição de Atributos de Carga de Trabalho

O modelo de carga de trabalho é, na verdade, a composição de modelos de diversos atributos que compõem a carga do sistema. É preciso, então, definir quais atributos serão usados na elaboração do modelo.

Os usuários da grade interagem com o sistema submetendo *jobs* para serem executados na infraestrutura e, após finalizarem as execuções, recebem como retorno os resultados gerados. O atributo usado para descrever esse comportamento das submissões de *jobs* pelos usuários é o:

- **Intervalo Entre Submissões (IES):** Intervalo de tempo entre a submissão de dois *jobs* consecutivos por um mesmo usuário.

Em geral, as aplicações dos usuários são submetidas para a grade em *jobs* do tipo *Bag-of-Tasks* (BoT) [Iosup et al. 2007], que consiste em um conjunto de tarefas independentes que podem ser executadas em paralelo sem a necessidade de comunicação entre elas. Os atributos dessas aplicações que compõem o modelo de carga de trabalho são:

- **Tempo de Execução dos Jobs (TEJ):** Somatório do tempo de execução de todas as tarefas que compõem um mesmo *job* do tipo BoT, considerando uma máquina de referência.
- **Tempo de Execução das Tarefas (TET):** Tempo necessário para que uma tarefa seja executada em uma máquina de referência.

Portanto, modelando os atributos listados acima, pode-se gerar submissões de aplicações à grade com características determinadas por seus modelos.

### 3.1.2. Coleta de Dados

Para que o modelo para geração de carga de trabalho seja criado, é preciso que ele seja baseado em observações de dados históricos de sistemas existentes, ou que seja elaborado a partir da intuição de como a demanda de um certo sistema deve se comportar. A primeira abordagem é a mais indicada, já que se sustenta em dados reais, e foi a utilizada nesta metodologia. Alguns rastros que podem ser utilizados na coleta de dados estão disponíveis no Grid Workload Archive (GWA<sup>1</sup>) e no Parallel Workload Archive (PWA<sup>2</sup>).

### 3.1.3. Agrupamento de Usuários

Sistemas como grades computacionais são compostos por diversos usuários e aplicações, que nem sempre possuem as mesmas características. Por exemplo, alguns usuários podem submeter *jobs* para a grade com uma alta frequência, enquanto outros submetem com baixa frequência. Alguns *jobs* podem ser de longa duração, sendo compostos por uma grande quantidade de tarefas, e outros podem ser menores e com poucas tarefas. Deste modo, modelou-se os atributos de carga de trabalho agrupando usuários em diferentes perfis de acordo com suas características, criando um modelo para cada grupo encontrado. Como no trabalho de Javadi et al. [Javadi et al. 2011], dois métodos de agrupamento foram considerados para a modelagem: o *k-means* e o hierárquico.

O agrupamento hierárquico é realizado de forma iterativa, juntando em cada passo os grupos que possuem maior similaridade, a partir de uma matriz de distâncias entre cada ponto. Inicialmente cada elemento representa um grupo, então são realizados vários

---

<sup>1</sup><http://gwa.ewi.tudelft.nl/>

<sup>2</sup><http://www.cs.huji.ac.il/labs/parallel/workload/>

passos de agrupamento até que no fim é obtido apenas um grupo que descreve todos os pontos. Apesar de ser lento por necessitar de diversas iterações, esta abordagem tem a vantagem de apresentar uma visão de agrupamentos de diversos tamanhos, ao contrário do *k-means* que necessita de uma quantidade  $k$  específica de grupos. Além disso, o algoritmo de agrupamento hierárquico é determinístico, apresentando sempre o mesmo resultado para uma dada matriz de distância entre os pontos, ao contrário do *k-means* que apresenta um comportamento não determinístico, podendo apresentar diferentes resultados em execuções diferentes.

Para a modelagem dos atributos de carga de trabalho, o agrupamento hierárquico se mostrou mais adequado, pois a quantidade de grupos não necessita ser especificada de antemão, apresentando uma visão geral de cada passo do agrupamento para só depois ser definida a quantidade de grupos adequada. Outra vantagem é o fato do algoritmo ser determinístico, pois só uma execução é necessária para a obtenção dos grupos com validade estatística, o que reduz o tempo necessário para o agrupamento.

