2
Gerenciamento de Log

2.1
Definições básicas

Os logs são fontes riquíssimas de informação e são gerados pelos servidores e pelas aplicações conforme eventos significativos acontecem. Em [1], *log* é definido como um conjunto de registros com marcação temporal, que suporta apenas inserção, e que representa eventos que aconteceram em um computador ou equipamento de rede. Estes registros constituem a fonte básica de informação tanto para a detecção e resolução de problemas quanto para informações de negócio, como métricas de acesso e comportamento de usuários.

O conjunto de atividades desempenhadas na operação de logs, como visualização, armazenamento, compactação, e assim por diante, pode ser denominado de processo de gerenciamento de logs. A importância deste processo é evidenciada por publicações que orientam como as atividades deste processo devem ser conduzidas [1,3]. A seção 2.4 deste capítulo detalha as atividades envolvidas em um sistema que tem como objetivo suportar o processo de gerenciamento de logs.

No contexto de uma empresa de Internet, como é o caso da Globo.com, os logs das aplicações são usados tanto para extrair informações de negócio, quanto para identificar e resolver problemas. Estas duas necessidades básicas de logs, informação de negócio e administração de aplicações, se tornam um desafio quando a realidade dos negócios na Internet impõe que as informações de acesso sejam geradas em tempo real e que o parque de servidores, cada vez maior para atender a crescente demanda, seja mais bem administrado.

Nesse sentido, sob o ponto de vista da administração de sistemas, quando o volume de acesso aumenta nas aplicações, o número de servidores tende a crescer também, e, com isso, o volume de log não só aumenta, mas como também passa a estar distribuído pelos servidores. Esta distribuição de logs é uma realidade em diversas aplicações que utilizam desde dezenas de

PUC-Rio - Certificação Digital Nº 0621305/CA

servidores, como é o caso da Globo.com, até milhares de servidores, como é o caso do Google 4.

Desta forma, é importante notar que, ao mesmo tempo em que os desafios técnicos em gerenciar um crescente e distribuído volume de logs se tornam evidentes, as demandas de negócio pressionam por informações em tempo real e por uma administração mais eficiente das aplicações que garanta sempre a disponibilidade.

2.2 Logs como um fluxo de eventos

Um dos pontos importantes desta tese é olhar os logs sob um ponto de vista diferente, ou seja, tratar os arquivos de logs de uma aplicação como um fluxo de eventos, para então aplicar técnicas de correlação de eventos baseadas em regras e consultas, que serão abordadas no Capítulo 3.

Um *evento* é definido 12 simplesmente como a ocorrência de uma atividade, ou, no contexto de sistemas de computação, definido como uma representação de uma atividade que acontece em um sistema de computação. Desta forma, cada linha gerada em um arquivo de log corresponde a um evento e a seqüência destes eventos corresponde a um fluxo de eventos. Sobre estes fluxos podem então ser definidas regras e consultas para análise em tempo real. A figura a seguir ilustra esta idéia:

