

생산 자동화 시스템 운영에서의 장애 검출 방안

Fault Detection Method on Operation of Production Automation System

홍지윤, 조광문

천안대학교 정보기술대학원

Hong Ji-Yun , Cho Kwang-Moon

Graduate School of Information Technology,
Cheonan University

요약

공장 자동화 시스템이 증가됨에 따라 인건비 감소와 생산량 증가, 효율적인 작업 운영, 그리고 고도의 정밀 작업과 인재의 감소 등 여러 가지 이점이 있지만, 자동화 시스템의 문제로 생산 작업이 중단될 경우에 공장의 생산에 미치는 영향 또한 막대하다. 본 논문에서는 공장 자동화 시스템을 효율적으로 관리할 수 있도록 사전에 시스템 장애를 모니터링 할 수 있는 장애 검출 방안에 대하여 제시하고, 이에 따른 효과를 제시하였다. 또한 제시된 방법은 시스템 내 각종 서버의 문제를 검출할 수 있고, 관리자는 모니터링 한 내용을 바탕으로 장애를 조치하는데 있어서 상당한 효과를 볼 수 있었으며, 이것이 생산량 증가로 이어지는 성과를 거둘 수 있다.

1. 서론

제품의 다양화, 제품의 대량 생산, 고객 만족을 위한 제품의 특성화, 차별화의 추세에 따라 많은 제조 공장에서는 MES(Manufacturing Execution System)를 도입하여 제품을 생산하고 있다. MES는 자동화의 중추로서 공장 내에서 작업을 수행하기 위한 스케줄링 및 작업 지시, 품질 관리, 작업 실적 관리 등의 제반 활동을 지원하고, 현장 상태의 실시간 정보 제공을 통하여 관리자와 작업자의 의사 결정을 지원하며, 조직 내의 목표를 달성할 수 있도록 제품 주문, 생산, 운영 및 실행에 이르기까지 플랜트의 전반적인 정보를 제공하는 시스템이다.

이러한 공장 자동화 시스템의 도입으로 인해, 인건비를 줄일 수 있으며, 인간의 평균 노동 시간인 8시간과 비교할 때 기계는 작동할 수 있는 조건이 부여되는 한 멈추지 않고 작업할 수 있으므로, 24시간 가동으로 인하여 업무 속도가 빨라지고, 또한 고도의 정밀 작업이 가능하며, 공정 진행 상 인내가 발생할 수 있었던 위험요소를 배제할 수 있다. 또한 제품 공정이 진행되는 상태를 실시간으로 모니터링 하여, 생산을 계획할 수 있으며, 다양한 제품을 생산함으로써 고객 만족도를 높일 수

있는 고객 중심의 차별화된 제품을 생산할 수 있는 제조 공법도 만들어 졌다.

그러나 공장 자동화 시스템은 제품의 생산에 있어 많은 이점을 가지고 작업을 수행하고 있지만 다음에서 제기되고 있는 문제점과 같은 큰 단점을 갖고 있다.

그것은 시스템을 구성하고 있는 데이터베이스, 시스템, 응용 프로그램 등 각종 컴포넌트들에 대한 장애가 발생했을 때 장애가 발생하면서부터 장애가 복구될 때까지의 시간이 많이 소요되며, 그러한 문제로 인해 제품을 생산하지 못해 많은 피해가 발생 한다는 것이다. 이것은 제품이 출하 되지 못하여 발생하는 피해에 따른 문제와, 공장 내의 각각의 많은 제조 공정에 분포되어 있는 물량에까지도 큰 피해를 발생시킬 수 있다.

그러므로 실시간 모니터링과 성능 문제 진단, 성능 튜닝 및 분석을 유기적으로 지원하는 틀을 통해서 대규모 시스템의 효과적인 성능 관리 체계의 구축이 필요한 실정이다.

본 논문에서는 효과적인 공장 시스템 운영에 있어서 생산 자동화 시스템의 운영에서의 MES 장애 조기 검출 방안에 대해서 제안하였다.