Para realizar o agrupamento hierárquico, é necessário fornecer como entrada uma matriz de distância entre os pontos a serem agrupados. No caso dos atributos de carga de trabalho, esses pontos serão os usuários de acordo com as suas características de uso da grade (*intervalo entre submissões de jobs*) ou de suas aplicações (*tempo de execução de tarefas e tempo de execução total de jobs*). Uma maneira de descrever as características de cada usuário é através da distribuição de probabilidade dos dados obtidos para cada atributo. Isto pode ser feito gerando a função distribuição acumulada (FDA) empírica dos dados. Após obter a FDA para cada atributo de cada usuário, deve-se usar algum método para indicar qual a distância entre cada distribuição, que será usada no algoritmo de agrupamento hierárquico. Um dos métodos mais utilizados para comparar FDAs empíricas é o teste Kolmogorov-Smirnov (KS), que calcula a distância máxima entre distribuições de probabilidade acumulada [Feitelson 2011]. Este teste foi escolhido para ser usado na geração da matriz de distância do agrupamento hierárquico. Quanto maior a estatística “D” fornecida pelo teste KS, maior a distância máxima entre as distribuições de probabilidade acumulada empíricas.

Cada grupo obtido para cada atributo é descrito por uma distribuição de probabilidade, que descreve o comportamento dos usuários e características das aplicações que compõem os grupos. A próxima etapa descreve o processo de ajuste de funções de distribuição de probabilidade que descrevem os atributos para cada grupo.

### 3.1.4. Ajuste de Funções de Distribuição de Probabilidade

Após o agrupamento de usuários de acordo com as características de cada atributo, são realizados ajustes de funções de distribuição de probabilidade que irão descrever cada grupo para cada atributo da carga de trabalho. Inicialmente, é preciso definir um conjunto de funções de probabilidade candidatas ao ajuste de distribuições. Assim como trabalhos similares [Iosup et al. 2008b, Kondo et al. 2010], foram escolhidas distribuições que são amplamente usadas na modelagem de sistemas paralelos e apresentam baixa complexidade: exponencial, normal, log-normal, gamma e weibull.

Em seguida, é preciso encontrar para cada distribuição quais os valores dos

parâmetros adequados para se obter o melhor ajuste possível. O método utilizado foi o de Estimação por Máxima Verossimilhança (MLE, do inglês *Maximum Likelihood Estimation*), que é amplamente usado e consiste em obter os valores dos parâmetros das distribuições que resultem na maior probabilidade de gerar uma amostra que coincida com os dados empíricos em questão [Feitelson 2011].

Depois de definidos os melhores parâmetros para o ajuste de cada distribuição candidata, é preciso determinar a qualidade de cada ajuste realizado e escolher qual das distribuições obteve o ajuste de melhor qualidade. Para isto, são realizados os testes de Qualidade de Ajuste (GoF, do inglês *Goodness-of-Fit*), que fornecem métricas que indicam quão bom foi o ajuste de uma função de distribuição de probabilidade com os dados [D'agostino and Stephen 1986]. O método Kolmogorov-Smirnov (KS), usado em trabalhos similares [Kondo et al. 2010, Iosup et al. 2008b, Lublin and Feitelson 2003] foi o usado nesta metodologia.

Uma dificuldade encontrada na comparação de FDAs empíricas com teóricas com o teste KS é que sua métrica é muito sensível para amostras muito grandes. Para contornar este problema, foi utilizada uma abordagem usada em outros trabalhos [Javadi et al. 2011, Kondo et al. 2010, Nurmi et al. 2005], em que diversas amostras de tamanho fixo são extraídas das distribuições a serem comparadas e os pares de amostra de mesmo tamanho são comparados entre si. O tamanho de cada amostra usado foi 30 e o teste foi repetido 1.000 vezes para cada par de distribuições, para assim obter a média dos valores geradas pelo teste KS. Estes valores foram os mesmos usados nos trabalhos citados, onde argumentou-se ser valores suficientes para medir a qualidade do ajuste.

Do resultado do teste KS pode-se obter um p-valor, que diz respeito ao resultado do teste de hipótese em que a hipótese nula é que a distribuição de probabilidade dos dados observados é a mesma da função de distribuição de probabilidade teórica utilizada no ajuste. Se o p-valor for maior que o nível de significância  $\alpha$  escolhido, então a hipótese nula não é rejeitada, e o ajuste é considerado satisfatório. O nível de significância usado neste trabalho foi  $\alpha = 5\%$ , assim como em trabalhos similares [Feitelson 2011].