PUC-Rio - Certificação Digital Nº 0621305/CA

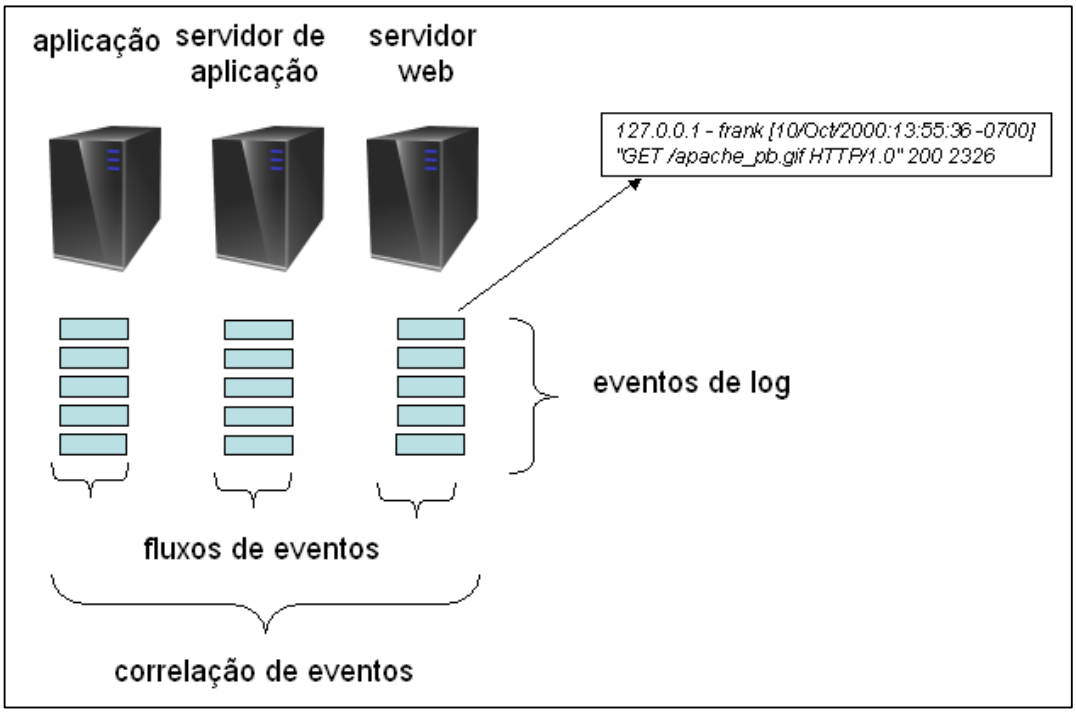

Figura 1 - Logs como fluxos de eventos

2.3 Caracterização e padrões de logs

Dado o grande conjunto de aplicações existentes, é inviável haver um padrão único para todas as mensagens de log. Cada tipo de aplicação distinta possui um formato diferente para seus registros de logs. Contudo existem informações básicas que estão presentes na maioria destes registros 5:

- Marcação temporal (*timestamp)*: contém a informação de data e hora em que o registro de log foi gerado. É muito importante para que se possa comparar a ocorrência de um evento com outros registros de log no tempo e, assim, correlacioná-los.
- Severidade: caracteriza o quão importante é aquele registro tendo em vista a integridade do sistema. A severidade é determinada sobre uma escala gradativa que se inicia com mensagens informativas e termina em mensagens que caracterizam uma falha grave no sistema. A tabela a seguir descreve esta escala:

PUC-Rio - Certificação Digital Nº 0621305/CA

Nível	Descrição
Debug ou *trace*	Informações sobre o fluxo de execução do sistema, muito utilizadas em fase de desenvolvimento para correção de erros na aplicação.
Info ou *notice*	Caracteriza um evento meramente informacional.
Warn	Caracteriza um evento sobre o qual se deva ter uma maior atenção e, possivelmente, uma ação pró-ativa para prevenção de erro. Contudo, um erro no sistema não foi caracterizado.
Error	Caracteriza um evento de erro no sistema.
Fatal	Caracteriza um evento de erro grave, ou seja, que pode ocasionar degradação ou indisponibilidade de um sistema.

Tabela 1 – Níveis de severidade para eventos de log

Quando a estrutura de gerenciamento de logs é centralizada, outra informação importante e presente nos eventos de log é a fonte. Esta informação identifica que servidor, ou aplicação, gerou aquele evento de log.

Felizmente existem hoje diferentes padrões de log estabelecidos. Como esta dissertação está inserida no contexto de gerenciamento de aplicações, serão abordados dois padrões muito utilizados em servidores *Web* e em aplicações *Java* 6.

Para os servidores *Web*, como o caso do Apache 7, foi estabelecido um padrão para os logs de acesso, o *Common Logfile Format* 8, que possui um conjunto de informações fixas a serem registradas a cada evento de log. A Tabela 2 indica a seqüência das informações em uma única linha do *Common Logfile Format*. A Figura 2 mostra um exemplo de mensagem de log para este padrão.

