

멀티-태스킹을 이용한 효율적인 Bio-Automation 시스템

정상용^o, 이상호

충북대학교 전자계산학과

goodjisy@daum.net^o, shlee@chungbuk.ac.kr

Efficient Bio-Automation System using Multi-Tasking

Sang-Yong Jung^o, Sang-ho Lee

Chungbuk National University

요 약

최근의 생물학 연구는 "시스템 생물학 또는 시스템 생명공학"으로 발전하면서 생명체 분석의 속도와 용량이 상상을 초월할 만큼 발전하고 있으며, 이것이 성공여부 판가름의 중요한 변수로 작용함에 따라 다량의 샘플들을 빠른 시간 안에 처리하기 위한 high throughput bio-automation system이 속속 개발되고 있다. 이 연구에서는 현재까지 개발되어 상용화 된 high throughput bio-automation system들의 멀티-태스킹에 있어서의 문제점을 지적하고 개선된 합리적인 멀티-태스킹을 제시하였다.

1. 서 론

2001년 2월 12일, 미국 워싱턴과 영국 런던 등에서 미국 국립보건원(NIH:National Institutes of Health)과 영국의 웰컴트러스트생어센터(the Wellcome trust and the Sanger center)를 중심으로 한 인간 유전체 프로젝트(HGP:Human Genome Project) 및 미국의 생명공학 벤처 기업인 셀레라 지노믹스(Celera Genomics)에 의해서 인간 유전체 지도가 발표되었다. 당초 2005년이 목표였던 인간 유전체 프로젝트가 무려 4년이나 빨리 발표될 수 있었던 것은 셀레라 지노믹스에서 고안한 지놈샷건법(whole genome shotgunning)과 막대한 자본을 들여 구비한 자동염기서열분석기(automatic DNA sequencer), 그리고 이들의 데이터를 분석, 조합해낸 슈퍼컴퓨터의 역할 때문이었다. 이렇듯 생명체 분석의 속도와 용량이 상상을 초월할 만큼 발전하고 있으며, 이것이 성공여부 판가름의 중요한 변수로 작용함에 따라 다량의 샘플들을 빠른 시간 안에 처리하기 위한 HTBAS(high throughput bio-automation system)가 속속 개발되고 있으며, 이들의 성공으로 인해 기존에 튜브 타입을 이용해서 사람이 손으로 하나의 샘플을 처리하는데 50~60분 정도가 소요되었던 일들에 속도 향상 및 효율성 향상까지도 제공 받을으로써 BAS(bio-automation system)들은 생명 공학 연구에 없어서는 안 될 중요한 장비들로 자리 매김을 하고 있다. 게다가 최근의 생물학 연구는 "시스템 생물학(systems biology) 또는 시스템 생명공학(systems bioengineering)"으로 발전되어 가고 있다. 즉, 기존의 분자 단위의 연구 틀에서 벗어나 단위 생물 개체를 총체적으로 접근할 수 있는 시스템적 연구를 요구하고 있으며, 연구 방법 또한 소량의 실험재료로서 다량의 실험을 동시에 수행할 수 있는 새로운 공학적 접근을 필요로 하고 있다[1].

이 연구에서는 현재까지 개발되어 상용화 된 HTBAS 중 Prep. machine¹⁾을 대상으로 멀티-태스킹(multi-tasking)에 있어서의 순차적 단순 반복이라는 문제점을 지적하고 해결

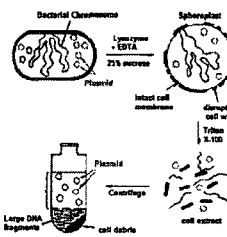
하기 위한 알고리즘을 제시하여 기 개발된 기기의 H/W적인 변형 없이 새 알고리즘을 적용한 제어 소프트웨어의 개선만으로 합리적인 멀티-태스킹(reasonable multi-tasking)을 이루어 기 개발된 기기의 효율성을 높이고 앞으로 개발될 BAS에게는 합리적인 멀티-태스킹의 방향을 제시하여 시스템의 성능 향상과 생명체 분석의 속도 개선에 이바지할 수 있는 방법을 제시한다.

이 논문의 구성은 다음과 같다. 2장에서는 Alkali lysis method 실험법의 수작업 실험을 알아보고 이를 토대로 자동화를 위한 기계적 동작으로 변형하여 기술한다. 3장에서는 기존의 장비들의 문제점을 제시하고 4장에서는 합리적인 개선 방향을 제시한다. 그리고 5장에서 결론을 맺는다.

2. 관련 연구

이장의 구성은 다음과 같다. 2.1절에서는 Alkali lysis method[2]를 기술한다. 2.2절에서는 플라스미드 DNA를 얻기 위한 Alkali lysis method의 수작업을 기술하고 2.3절에서는 2.2절의 수작업을 토대로 자동화를 위한 기계적 동작들의 표현으로 옮긴다.