Após os ajustes de distribuições, têm-se diversas funções de distribuição de probabilidade que representam o comportamento de usuários e as características de suas aplicações em uma grade computacional.

### **3.2. Geração da Carga de Trabalho Sintética**

A segunda fase do processo consiste em gerar a carga de trabalho sintética a partir do modelo hierárquico extraído. Primeiro, é definida a quantidade de usuários desejada para compor a grade computacional e a qual perfil (ou grupo) cada usuário pertence. Depois, os valores para cada atributo são gerados através de suas distribuições, com a geração de números aleatórios. Em seguida, a geração de *jobs* de cada usuário é feita a partir dos valores gerados para cada atributo. Por fim, os *jobs* de todos os usuários são agregados para formar a carga total do sistema.

#### **3.2.1. Mapeamento de Usuários da Grade a Perfis**

A grade é composta por diversos *sites*, onde cada um está relacionado a um domínio administrativo e possui um conjunto de usuários. Cada usuário terá um perfil, descrito por

um conjunto de funções de distribuições de probabilidade, existindo uma função para cada atributo do modelo. Os perfis são obtidos da fase de extração do modelo. A quantidade de usuários por site da grade é definida como um parâmetro de entrada do gerador da carga de trabalho sintética.

O mapeamento de usuários a perfis é feito com base em dados obtidos da fase de extração do modelo. Na fase de agrupamento, descrita na Seção 3.1, que extrai os perfis para cada atributo de carga de trabalho de algum rastro, também é obtida a proporção de usuários do sistema que cada perfil (ou grupo) possui. Então, a probabilidade de um usuário ter um determinado perfil na carga sintética será igual à proporção de usuários que este perfil possui no rastro do sistema do qual o modelo foi extraído. Por exemplo, se de um rastro de um sistema com 100 usuários foram extraídos 3 perfis, onde o Perfil 1 cobre 60 usuários, o Perfil 2 cobre 25 usuários e o Perfil 3 cobre 15 usuários, a probabilidade de um usuário ter cada um desses perfis na geração de carga sintética será de 60%, 25% e 15%, respectivamente.

### 3.2.2. Geração de Jobs por Usuário

Após serem definidos os usuários do sistema, cada um associado a um perfil que possui um conjunto de distribuições de probabilidade que irá descrevê-lo, os valores para cada atributo são gerados através de suas distribuições, com a geração de números aleatórios. Em seguida, os valores gerados são combinados para formarem os *jobs* submetidos por cada usuário da grade.

Os seguintes passos descrevem o fluxo para a geração de um *job* da carga de trabalho. Primeiro, obtém-se o valor do intervalo entre submissões, isto é, o tempo entre a submissão do *job* anterior e do próximo que será submetido. Depois, obtém-se através de sua distribuição de probabilidade o valor do tempo de execução total do *job*, ou seja, do somatório do tempo de execução de todas as tarefas que compõem o *job*. Depois disso, geram-se diversos valores para tempos de execução de tarefas, até que o somatório desses valores gerados atinjam o valor do tempo de execução total do *job*. Por exemplo, se o valor para o tempo de execução total do *job* for de 10 horas e a distribuição do tempo de execução de tarefas for constante em 30 minutos, serão geradas 20 tarefas de 30 minutos cada para compor o *job*.

### 3.2.3. Agregação e Geração da Carga Total do Sistema

Por fim, as cargas de trabalho de todos os usuários são agregadas, gerando assim a carga total do sistema. Como resultado final do processo, será criado um registro com uma lista de *tarefas* submetidas à grade, que contém informações do *job* do qual elas fazem parte, do usuário que as submeteu, do *site* do sistema que o usuário faz parte, do horário em que a tarefa foi submetida e quanto tempo de processamento é necessário para que a tarefa seja executada em uma máquina de referência.

## 4. Avaliação da Modelagem

Para a avaliação da modelagem proposta, será realizada a aplicação da mesma para um conjunto de sistemas de grades computacionais existentes, obtendo modelos para cada um



deles. Uma nova métrica é proposta para avaliar a qualidade do agrupamento de usuários e do ajuste de distribuições aos dados. Em seguida, é feita uma análise de cargas de trabalho sintéticas geradas a partir do modelo.

