

박필성(수원대)	
AM-10	Asynchronous Parallel iterative methods for some large sparse matrix problems on distributed systems
<p>현재까지 병렬 프로그래밍 패러다임에 있어서는 여전히 동기적 알고리즘(synchronous algorithm)이 주류를 이루는데, 이는 주로 동등한 성능의 다수 개의 CPU를 가진 하나의 병렬 시스템에서의 계산에 적합한 방법이다.</p> <p>그러나 1990년대 초반부터 네트워크 속도의 향상과 저가의 고성능 프로세서의 등장으로, 고성능 컴퓨팅 분야는 점차 고가의 상용 슈퍼컴퓨터로부터 쉽게 제작이 가능한 저가의 클러스터로 옮겨가는 경향이 일어났으며, 더 나아가서는 지리적으로 멀리 떨어진 다수의 컴퓨터를 마치 하나의 병렬 컴퓨터처럼 사용하는 GRID로까지 발전하고 있다.</p> <p>그러나 병렬 동기적 알고리즘은 올바른 연산을 위해서는(즉 직렬 알고리즘과 같은 결과를 내기 위해서) 동기화(synchronization)가 필수적이며, 작업을 일찍 끝낸 프로세서들은 동기점(synchronization point)에서 데이터 교환을 위해 느린 프로세서들을 기다려야 하므로 부하 균형이 필수적이다. 그러나 비록 같은 크기로 작업을 나눈다 하더라도 프로세서들의 성능이 서로 다르거나, 각 프로세서들의 다른 작업을 위한 부하가 각기 다르며 시시각각으로 변할 경우는 부하균형이 거의 불가능하다.</p> <p>이처럼 동기화의 필요성에 의한 연산의 지연을 해결하는 하나의 방안으로 비동기 반복법(asynchronous iteration)의 개념이 나왔으며, 지금도 활발히 연구되고 있다. 이는 알고리즘의 동기점을 가능한 한 제거함으로써 빠른 프로세서의 유휴 시간을 줄이는 것이 목적이다. 즉 비동기 알고리즘에서는, 각 프로세서는 다른 프로세서로부터 갱신된 데이터가 올 때까지 기다리지 않고 계속 다음 작업을 수행해 나간다. 따라서 동시에 갱신된 데이터를 교환한 후 다음 단계로 진행하는 동기 알고리즘에 비해, 미처 갱신되지 않은 데이터를 사용하는 경우가 많으므로 전체적으로는 연산량 대비의 수렴 속도는 느릴 수 있다. 그러나 각 프로세서는 거의 유휴 시간이 없이 연산을 수행하므로 wall clock time은 동기 알고리즘보다 적게 걸리며, 때로는 50%까지 빠른 결과도 보고되고 있다.</p> <p>본 연구에서는 각기 선형 시스템과 주요 고유치를 구하는 고유치 문제를 푸는 비동기적 반복 알고리즘을 고안하고 MPI(Message Passing Interface)를 사용하여 수원대학교의 Atom cluster에서 구현하고 동기병렬 알고리즘의 결과와 비교하였다.</p>	