2.1 Alkali lysis method



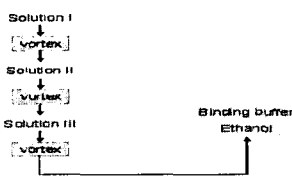
[그림 2.1] 플라스미드 추출

sucrose나 glucose가 첨가되어 세포막은 유지되도록 하면서 세포벽만을 선택적으로 균등균등한 부서지도록 하여, 크기가 큰 chromosomal DNA가 너무 잘게 조각나는 것을 방지하면서 용액에 DNA를 노출시킨다. 이때 바로 alkali 용액에 노출된 DNA는 모두 denature(변성:DNA 이중가닥이 풀리는 것)되지만 크기가 작은 supercoiled 플라스미드 DNA는 그대로 침착하게 되지 않는 성질을 이용한다. pH를 급격히 7.0까지 복구시키면 크기가 작고 supercoiled form을 유지하고 있던 플라스미드는 비교적 renaturation이 잘 되지만, chromosomal DNA는 미처 renature되지 못하고 영기게 되고 이렇게 얻인 chromosomal DNA를 원심분리로 제거하고 플라스미드 DNA만을 얻는다.

1) Prep. machine 은 비교적 실험과정이 간단하며 Alkali lysis method로 실험방법이 보편화되어 있는 플라스미드 추출 과정을 자동화한 기계이다.

2.2 Plasmid DNA extraction using Alkali lysis method

2.1절에서 설명한 것과 같은 방법이 플라스미드 DNA를 얻기 위한 보편적으로 사용되는 Alkali lysis method이며 이를 수행하기 위해 실제적으로 실험실에서 이루어지는 수작업을 기초로 간략하게 표현하면 다음과 같이 진행된다. 2)



[그림 2.2] 플라스미드 추출 순서

solution 1, solution 2, solution 3은 산과 염기를 이용해 박테리아의 세포벽과 세포막을 부수는 과정이며 filtration 과정에는 trapping plate와 binding plate를 사용 solution 1, 2, 3 과정을 거친 전체 시약을 trapping plate에 옮기고 binding plate를 밑에 놓고 그 위에 trapping plate를 올려 놓고 진공을 걸어주면 trapping plate에는 분자의 크기가 큰 것만 남게 되고 DNA를 포함하여 작은 분자들은 binding plate로 옮겨지게 된다. 그러면 trapping plate는 버리고 binding plate에 에탄올을 넣고 다시 진공을 걸어준다. 여기서 에탄올을 사용하는 것은 여러 가지가 섞여 있는 시약에서 플라스미드 DNA만을 뽑아 내기 위해세척을 하는 과정으로 여기서 다시 진공을 걸어주면 다른 물질들은 밑으로 다 빠져나가게 되고 binding plate에는 DNA 만이 남게 된다. 그러면 증류수로 binding plate에 남아있는 DNA를 녹여서 dispenser로 여기에 남아 있는 시약들을 result plate에 옮기게 되고 result plate에는 원하는 DNA만을 갖고 있는 플라스미드 DNA만을 추출할 수 있게 된다.

2.3 Plasmid DNA extraction sequence using Alkali lysis method

2.2에서 설명된 것과 같은 과정을 좀 더 기계적인 수행에 가깝게 하나 하나의 기계적 동작의 집합인 setp별로 나누어 간략히 표현하면 다음과 같고 실제적으로 기계를 제어하기 위한 명령은 다르겠지만, 수행과정에 있어서는 크게 벗어나지 않는다.

Step I

- Dispense solution I into sample pellet plate: 200uL, depth 0.0
- Vortex : 1500 rpm, 3 min
- Vortex : 1500 rpm, 5 sec
- Transfer sample pellet plate to trapping plate: 800uL
- Vortex : 1500 rpm, 5 sec
- Dispense solution II into trapping plate: 200uL
- Vortex : 1000 rpm, 25 sec
- Dispense solution III into trapping plate: 200uL
- Vortex : 1300 rpm, 13 sec

Step II

- Wait reaction : 6min 40 sec
- Filtration trapping plate into binding plate in vacuum block : 10 min

Step III

- Filtration binding plate : 70 sec
- Dispense WB into binding plate
- Filtration : 70 sec
- Dispense ethanol into binding plate

2) 실험은 바이오니아사에서 생산되는 AccuPrep™ Plasmid Extraction Kit를 이용해 이루어졌으며, 이를 바탕으로 실험과정을 기술하였음.

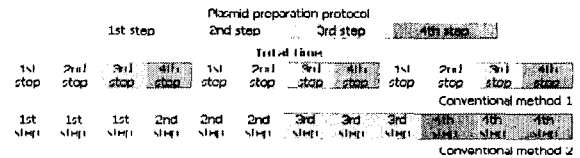
-Filtration (Dry) : 10 min

Step IV

- Dispense EL into binding plate
- Wait reaction : 5 min
- Filtration binding plate to 96 well plate : 2 min

3. Problems of bio-automation systems

이 장에서는 현재 사용되는 시스템들의 일반적인 수행과정(플라스미드 DNA를 한번 추출)에 있어서의 문제점을 설명한다.



