http://software.intel.com/en-us/articles/parallel-amplifier-under-the-hood/

인텔® 패러렐 앰플리파이어를 이용한 분석 유형

최종 수정일 : 2009 년 1 월 20 일 오후 5 시 23 분(PST)
등급★★★★★

인텔® 패러렐 앰플리파이어의 세 가지 분석 유형은 프로그램 성능 정보를 여러 가지 측면에서 제공할 수 있도록 고안되었습니다. 각각의 분석 레벨에서 추가적인 정보를 수집합니다. 다음은 작동 원리입니다.

핫스팟(Hotspots)

첫 번째 분석 유형인 *핫스팟(Hotspots)*은 가장 단순한 유형입니다. 핫스팟의 실행은 (인텔® 패러렐 앰플리파이어 밖에서) 응용 프로그램을 정상적으로 실행할 때 걸리는 시간과 비슷합니다. 응용 프로그램이 실행될 때 인텔 패러렐 앰플리파이어의 데이터 콜렉터는 주기적으로 샘플을 채취합니다. 각 샘플에 대해 콜렉터는 운영 체제와 협력해 프로그램을 중단시키고 데이터를 수집합니다. 콜렉터는 응용 프로그램을 실행하는 각 CPU 코어의 명령어 포인터(IP)와 호출 스택을 기록합니다(운영 체제 데이터 구조의 일부로 저장). 응용 프로그램 실행이 종료되면 인텔 패러렐 앰플리파이어는 IP 샘플을 사용해 각 함수의 실행 시간을 파악하고 호출 스택 샘플을 사용해 전체 프로그램에 대한 호출 트리를 생성합니다. 핫스팟 목록과 호출 트리를 생성하기 위해 인텔 패러렐 앰플리파이어는 프로그램 디버그 데이터베이스(.pdb file)에 저장된 프로그램 데이터 및 명령어 공간에 대한 정보를 이용합니다.

병행성(Concurrency)

두 번째 분석 유형인 *병행성(Concurrency)*도 동일한 방식으로 작동하나 좀 더 많은 데이터를 수집합니다. 또한 병행성은 각 응용 프로그램 스레드의 상태(실행 중, 실행 대기, 차단) 데이터를 기록합니다. 모든 샘플을 수집한 후 인텔 패러렐 앰플리파이어는 데이터를 분석해 응용 프로그램 스레드가 활성화되어 있는 시간의 비율을 결정합니다. 이를 *병행성 레벨(Concurrency Level)*이라고 부르며 함수별로 세분화됩니다. 응용 프로그램에 대한 이상적인 병행성 레벨은 시스템에 장착된 프로세서의 개수입니다. 인텔 패러렐 앰플리파이어에서는 이러한 상태를 *완전 가동(Fully Utilized)*이라고 부릅니다.

잠금 및 대기(Locks and Waits)

마지막 분석 유형인 *잠금 및 대기(Locks and Waits)는* 가장 프로세서 집약적입니다. 응용 프로그램을 잠금 및 대기 분석에서 실행할 때 더 오랜 시간이 걸릴 수 있으며, 그 이유는 다음과 같습니다. 앞에서 설명한 데이터 수집에 더해 인텔 패러렐 앰플리파이어는 컴파일된 프로그램에

명령어를 추가합니다. 이러한 명령어는 프로그램이 스레딩과 동기화 API 호출을 사용할 때마다 삽입되고, 명령어의 목적은 스레드가 비활성화 상태일 때마다 경과된 대기 시간을 측정하는 것입니다. 이러한 시간 정보는 응용 프로그램의 대기 장소와 대기 목적을 설명하기 위해 .pdb 파일의 심볼 정보와 결합됩니다. 스레드의 대기를 유발할 수 있는 잠금 및 기타 구조를 *동기화 개체(Synchronization Objects)*라고 부릅니다. 이 분석을 실행한 후 인텔 패러렐 앰플리파이어는 각각에 대한 대기 시간 및 대기 중 응용 프로그램의 병행성과 함께 동기화 개체 목록을 보여줍니다.