5
Conclusão

Por muitas vezes, nos deparamos com situações onde os logs gerados por um determinado software não são suficientes para diagnosticar a causa dos seus defeitos, e por este motivo, a opção pela utilização de novas ferramentas que nos auxiliem a encontra-los são sempre bem vindas. Dessa forma, este trabalho apresentou uma forma simples, porém funcional, de interceptar mensagens trocadas pelos componentes do software com a finalidade de auxiliar a depuração de sistemas distribuídos. Sem a necessidade de instrumentação e, através da inserção de código no núcleo do sistema operacional, capaz de interceptar as chamadas de sistema que realizam eventos de E/S.

Enquanto ainda não existe uma ferramenta de diagnóstico tipo bala de prata, capaz de encontrar todos os tipos de defeitos em sistemas distribuídos, qualquer contribuição que auxilie no processo de depuração pode ser considerado de valor considerável. Entretanto, existe um compromisso entre a dificuldade instrumentar o código (seja por uso de mensagens, inclusão de *breakpoints*,...) e o grau de dificuldade de encontrar o defeito observado.

Nesse ponto, ferramentas não intrusivas, como a proposta por este trabalho, mesmo quando utilizada para analisar a comunicação ponto a ponto entre componentes, em uma metodologia *top-down*, ainda traz benefícios por exigir o mínimo de esforço para a preparação do ambiente para monitoração e pela forma de navegação pelos pacotes trocados por estes, por conta da capacidade de decodificação avançada de protocolos e interface intuitiva disponibilizada pelo *wireshark*.

Ainda que os testes realizados por este trabalho indiquem que existe um grave problema de desempenho em condições de elevada concorrência, a proposta de monitorar eventos de E/S mostrou ser um método de depuração válido, capaz de mostrar informações que podem ficar ocultas a quem estiver depurando o software, pois, como indicado na seção 4.4.4, pode ser que o *middleware* gerencie bem uma condição de erro, porém, há situações onde o software pode não ter sido preparado para lidar com tal condição, culminando em um defeito. Nesse caso, a troca de mensagens entre componentes mostra claramente qual evento fez com que o defeito se manifestasse. Sendo assim possível replicar a mensagem responsável por causar o defeito, ou a utilizar para análises posteriores.

Entretanto, não podemos descartar a hipótese de que as demais ferramentas avaliadas por este trabalho identifiquem, de maneira similar, o tipo de defeito acima referido. Porém, provavelmente, a apresentação não será de forma tão amigável quanto à forma proposta por esta ferramenta, através do uso da interface GUI disponível no *wireshark*.

5.1 Trabalhos futuros

Este trabalho abre possibilidades de trabalhos futuros na área de melhora de desempenho do processo de interceptação das chamadas de E/S. Caso uma das soluções propostas na seção 4.5 seja desenvolvida, já seria possível estudar a redução da degradação imposta pela utilização da região crítica observada por esta ferramenta.

Além dos ajustes de desempenho, há oportunidades de estudo sobre a possibilidade de utilização dos arquivos *libpcap* produzidos por esta ferramenta com o propósito de reproduzir a sequência de eventos realizados pelo software monitorado, de forma análoga as ferramentas de *checkpoint and restore* descritas na seção 2.2.7, porém com uma abrangência mais simplificada. Uma ferramenta com este estilo seria capaz de eliminar a necessidade de criação de scripts *ad-hoc* com o objetivo de reconstruir a sequência de chamadas armazenadas nos arquivos *libpcap*.

Ainda sobre os arquivos libpcap, seria possível construir um "servidor de mediação" desses arquivos, que por conta da utilização da metodologia *bottom-up* foram distribuídos pelos servidores monitorados, poderiam ser concatenados em um único ponto com o propósito de facilitar a manipulação destes. Além disso, ainda seria possível trabalhar no mecanismo que reconstrua a causalidade das mensagens, complementando o propósito do servidor proposto. Os fundamentos utilizados por França [13] em sua dissertação de mestrado poderiam ser utilizados para auxiliar na referida reconstrução.

Além das propostas apresentadas, alterando o foco para a interface GUI, há condições de se construir um assistente para a criação de *plug-ins* para o *wireshark* com o objetivo de facilitar a criação e o carregamento das extensões com as regras de decodificação das IDLs.