#### 4.1. Rastros para Extração dos Modelos

Para a coleta dos dados dos atributos de carga de trabalho, foram utilizados rastros de grades computacionais disponíveis no Grid Workload Archive. Eles estão listados abaixo, juntamente com algumas características do sistema dos quais foram obtidos:

ID	Sistema (Local, Tipo)	#Jobs	#Sites	#Usuários
GS1	DAS-2 (Holanda, acadêmico)	602 mil	5	332
GS2	Grid'5000 (França, acadêmico)	915 mil	15	473
GS3	NorduGrid (Europa, acadêmico/produção)	781 mil	75+	387
GS4	AuverGrid (França, produção)	404 mil	5	405
GS5	SHARCNET (Canadá, produção)	1,2 milhões	10	412
GS6	EGEE/LCG (Europa, produção)	188 mil	220+	216

**Tabela 1. Rastros usados na extração do modelo [Iosup and Epema 2011].**

#### 4.2. Metodologia de Avaliação

Como visto na Seção 3.1, os atributos de carga de trabalho escolhidos foram: o Intervalo Entre Submissões (IES), o Tempo de Execução de Jobs (TEJ) e o Tempo de Execução das Tarefas (TET). O próximo passo do processo é definir a quantidade de grupos que serão extraídos para cada atributo do modelo.

Como dito anteriormente, cada grupo para cada atributo terá uma função de distribuição de probabilidade associada, que irá descrever o perfil dos usuários que compõem esses grupos. Para avaliar a metodologia com relação ao agrupamento de usuários e ajuste de distribuições de probabilidade, propõe-se uma métrica que determina a qualidade do ajuste para diferentes agrupamentos, que é definida abaixo:

- **Coefficiente de Qualidade de Ajuste ( $Q$ ):** proporção de usuários que possuem ajuste de distribuição de probabilidade satisfatório, considerando que cada usuário faz parte de um grupo e que cada grupo possui apenas uma função de distribuição de probabilidade para descrever cada atributo. Um ajuste de distribuição de probabilidade é considerado satisfatório quando o p-valor obtido no teste KS é maior que o nível de significância  $\alpha = 5\%$ .

Quanto maior a quantidade de grupos, menor a quantidade de usuários por grupo e maior a quantidade de distribuições que serão usadas para descrever o comportamento desses usuários. Se a quantidade de grupos ( $k$ ) para a modelagem de um atributo for 1, existirá apenas uma distribuição de probabilidade que irá representar o comportamento de todos os usuários do sistema. Isto faz com que o modelo fique bastante genérico, o que é um ponto positivo. Porém, a probabilidade de apenas uma distribuição obter um ajuste satisfatório para todos os usuários do sistema é baixa ( $Q \approx 0$ ), pois a tendência é que usuários possuam diferentes características.

Por outro lado, se  $k$  for igual à quantidade total de usuários do sistema, então cada grupo só será composto por um usuário e para cada usuário do sistema existirá uma distribuição de probabilidade que irá representar seu comportamento. Desta forma, será mais fácil obter ajustes satisfatórios ( $Q \approx 1$ ), pois cada distribuição só terá que representar