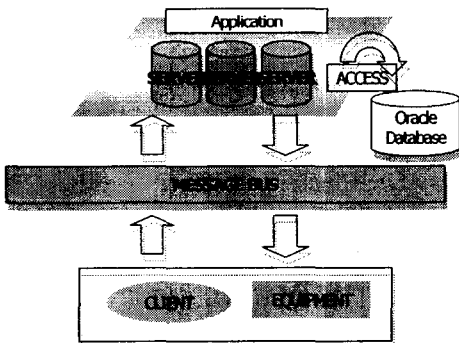
공장 자동화 시스템에 있는 각종 컴포넌트들을 사전

에 모니터링 하여, 장애가 발생하지 않도록 대비하고, 만약 장애가 발생하였을 때, 모니터링 한 데이터를 가지고 관리자가 조기에 조치할 수 있는 방법을 제시한다.

2. 시스템 운영상 발생하는 문제점

공장 자동화 시스템을 운영하는데 있어 발생하는 문제점을 크게 구분하면 기존에 운영되고 있는 시스템에 이상이 발생하는 경우와, 새로운 제조 공법상의 규칙(rule)을 적용하면서 생기는 문제점으로 분류할 수 있다. 만약 새로운 규칙을 적용하면서 장애가 발생한다면, 장애 발생 전의 상태로 원상 복구를 하면 장애가 쉽게 조치되며, 그 피해 또한 적다.

그러나 기존에 운영되던 시스템에 이상이 발생 하였다면, 어떤 시스템에서 어떤 문제로 인해 장애가 발생하였는지에 대한 원인 분석이 어려울 뿐만 아니라 그 조치 또한 쉽지 않다. [그림1]은 공장 자동화 시스템의 구조를 나타낸 것이다.



▶▶그림1. SYSTEM Configuration

그러므로 다음과 같은 관리가 필요하다.

첫째, 시스템을 구성하고 있는 응용 프로그램, 시스템, 데이터베이스 등 각 컴포넌트들에 대한 통합적인 장애 및 성능 관리를 유기적으로 수행해야 한다.

둘째, 실시간 모니터링으로 데이터베이스에서 발생하는 장애나, 응용 프로그램의 실행을 모니터링 해야 하며, 성능 문제 진단, 탐지를 통해 발견된 문제에 대한 적절한 해결책을 제시할 수 있는 해결 방법을 제공해야

한다. 그리고 성능 튜닝 및 분석을 유기적으로 지원하는 툴을 통해서 대규모 시스템의 효과적인 성능 관리 체계 구축이 필요하다. 이러한 응용 프로그램 모니터링 툴(monitoring tool)은 관리자들이 IT 조직을 관리하고 향상시키는데 낭비되는 시간을 절약하고, 장애가 발생하였을 때 그 장애 내용을 파악하고, 조치하는데 많은 도움을 줄 수 있다. [그림2]는 성능 관리의 측면에서 보았을 때, 실시간 모니터링, 성능 튜닝 및 분석, 성능 문제 진단과 같은 시스템 관리 구성에 대한 요소들을 작성한 것이다.



3. 모니터링 툴의 구성 요소

관리자들이 모니터링 툴을 사용하여 시스템 운영을 하기 위해서는 다음과 같은 기능들이 제공될 수 있어야 한다.

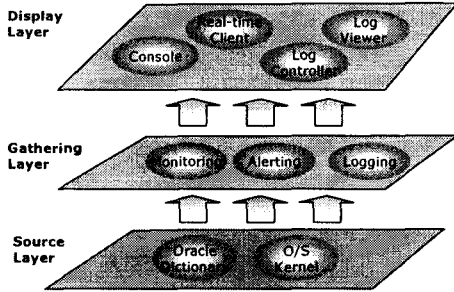
첫째, 실시간 모니터링 기능이 있어서 현재 데이터베이스에서 발생하는 장애, 에러 발생, 성능 저하 등의 여부를 항상 감시하고 그 발생을 사전에 인지하여 조치하거나 문제 발생시에 실시간 경보를 받을 수 있어야 한다.

