

시스템 관리 기능을 위한 결함분석기 개발

윤 재 식^o, 고 정 국, 김 길 용
부산대학교 컴퓨터공학과

Development of a Fault Analyzer for System Management Facility

Jae-Sig Yun^o, Jeong-Gook Koh and Gil-Yong Kim
Dept. of Computer Engineering, Pusan National University

요 약

최근 들어 중대형 컴퓨터 시스템에서 결함 진단 구조를 필수적으로 구비하는 추세에 있다. 본 논문은 이러한 추세에 따라 시스템에서 발생하는 결함을 탐지, 분석 후 해결 방안을 자동으로 제시해주는 결함분석기를 개발하였다. 결함 해결 방안은 결함분석보고서 형태로 제시되는데, 시스템 관리자는 결함분석보고서를 통해 결함 원인을 파악하고, 결함을 해결한다. 한편, 결함분석기가 효과적으로 결함 분석을 하기 위해서는 지식기반 데이터베이스의 지속적인 유지, 관리가 중요하다. 본 논문에서는 이러한 지식기반 데이터베이스의 관리방법도 개선하였다.

1. 서 론

컴퓨터 기술의 발전과 활용도가 증가됨에 따라 컴퓨터에 대한 의존도도 함께 증가되게 되었다. 이러한 의존도가 증가됨으로서 컴퓨터 시스템에서 발생하는 결함이 미치는 피해 또한 커지게 되었고, 결함을 자동으로 진단할 수 있는 결함 진단 구조가 필수적으로 요구되게 되었다. 본 연구에서는 시스템에서 발생하는 결함을 탐지, 분석 후 결함의 원인과 증상 및 해결 방안을 자동으로 제시해 주는 결함분석기를 개발하였다. 개발된 결함분석기는 결함 발생시 관리자의 경험이나 감각에 의존하지 않고, 이전에 발생하였던 결함의 증상 및 원인, 해결 방안들을 바탕으로 정확하고 효율적인 해결 방안을 자동적으로 제공할 수 있도록 하고 있다.

본 논문에서는 결함 탐지 및 처리 기능 개선을 위해서 기존의 단일 로그 활용 방법을 개선하여 다중 로그를 활용한 탐지 방법을 개발하였다. 단일 로그 방법에 비해 다중 로그 방법은 결함 분석 효율을 높여 보다 정확한 결함 탐지를 가능케 하고 있다.

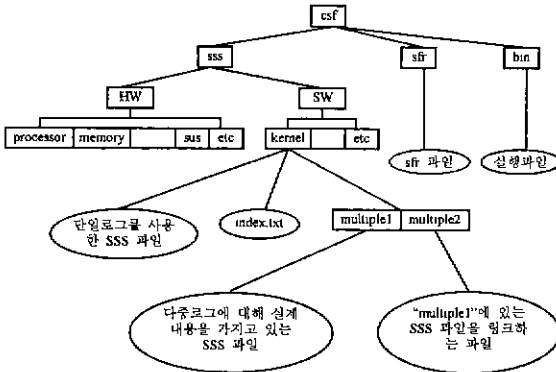
본 논문의 구성은 다음과 같다. 2장에서는 결함 분석을 위해 사용되는 결함에 대한 정보를 담고 있는 지식기반 데이터베이스의 구조를 기술한다. 3장에서는 결함의 조건과 결함분석기의 결함 탐지과정을 기술하고, 4장에서는 결론을 기술한다.

2. 지식기반 데이터베이스

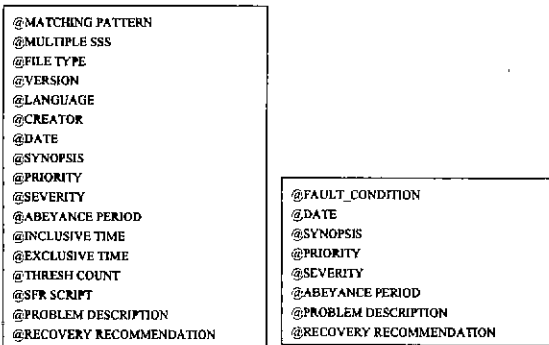
지식기반 데이터베이스의 전체 구조는 <그림 1>과 같다. 지식기반 데이터베이스는 계층적인 구조를 갖는다. 지식기반 데이터베이스는 결함 발생 원인에 따라 하드웨어 결함과 소프트웨어 결함의 두 가지로 나뉘며, 결함발생 모듈에 따라 하드웨어 결함은 processor, memory, hdd, fdd, cdrom, tape, tty, ethernet, cable, sus, etc 로, 소프트웨어 결함은 kernel, csf, miscappl, etc 로 구분된다.