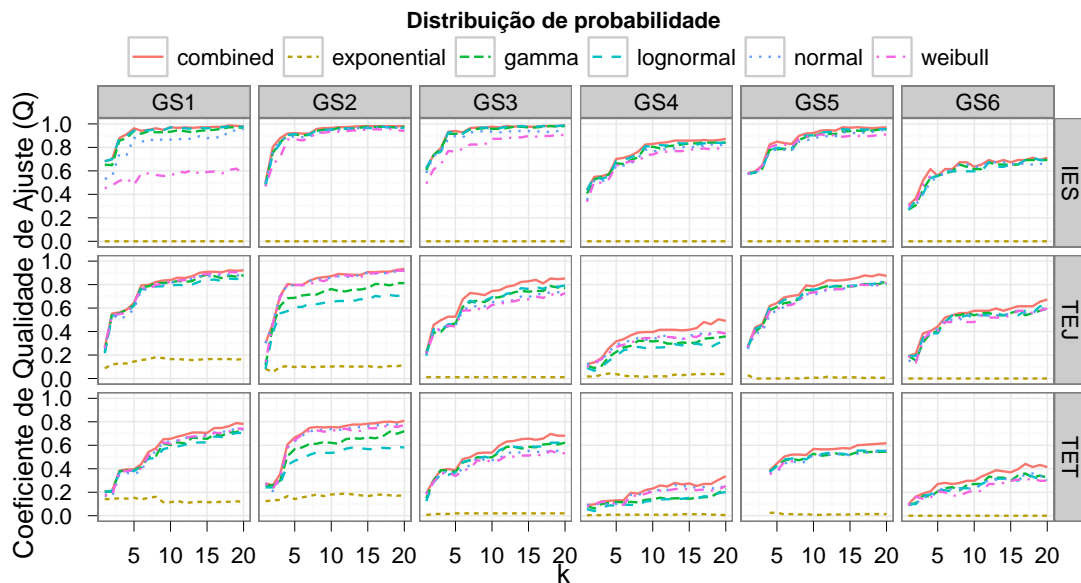


Figura 2. Coeficiente de Qualidade de Ajuste ( $Q$ ).

o comportamento de um único usuário. Porém, como a quantidade de grupos será grande e, conseqüentemente, existirão muitas distribuições de probabilidade que só representarão um único usuário cada uma, o modelo se tornará muito específico e apresentará maior complexidade. Desta forma, procura-se escolher uma quantidade  $k$  de grupos que não seja tão grande, mas que também consiga obter uma alta proporção de usuários com ajustes de distribuição satisfatórios (ou seja, altos valores de  $Q$ ).

Para o ajuste de distribuição de probabilidades, uma transformação logarítmica com base 2 foi aplicada aos dados, para reduzir o impacto de valores extremos no ajuste. Esta é uma prática usada em modelagem de carga de trabalho que não invalida o ajuste de distribuições [Lublin and Feitelson 2003, Iosup et al. 2008b]. Note que para gerar a carga a partir do modelo é preciso que a função inversa seja aplicada, ou seja, a função exponencial com base 2 deve ser usada.

### 4.3. Avaliação

Nesta seção são apresentados resultados da avaliação da modelagem proposta. A Figura 2 mostra a métrica de Coeficiente de Qualidade de Ajuste ( $Q$ ) com quantidades de grupos ( $k$ ) variando de 1 a 20. Os agrupamentos e ajustes de distribuições foram feitos para cada um dos 6 rastros de grades de serviço (GS) separadamente. Cada gráfico representa um cenário de atributo de carga de trabalho e rastro de sistema, apresentando um total de 18 cenários. As 5 distribuições usadas no ajuste são as mencionadas na Seção 3.1.4. Além do resultado dos ajustes usando cada uma distribuições separadamente, também é apresentado o resultado chamado *combined*, que permite que haja uma combinação de diferentes distribuições para os diferentes grupos, sendo escolhida para cada grupo a melhor distribuição obtida entre as 5 candidatas, de acordo com o teste KS de qualidade de ajuste.

Observa-se que, à medida que o  $k$  aumenta, o coeficiente de qualidade de ajuste

também tende a aumentar. Isto é esperado, pois quanto maior o  $k$ , maior a quantidade de distribuições usadas no ajuste e menor a quantidade de usuários que cada distribuição terá que representar, aumentando assim a probabilidade de se obter ajustes satisfatórios para os dados dos usuários. Na maioria dos casos, o coeficiente  $Q$  tem um alto crescimento ao incrementar os valores de  $k$  iniciais. Após um certo valor de  $k$ , a curva de  $Q$  tende a estabilizar, não havendo mais um crescimento significativo à medida que o  $k$  aumenta.

Percebe-se que os piores resultados ocorrem quando existe apenas um grupo descrevendo todos os usuários ( $k = 1$ ). O fato da qualidade de ajuste melhorar consideravelmente quando é usado mais de um grupo, como visto na Figura 2, destaca a importância do uso do agrupamento de usuários em perfis na metodologia de modelagem. Deste modo, deseja-se obter um  $k$  que não seja muito grande, mas que apresente um  $Q$  próximo ao máximo possível.

A distribuição exponencial apresenta valores de  $Q$  próximos a zero para todos os cenários. Este fato é relevante, pois esta distribuição é bastante usada em análise de desempenho de sistemas, principalmente na modelagem do intervalo entre submissões de *jobs*, quando advoga-se que a taxa de submissões segue uma distribuição Poisson e, portanto, o intervalo entre submissões segue uma exponencial. Neste caso, ela não se mostrou adequada para a modelagem dos atributos em questão.