둘째, 성능 튜닝 및 분석 기능이 제공되어, 성능문제의 원인이 되는 SQL을 튜닝하고, 실시간에 분석하지 못한 문제점을 기록된 정보로부터 상세히 분석하고 근본적인 해결책을 강구할 수 있어야 한다.

셋째, 성능 문제에서는 진단, 문제 발생시에 근본 원인이 무엇인지 신속하게 진단하고 그 문제를 조치하는 것으로 대부분 어느 세션의 현 수행 SQL을 추적할 수

있어야 한다.

[그림 3]은 모니터링 툴의 구조를 나타낸 것이다.



▶▶ 그림 3. Monitoring Tool Architecture

각 계층에서는 시스템을 원활히 관리하고자 [표 1]과 같이 각 계층마다 작업을 분산하여 실행한다.

[표 1] 모니터링 툴의 계층 기능

Layer	작업 수행 내역
Display Layer	성능 정보를 조회하거나 알람을 송수신하는 프로그램이 위치해 있는 단계로서, 실시간 모니터링과 진단, 로깅 및 분석 활동을 수행하기 위한 클라이언트를 수행
Gathering Layer	성능 정보를 수집하는 프로세스가 위치해 있는 단계로서, 각종 성능 정보를 수집하고, 설정된 임계치의 초과 여부를 검사하며, 각종 진단을 위한 정보를 수집하여 기록
Source Layer	성능 정보가 위치해 있는 단계로서, 오라클 메모리로부터 직접 읽거나 또는 오라클 디서너리로 부터 쿼리를 수행함으로써 오라클의 성능 정보를 수집. 또한 O/S 커널이 제공하는 API를 수행함으로써 O/S의 성능 정보를 수집

4. 시스템 모니터링을 위한 스크립트

다음은 시스템을 모니터링 하기 위한 방법 중에서, Gathering Layer에 속하는 부분으로 모니터링 툴에 출력(display)되는 데이터를 시스템 내에서 직접 데이터를 추출해 내는 스크립트를 작성한 것이다. 그리고 여러 스크립트를 이용해 추출된 데이터를 출력하는 모니터링 툴의 화면을 제시한다.

4.1 시스템 메모리와 CPU 사용량 추출

시스템의 메모리나 CPU 사용량은 [그림 4]와 같은 스크립트를 통해 모니터링이 가능하다.

```
set VMSTAT = 'vmstat 1 2 | tail -1 | awk '{print "' $4, $5, $6, $16, $17 "'}'
set VM_ACT = 'echo $VMSTAT | awk -F "K" '{print "' $1"'}'
set VM_FREE = 'echo $VMSTAT | awk -F "K" '{print "' $2"'}'
set VM_WIRE = 'echo $VMSTAT | awk -F "K" '{print "' $3"'}'
set VM_US = 'echo $VMSTAT | awk '{print "' $4"'}'
set VM_SY = 'echo $VMSTAT | awk '{print "' $5"'}'
```

▶▶ 그림 4. 메모리, CPU 사용량 추출 스크립트

4.2 디스크 Free 영역 추출

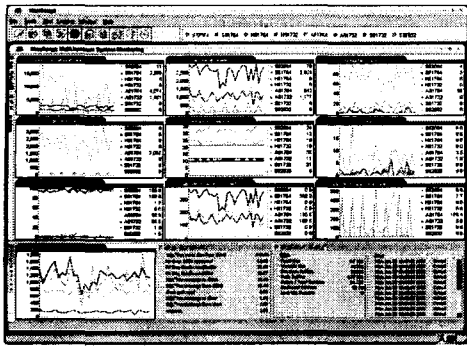
시스템에서 사용되는 각각의 디스크가 사용하고 있는 데이터의 양은 [그림 5]와 같은 스크립트를 이용하여 모니터링이 가능하다.

```
set DF = 'df /DISKNAME | awk '{print "' $5 "'}' | grep -v Capacity | awk -F '%' '{print $1}' `
set D_FREE = 'echo $DF | awk '{print "' $1 "'}'
```