지식기반 데이터베이스는 SSS(Symptom Signature Specification) 파일의 집합으로서, 단일로그용과 다중로그용으로 구성된다. 단일로그용 SSS 파일은 한 개의 로그로 결함의 조건을 기술하고, 다중로그용 SSS 파일은 여러 개의 로그로 결함의 조건을 기술한다. 각 모듈에는 "index.txt"라는 파일이 있어서 해당 모듈에 존재하는 단일로그용 SSS 파일의 종류를 기술한다. 단일로그용 SSS 파일의 "MULTIPLE SSS" 필드에 해당 로그가 결함의 조건에 포함되는 다중로그용 SSS 파일의 수를 기술함으로써 탐지하고자 하는 로그에 해당하는 다중로그용 SSS 파일을 찾는다. "MULTIPLE SSS" 필드를 사용해서 찾게 되는 SSS 파일은 "multiple2" 디렉토리에 존재하는 데, 이 SSS 파일은 "multiple1" 디렉토리에 있는 실제 SSS 파일을 링크하고 있다. 이렇게 실제 SSS 파일과 링크 파일을

분리해서 관리하는 것은 이름만 다르고 동일한 내용을 가지는 SSS 파일들이 존재함으로써 생기는 공간의 낭비를 막기 위함이다. SSS 파일 형식은 단일로그용 SSS 파일과 다중로그용 SSS 파일의 경우에 각각 <그림 2>와 같다.



<그림 1> 지식기반 데이터베이스의 구조



(a) 단일로그용 SSS 파일

(b) 다중로그용 SSS 파일

<그림 2> SSS 파일의 형식

3. 결함분석기

결함 분석기의 역할은 CSFd로부터 에러 로그를 전달받아서 결함 여부를 분석하고 그 결과를 CSFd에게 돌려주는 것이다. 결함 분석 결과는 크게 3가지로 구분되는데, 결함으로 진단된 경우 "FAULT_DETECT"로, 단순 로그일 경우 "FAULT_LOG", 그리고, 결함의 유형을 알 수 없을 경우 "FAULT_UNKNOWN"으로 설정된다. 한편 결함 분석 결과가 "FAULT_UNKNOWN"일 경우에는 Escalation Manager를 통해 원격지원센터로부터 결함 해결 방안인 SSS 파일을 다운로드하며, 이것을 이용해 결함을 분석한 후 그 결과를 CSFd에게 보낸다.

3.1 결함의 조건

결함분석기의 결함 탐지 방법은 결함 진단 시 로그 활용 형태에 따라 단일로그를 활용하는 방법과 다중로그를 활용하는 방법으로 구분되며, 각각의 경우에 결함으로 진단하는 조건이 상이하다.

3.1.1 단일로그를 활용한 방법에서의 결함조건

결함 진단 시 단일로그를 활용하는 방법에서는 특정 에러 로그가 주어진 시간동안 미리 정해진 횟수 이상 발생했는지를 검사한다. 단일로그 활용 방법에서의 결함 조건은 다음과 같다.

- ① 특정 로그가 Exclusive Time¹으로 정한 시간 이내에 발생했는지를 검사한다. 로그가 Exclusive Time 이내에 발생한 적이 있으면 이 로그 자체를 무시해버린다. 이는 같은 로그가 계속해서 발생하는 것을 막기 위한 것이다.
- ② 해당 로그가 Exclusive Time 이후에 발생했으면, Inclusive Time² 안에 Thresh Count³로 정한 횟수만큼 발생했는지 검사한다.
- ③ 해당 로그가 Inclusive Time 안에 Thresh Count 이상 발생했으면, 분석하고 있는 로그에 대한 결함이 Abeyance Period 이내에 발생한 적이 있는지 검사한다. Abeyance Period⁴ 이내에 해당 로그에 대한 결함이 발생한 적이 없을 경우에, 최종적으로 결함으로 진단한다. 이는 같은 결함이 계속해서 발생하는 것을 막기 위한 것이다.

3.1.2 다중로그를 활용한 방법에서의 결함조건

다중로그를 활용하는 방법에서는 다수의 에러 로그를 사용해서 결함을 진단하기 때문에 결함 조건을 구성하는 로그가 다수 존재한다. 다중로그 활용 방법에서의 결함 조건은 다음과 같다.

- ① 결함 조건을 구성하는 로그들이 각각의 결함 조건에 해당하는지 검사한다. 각 로그는 로그가

¹ 로그를 인정할 것인지, 무시할 것인지를 결정하는 시간 Exclusive Time 이 "1 Day"로 설정되어 있으면 로그가 발생한 이후, 하루 안에 발생한 동일한 로그는 모두 무시한다

² 로그가 일정 시간안에 일정 횟수 이상 발생할 경우 이 로그를 결함으로 진단하게 되는데, 여기서 주어진 시간이 Inclusive Time 이다

³ 로그가 일정 시간안에 일정 횟수 이상 발생할 경우 이 로그를 결함으로 진단하게 되는데, 여기서 주어진 횟수가 Thresh Count 이다

⁴ 동일한 결함이 계속해서 발생하는 것을 막기 위해 설정하는 시간이다 이 시간이 "1 Day"로 설정되어 있으면 결함이 발생한 이후 하루안에 발생한 동일한 결함은 결함으로 인정하지 않고, 단순한 로그로 인정한다

Inclusive Time 이내에 Thresh Count 이상 발생했을 경우 결함의 조건이 만족되는 것으로 본다.