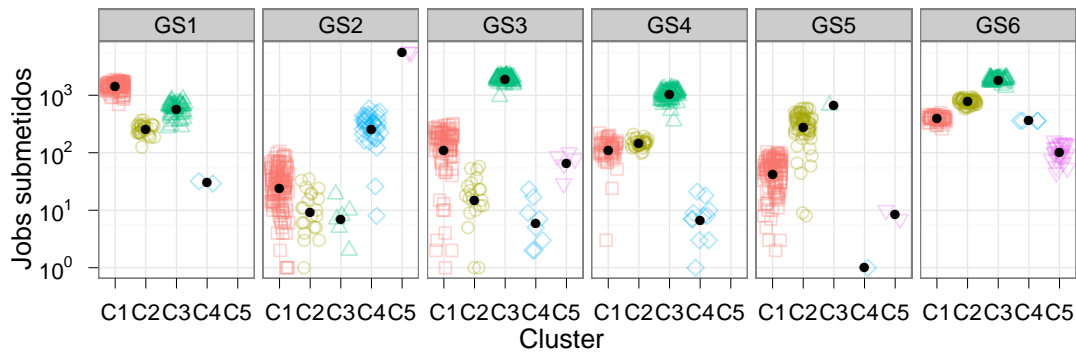
Não há diferença muito significativa entre os resultados das outras distribuições, havendo apenas alguns cenários isolados em que alguma distribuição se sobressai com relação a outra. Porém, como esperado, usar uma combinação (*combined* no gráfico) das melhores distribuições para cada grupo sempre apresenta os melhores resultados, sendo portanto a abordagem usada nos modelos.

Está fora do escopo deste trabalho elaborar uma técnica que procure obter o melhor valor de  $k$  para cada atributo. Observa-se através dos gráficos que, para a maioria dos cenários, para quantidades de grupos maiores que  $k = 5$ , a curva do Coeficiente de Qualidade de Ajuste ( $Q$ ) para a abordagem *combined* já apresenta um aumento pouco expressivo. Então, por simplicidade, foi escolhida a quantidade de grupos  $k = 5$  para a modelagem de todos os atributos de todos os rastros.

#### 4.4. Análise de Cargas de Trabalho Sintéticas

Na aplicação do modelo, foram obtidos para cada atributo 5 grupos de usuários para cada um dos 6 rastros usados na extração. Para gerar cargas de trabalho sintéticas, é preciso especificar a quantidade  $N$  de usuários que irão compor a grade e qual o rastro que será usado na coleta dos dados. Ao gerar a carga, cada usuário é associado a um dos 5 grupos. A probabilidade de um usuário ser associado a um grupo é igual à proporção de usuários que faziam parte deste grupo no rastro original, como especificado na Seção 3.2.1.

Na Figura 3 é apresentada a quantidade de *jobs* submetidos por cada usuário durante um mês de carga sintética de uma grade computacional configurada para conter 200 usuários no total. Os usuários são divididos de acordo com seus grupos (*clusters*) no eixo-x e pelo rastro usado na extração nos diferentes gráficos. Os símbolos coloridos não preenchidos representam os valores para cada usuário individualmente, enquanto os pontos pretos preenchidos representam as médias dos *jobs* submetidos pelos usuários que compõem cada um dos grupos. O eixo-y é apresentado em escala de log.