PUC-Rio - Certificação Digital Nº 0621305/CA

Informação	Descrição
Remotehost	Nome ou endereço IP do computador que realizou o acesso
RFC931	O nome identificador do usuário remoto
Authuser	O nome de usuário com o qual o usuário se autenticou
Date	Data e hora da requisição
Request	A linha de requisição exatamente como gerada pelo cliente
Status	O código[1] HTTP9 retornado
Bytes	O tamanho do conteúdo do documento transferido

Tabela 2 – Descrição de campos do *Common Logfile Format*

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif http/1.0" 200 2326

Figura 2 – Exemplo de registro de evento de log no padrão *Common Logfile Format*

Para aplicações escritas em *Java* e servidores de aplicação como o *Jboss Web* 11, a biblioteca *log4j* 10 é hoje bastante utilizada, podendo considerá-la um padrão para geração de logs. Contudo, esta biblioteca permite especificar os arquivos e o formato das linhas de log a serem geradas e, sendo assim, as mensagens geradas podem variar de acordo com o que foi configurado. A Tabela 3 traz um exemplo de mensagens de log em diferentes formatos geradas pelo *log4j*:

[1] Uma listagem completa com os códigos HTTP podem ser encontrada na referência 9.

PUC-Rio - Certificação Digital Nº 0621305/CA

Mensagem de log em formato de texto simples:
INFO [main] (MyApp2.java:12) - Entering application.
Mensagem de log em formato XML:
<log4j:event logger="org.apache.log4j.xml.XMLLayoutTestCase" *timestamp="Mon Aug 10 21:07:25 CEST 2006"* *sequenceNumber="145"* *level="DEBUG"* *thread="main">* *<log4j:message><![CDATA[hi]]></log4j:message>* *<log4j:throwable><![CDATA[java.lang.Exception* *at org.apache.log4j.xml.XMLLayoutTestCase.testNull(X)* *at java.lang.reflect.Method.invoke(X)* *[..]* *]]></log4j:throwable>* *</log4j:event>*

Tabela 3 – Diferentes formatos de mensagens de log com *log4j*

Takada e Koike 13 propõem um formato genérico para linhas de logs bastante simples. O *General Log Format*, formato utilizado pela aplicação de visualização de logs apresentada no artigo, contém quatro campos, sendo dois pré-fixados e dois livres. Os campos fixos são a marcação temporal e o campo "mensagem". Existem outros dois campos nomeados como "*tag 1"* e "*tag 2*" que podem ser utilizados para outras informações, por exemplo, a severidade e a identificação de quem originou a requisição.

Outra iniciativa de proposta de formato de log é encontrada em 24, que propõe um padrão baseado em XML para Bibliotecas Digitais.

Baseando-se neste estudo sobre os padrões e nas principais informações de um evento de log e considerando os requisitos que serão expostos no Capítulo 4, um padrão será proposto nesta dissertação, também no Capítulo 4, para representar os eventos de log.

PUC-Rio - Certificação Digital Nº 0621305/CA

2.4
Sistema de gerenciamento de logs

O gerenciamento de logs sob o ponto de vista de um sistema envolve um conjunto amplo de atividades. Baseando-se em [1, 3, 5], esta dissertação lista as seguintes atividades relacionadas ao gerenciamento de logs:

- **Análise léxica e transformação**: é o processo que ,analisa,as linhas de mensagens de log e produzir uma saída formatada em um padrão mais adequado para futuro processamento.
- **Transmissão e recepção**: é o processo que transmite os eventos de logs para um servidor central, em conjunto com o processo que recebe as mensagens de log no servidor.
- **Análise sobre eventos de log**: é o processo que, através de algoritmos, regras ou consultas extrai informação relevante sobre mensagens de log.
- **Compactação**: é o processo que , reduz o tamanho de uma informação através de algoritmos de compactação de texto. Como exemplo, pode-se citar os algoritmos baseados no algoritmo de Lempel-Ziv-Welch.
- **Armazenamento**: é o processo que compreende em guardar os registros de log por um tempo determinado. Este processo é necessário muitas vezes para fins legais e, também, desejado para que se possa ter um histórico.
- **Indexação e Busca**: é o processo responsável por estruturar os registros de logs de forma que sejam posteriormente pesquisados.
- **Visualização**: é o processo que permite a visualização dos registros de log, sejam eles em tempo real ou dos registros já armazenados.