▶▶ 그림 5. 디스크 Free 영역 추출 스크립트

4.3 추출된 데이터의 그래픽 표현

위와 같은 스크립트로 여러 모듈 별로 동시에 접속하여 추출된 데이터를 데이터베이스에 연결하여 데이터를 저장한 뒤, 데이터베이스에 접속하여 각 데이터베이스의 수행 현황을 단일 화면에서 비교 모니터링 할 수 있다. 또한 성능 지표와 O/S의 성능 지표를 단일 화면에서 모니터링 할 수 있다. [그림 6]은 시스템 모니터링 화면을 나타낸 것이다.



▶▶ 그림 6. 시스템 모니터링 화면

5. 결과 분석 및 결론

본 논문에서는 공장 자동화 시스템에서 모니터링 툴을 이용하여 시스템을 관리한 후, 그 결과로 여러 가지 효과를 가져올 수 있었는데 [표 2]와 같은 다섯 가지 항목을 개선할 수 있었다.

[표 2] 모니터링 툴 사용 후 개선 사항

대규모 장애 감시	수많은 서버의 장애 및 성능 현황을 한 화면에서 통합 감시함으로써 성능 관리 업무의 효율성 증대
장애 감시	업무의 중단을 가져올 수 있는 장애 상황을 실시간 감시하고 그 원인을 즉시 알람하여 조치 가능
성능 감시	업무의 저해 요소가 될 수 있는 성능 저하 현상을 실시간 감시하여 조기에 조치 가능
부하 최소화	물리적으로 분리된 기계에서 원격 에이전트가 수행됨으로써 감시 대상 시스템에 미치는 부하를 최소화
진단 및 사후 분석	실시간 진단과 사무 분석이 마우스 조작에 의해서 가능

이로 인해, 단축된 주기별로 정보의 수집이 가능하게 되었으며, 신속한 정보의 수집이 가능하여졌고, 성능 정보를 수집하기 위한 자원의 사용량이 적어 시스템을 안정되게 유지할 수 있었으며, 사용하기 편리하여 약간의 지식만 있으면 누구든지 시스템의 현황을 파악하여 그에 따른 조치를 할 수 있었다.

또한 진단 및 감시 활동이 편리하게 되어 만약 문제 발생 시 신속하게 조치할 수 있게 되어 시스템이 다운되는 시간을 줄일 수 있어서 생산이 증가되는 효과를 가져왔으며, 장애나 장애 이력을 통해 장애 대비 및 윌

별 또는 상황별 진단이 가능해지므로 지표 모니터링과 세션 모니터링이 동시에 이뤄질 수 있었다.

향후 연구 방향으로는 현재 논문에서 제시한 시스템 관리 모니터링뿐만 아니라, 데이터베이스 내의 각종 실행 영역 부분과 패키지 부분도 모니터링 하여 데이터베이스의 이상으로 인해 발생하는 장애 현상도 모니터링 할 수 있어 시스템의 장애를 막을 수 있는 연구가 이루어져야 할 것이다.

■ 참고문헌 ■

- [1] ORACLE, Introduction to Oracle: SQL*Plus and PL/SQL.
- [2] ORACLE, Administration.
- [3] ORACLE, Backup and Recovery.
- [4] COMPAQ, Command and Shell User's Guide.
- [5] COMPAQ, Compaq C Language Reference Manual.
- [6] COMPAQ, Programmer's Guide.
- [7] 장순 역, 시스템 관리의 핵심, 한빛미디어.
- [8] 조유진 역, UNIX 시스템 프로그래밍, 홍릉과학출판사.