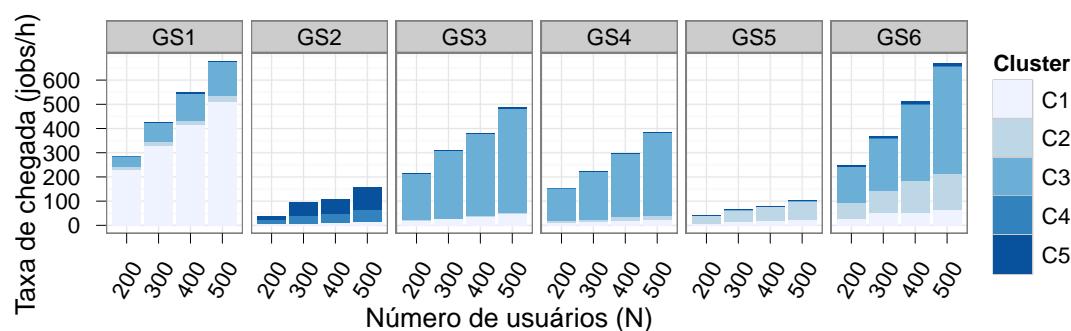


**Figura 3. Jobs submetidos por cada usuário em 1 mês de carga sintética.**

Note que existem diferentes perfis de usuários com relação à taxa de submissão de *jobs* à grade. Por um lado, existem usuários que submeteram mais de 1.000 *jobs* em um mês de carga, como os que compõem os grupos:  $\langle GS1, C1 \rangle$ ,  $\langle GS2, C5 \rangle$ ,  $\langle GS3, C3 \rangle$  e  $\langle GS6, C3 \rangle$ , onde a tupla  $\langle GS_i, C_j \rangle$  representa o grupo  $C_j$  do modelo extraído usando o rastro de grade  $GS_i$ . Por outro lado, existem grupos que seus usuários submeteram menos de 10 *jobs* durante o mesmo mês, como:  $\langle GS2, C2 \rangle$ ,  $\langle GS2, C3 \rangle$ ,  $\langle GS3, C4 \rangle$ ,  $\langle GS4, C4 \rangle$ ,  $\langle GS5, C4 \rangle$  e  $\langle GS5, C5 \rangle$ . É importante ressaltar que grupos obtidos a partir de diferentes rastros não possuem relação alguma entre si, mesmo sendo usados identificadores similares na apresentação dos resultados.

Também é possível notar que há uma diferença na quantidade de usuários que compõem cada grupo, vendo a quantidade de pontos que cada um apresenta. Observando os resultados do modelo do rastro  $GS1$ , por exemplo, verifica-se que os grupos  $C1$  e  $C3$  possuem uma grande densidade de pontos, o que significa que existem muitos usuários que fazem parte desses grupos. Por outro lado, o grupo  $C4$  possui poucos pontos e o grupo  $C5$  não possui ponto algum, o que significa que no rastro original a proporção de usuários que fazem parte desses grupos é muito baixa, o que resulta numa baixa probabilidade de um usuário da carga de trabalho sintética ser associado a esses grupos. Uma outra característica que pode ser notada é a diferença na variância dos usuários que compõem os mesmos grupos. Para alguns grupos, seus usuários possuem comportamentos bastante parecidos, o que é representado pelos pontos bastante concentrados ao redor da média, como acontece em todos os grupos do rastro  $GS6$ . Porém, existem grupos que apresentam uma maior variância no comportamento de seus usuários, como em quase todos os grupos obtidos do rastro  $GS2$ , em que os pontos estão bastante espalhados ao redor da média. Desta forma, com uma carga de trabalho que possui usuários identificados pelos seus diferentes perfis, com seus comportamentos extraídos de rastros reais, é possível fazer uma análise de soluções para a grade avaliando usuários de acordo com suas características e como cada um influencia no sistema de acordo com métricas de interesse.

Neste tipo de modelo, baseado no comportamento dos usuários, podemos variar a carga no sistema alterando a quantidade de usuários que compõem a grade. Na Figura 4, observa-se a taxa de chegada de *jobs* no sistema por hora, variando a quantidade de usuários  $N$  usada na geração da carga sintética em 200, 300, 400 e 500 usuários para cada rastro utilizado. Nota-se que a carga no sistema apresenta um aumento visualmente li-



**Figura 4. Taxa de chegada de jobs para cargas sintéticas.**

near à medida que se aumenta a quantidade de usuários. Também é possível observar a participação de cada grupo de usuários na carga total do sistema, representada pelas diferentes cores das barras. Percebe-se que alguns grupos possuem uma fatia bem maior do que outros. Isto ocorre pois alguns grupos possuem mais usuários do que outros na carga gerada, além de haverem diferenças na frequência de submissões de *jobs* dependendo do grupo que cada usuário está associado. Deste modo, é possível gerar cargas de trabalho com diferentes características de demanda dos usuários, porém mantendo as características dos usuários e a representatividade do sistema ao manter a proporção de cada grupo de usuários no sistema.

Para reprodutibilidade, as ferramentas desenvolvidas para extração dos modelos e geração das cargas sintéticas de acordo com a metodologia proposta estão disponíveis em: <http://www.lsd.ufcg.edu.br/%7Emarcus/grid-workload-model>.