- ② 결함 조건을 구성하는 모든 로그가 결함 조건을 만족하면, 해당 결함이 Abeyance Period 이내에 발생한 적이 있는지 검사한다. 현재 분석중인 결함이 Abeyance Period 이내에 발생한 적이 없을 경우에 최종적으로 결함으로 진단한다. 이는 단일로그의 경우에서와 마찬가지로 같은 결함이 계속해서 발생하는 것을 막기 위한 것이다.

3.2 결함 탐지과정

결함의 탐지는 단일로그와 다중로그의 경우로 나누어서 수행하는데 결함의 탐지 방법은 <그림 3>과 같다.

```

DetectFault()
{
    DetectSingleFault();
    for(i = 1 ; i <= MULTIPLE_SSS ; i++)
        DetectMultiFault();
}

DetectSingleFault()
{
    if(Exclusive Time내에 동일한 로그가 발생하면)
    {
        로그 무시 ;
        return ;
    }

    if(Abeyance Period내에 동일 결함 발생)
    {
        결함 분석 결과를 FAULT_LOG로 설정
        return ;
    }

    if(로그가 Inclusive Time내에 Thresh Count 이상 발생)
        결함 분석 결과를 FAULT_DETECT로 설정
    else
        결함 분석 결과를 FAULT_LOG로 설정
}

DetectMultiFault()
{
    if(결함 조건을 구성하는 모든 로그의 결함 조건 만족)
        if(Abeyance Period내에 동일 결함이 발생하지 않으면)
            결함 분석 결과를 FAULT_DETECT로 설정
}
    
```

<그림 3> 결함 탐지방법

우선 단일로그의 경우에 대해서 결함을 탐지하고, 단일로그용 SSS 파일의 "MULTIPLE SSS" 필드 값이 0 이 아닐 경우 다중로그의 경우에 대해서 결함을 탐지한다. 결함을 탐지 후 탐지된 결과는 결함분석보고

서(SFR 파일⁵⁾의 형태로 관리자에게 제시된다. 결함의 탐지 결과는 다음과 같이 결정한다.

- 단일로그에 대해서는 결함이 아닌 것으로 진단되고, 다중로그에 대해서는 결함으로 진단된 경우 최종 결과는 다중로그의 결함이 되고, 다중로그용 SSS 파일을 사용해서 SFR 파일을 작성한다.
- 단일로그에 대해서는 결함으로 진단되고, 다중로그에 대해서는 결함이 아닌 것으로 진단된 경우 최종 결과는 단일로그의 결함이 되고, 단일로그용 SSS 파일을 사용해 SFR 파일을 작성한다.
- 단일로그와 다중로그 모두에 대해서 결함으로 진단된 경우 양쪽 결함을 모두 취하게 되고, 단일로그용 SSS 파일과 다중로그용 SSS 파일을 혼합해서 SFR 파일을 작성하게 된다.

4. 결 론

본 논문에서는 시스템 동작 중 발생한 결함을 탐지, 분석 후 해결 방안을 자동으로 제시해 주는 결함 분석기를 개발하였다. 개발된 결함분석기는 단일로그를 활용한 방법을 확장하여 다중로그를 활용한 결함 탐지 방법을 채택하고 있으며, 시스템에서 발생하는 결함에 대한 정보는 지식기반 데이터베이스 내에서 SSS 파일 형태로 유지된다. 이러한 지식기반 데이터베이스의 효율적인 관리는 결함분석기의 성능에 직접적인 영향을 미친다.

본 논문에서 기술된 결함분석기는 결함 발생시 관리자로 하여금 기존의 경험이나 감각에 의존하지 않고, 보다 효율적으로 시스템에서 발생하는 결함에 대처할 수 있는 기능을 제공하고 있다.

참고 문헌

- [1] 이철훈, 유승화, "결함 허용 밀결함 멀티프로세서 시스템," 정보과학회지 제 11 권 제 3 호, pp.17-31. 1993.
- [2] 고정국, 옥을식, 윤재식, 김길용, "결함 관리 서비스를 위한 시스템 관리 기능의 설계" 한국정보처리학회 춘계 학술 발표대회 논문집, 1998 년 4 월

⁵ System Fault Report. 결함의 탐지 결과가 FAULT_DETECT 로 결정되었을 경우, 관리자에게 결함의 증상과 추천대응방안을 제시해 주는 파일