Para realizar estas atividades, uma nova classe de sistema foi criada e chamada de *Sistemas de Gerenciamento de Log* (*Log Management Systems* - LMS). Sah 1 propõe uma arquitetura completa de um sistema de gerenciamento de logs, voltada para sistemas com grande volume de acesso e distribuídos. Vale ressaltar que o sistema de gerenciamento de logs apresentado é um sistema proprietário.

A arquitetura proposta compõe-se de três serviços principais o *Master Service,* o *Parsing Service* e o *Storage Service*. O *Master Service* recebe requisições HTTP contendo linhas de logs em um formato XML. Cada requisição é passada então em paralelo para o *Parsing Service,* que irá através de expressões regulares retirar informações das linhas de log, e para o *Storage*

PUC-Rio - Certificação Digital Nº 0621305/CA

Service que irá indexar, compactar e armazenar os logs. A análise dos logs é feita sobre a forma de consultas SQL sobre uma base de dados modificada. Quanto à visualização, não foi claramente especificada como esta acontece no sistema.

Fischer 5 divide a arquitetura de sistemas de gerenciamento de log em três camadas: entrada, processamento e saída. Em sua tese, propõe o uso do protocolo XMPP 14 como método de transmissão e recepção de mensagens de logs, tendo como vantagens a interoperabilidade, segurança, independência de plataforma, confiabilidade e já atual grande adoção do protocolo XMPP.

Sobre as atividades de um LMS listadas anteriormente, a atividade de análise de logs é a que talvez tenha mais atraído o interesse de pesquisas. Existe uma literatura ampla sobre algoritmos de análise de logs. Como exemplos, pode-se citar:

- Adeva e Atxa 19 que propuseram um algoritmo baseado em técnicas de text-mining capaz de diferenciar comportamentos normais de maliciosos com base nos logs do servidor de aplicação.
- Vaarandi [17, 18] que publicou dois artigos sobre identificação de padrões freqüentes em logs.
- Algoritmos capazes de minerar arquivos logs e identificar perfis de usuários 19, períodos interessantes para o negócio 20, e algoritmos que conseguem descobrir estes padrões em tempo real 21.
- Algoritmos capazes de inferir regras sobre o comportamento de uma aplicação sobre um conjunto de logs [22, 23].

Além destes algoritmos, existem também os componentes que possibilitam uma forma mais simples de correlação de eventos de log, atividade que será definida no Capítulo 3, capaz de produzir resultados e responder aos eventos em tempo real. Estes componentes se baseiam em regras pré-determinadas para correlação de eventos e em consultas do tipo SQL que podem ser criadas e executadas continuamente sobre um fluxo de eventos. Esses componentes serão detalhados no Capítulo 3 desta dissertação.

Sobre a atividade de visualização de logs, 13 apresenta o Mielog, um sistema desenvolvido em C++ que propõe uma interface interativa para visualização e navegação sobre mensagens de log.

Ainda sobre atividades de gerenciamento de logs, é importante ressaltar que centralizar os logs não é uma tarefa simples. Quando se está lidando com volumes altíssimos de logs, estas mensagens tem que ser transmitidas, e isto

PUC-Rio - Certificação Digital Nº 0621305/CA

pode representar um tráfego alto na rede. Na Globo.com, por exemplo, um único servidor Web chega a gerar logs com mais de 2 GB por dia. Schlossnagle 15 propõe algumas arquiteturas para a centralização dos logs, entre elas, o uso de comunicação de grupo (*multicast*) para tentar diminuir o tráfego na rede quando as mensagens são transmitidas para mais de um destino.

PUC-Rio - Certificação Digital Nº 0621305/CA