## 5. Conclusões

Neste trabalho, foi proposto um novo modelo de carga de trabalho de grades computacionais. A metodologia para sua extração se baseia no agrupamento de usuários de acordo com seus perfis de uso da grade e de suas aplicações. Também foi proposta uma nova métrica que indica qual a proporção de usuários que obtiveram resultados para o teste de qualidade de ajuste acima do nível de significância determinado. Verificou-se que o agrupamento de usuários melhora a qualidade da modelagem em relação a abordagens que consideram um modelo único para todos os usuários do sistema. Além disso, a carga sintética gerada a partir do modelo pode ser facilmente variada mudando a quantidade de usuários desejada, mas mantendo as características dos usuários e a representatividade do sistema em questão.

## Referências

- Anglano, C., Brevik, J., Canonico, M., Nurmi, D., and Wolski, R. (2006). Fault-aware scheduling for bag-of-tasks applications on desktop grids. In *IEEE/ACM International Conference on Grid Computing, GRID '06*, pages 56–63, Washington, DC, USA. IEEE Computer Society.
- Bittencourt, L. F., Madeira, E. R. M., Cicerre, F. R. L., and Buzato, L. E. (2006). Uma heurística de agrupamento de caminhos para escalonamento de tarefas em grades computacionais. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*.

- Carvalho, M., Miceli, R., Jr., P. D. M., Brasileiro, F., and Lopes, R. (2010). Predicting the quality of service of a peer-to-peer desktop grid. In *Workshop on Desktop Grids and Volunteer Computing Systems (PCGrid)*. IEEE Computer Society.
- D'agostino, R. B. and Stephen, M. S., editors (1986). *Goodness-of-Fit Techniques*. Marcel Dekker, New York and Basel.
- Feitelson, D. (2011). Workload modeling for computer systems performance evaluation. Book draft disponível em: <http://www.cs.huji.ac.il/~feit/wlmod/>.
- Iosup, A. and Epema, D. (2011). Grid computing workloads. *IEEE Internet Computing*.
- Iosup, A., Jan, M., Sonmez, O., and Epema, D. (2007). The characteristics and performance of groups of jobs in grids. In *Euro-Par*, pages 382–393. Springer-Verlag.
- Iosup, A., Li, H., Jan, M., Anoep, S., Dumitrescu, C., Wolters, L., and Epema, D. H. J. (2008a). The Grid Workloads Archive. *Future Generation Computing Systems*, 24(7):672–686.
- Iosup, A., Sonmez, O., Anoep, S., and Epema, D. (2008b). The performance of bags-of-tasks in large-scale distributed systems. In *International Symposium on High Performance Distributed Computing (HPDC)*, pages 97–108.
- Javadi, B., Kondo, D., Vincent, J.-M., and Anderson, D. P. (2011). Discovering statistical models of availability in large distributed systems: An empirical study of seti@home. *IEEE Transactions on Parallel and Distributed Systems*, 99.
- Kondo, D., Javadi, B., Iosup, A., and Epema, D. (2010). The failure trace archive: Enabling comparative analysis of failures in diverse distributed systems. *IEEE International Symposium on Cluster Computing and the Grid (CCGrid)*, pages 398–407.
- Lublin, U. and Feitelson, D. G. (2003). The workload on parallel supercomputers: modeling the characteristics of rigid jobs. *Journal of Parallel and Distributed Computing*, 63:1105–1122.
- Marques-Neto, H. T., Valle, E. V., Castilho, L. H. d. D., Almeida, J. M., and Almeida, V. A. F. (2009). Caracterização hierárquica do comportamento dos usuários de sistemas par-a-par na internet de banda larga. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 61–74.
- Nurmi, D., Brevik, J., and Wolski, R. (2005). Modeling machine availability in enterprise and wide-area distributed computing environments. In *Euro-Par*, pages 432–441.
- Ponciano, L., Santana, J., Carvalho, M., Gaudencio, M., and Brasileiro, F. (2010). Análise de estratégias de computação verde em grades computacionais oportunistas. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 307–320.
- Shmueli, E. and Feitelson, D. G. (2009). On simulation and design of parallel-systems schedulers: Are we doing the right thing? *IEEE Transactions on Parallel and Distributed Systems*, 20:983–996.