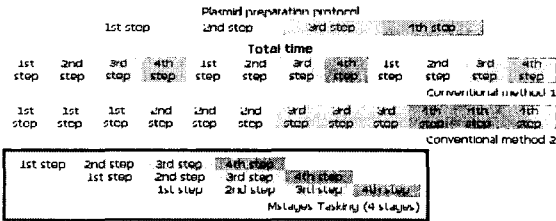
[그림 3.1] 현재 사용되는 시스템의 실행과정

현재 사용되는 시스템이 플라스미드 DNA를 추출하기 위해서는 2.3절의 전체 실행과정을 한 번 이상 수행해야 하는데, [그림 3.1] 시스템에서 플라스미드 DNA추출을 한 번 수행하는 과정인 step I ~ step IV까지의 수행에는 별 문제가 없어 보인다. 하지만 2.3절의 전체 실행과정을 2회 이상 반복 실행할 경우 현재 사용되고 있는 시스템들은 step I ~ step IV까지를 n번 반복 실행하거나, step I ~ step IV의 각 step을 n번 반복실행 하는 simple reaction 과정을 보인다. 따라서 전체 수행 시간은 2.3절의 전체 실행과정을 한번 수행하는 데 걸린 시간(step I ~ step IV) * n이 된다. 현재 시스템들이 대부분 상업적 목적으로 멀티-테스팅을 한다고 홍보되고 있지만 어디까지나 simple reaction 이지멀티-테스팅을 하고 있는 것은 아니다. 하지만 대부분의 시스템들이 시스템을 이루고 있는 각 요소들은 동시 동작이 얼마든지 가능하다. 그러나 올바르게 못한 스케줄링으로 인해 2회 이상 반복 작업을 하거나 많은 수의 작업을 수행할 경우에는 엄청난 시간과 비용을 낭비하고 있다. 따라서 동시 동작이 가능한 전체 시스템의 요소들을 철저히 관리하고 동시에 작업 할 수 있도록 자원관리와 작업 스케줄링의 문제점을 개선하여 전체시스템의 생산성과 효율성을 높여야 한다.

4. Reasonable bio-automation systems

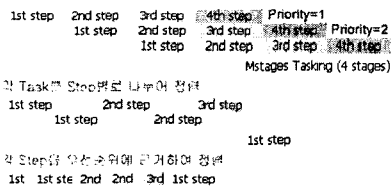
플라스미드 DNA를 추출하기 위한 2.3절의 전체 실행과정을 task라 한다면 한 번 이상 task를 수행할 때 현재 시스템들은 step I ~ step IV까지를 n번 반복 실행하거나, step I ~ step IV의 각 step을 n번씩 반복실행 하는 step 단위의 순차적 단순 반복 과정을 보이고 있는데 task의 실행과정에 있어서 같은 task내의 step의 blocking을 허용할 수는 없지만 다른 task에 관해서는 서로 독립적으로 병행 처리해도 플라스미드 DNA를 추출해 내는 전체 순서에는 아무런 영향을 주지 않으며 [그림 4.1]과 같이 플라스미드 DNA를 추출하기 위한 전체 시간은 현저하게 줄일 수 있게 된다. 하지만 task에서의 각 step별 사용 자원을 보면 step I에서는 Dispensor와 Vortex, step II에서는 Vacuum과 반응 대기상태, step III Dispensor와 Vacuum, step IV와 Dispensor와 Vacuum으로 병행 처리시 서로 다른 task들

의 step들이 같은 자원을 서로 중복 사용하기를 위하여 자원충돌 현상이 발생할 수 있게 된다.



[그림 4.1] 현재 시스템이 개선된 실행과정

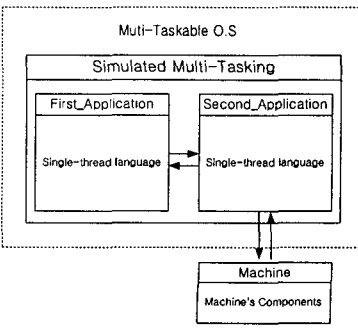
여기서 발생하는 문제는 [그림 4.2]와 같이 n개의 task를 하나의 total-task로 실행시간의 순서와 task 순서에 의한 우선순위로 정렬하여 하나의 task로 통합한 뒤 실행 시에 현재 step이 실행 중일 때 다음 step이 이용할 자원이 대기 상태에 있는지 체크하여 실행 가능할 때에만 병행 실행함으로써 자원충돌 현상을 해결할 수 있게 된다. 더 나아가 step안의 동작단위들을 살펴보면 이 역시 정해진 순서에 의해 앞 순서의 동작이 끝나면 다음 동작으로 이어져 오직 한 번에 한 자원들만이 동작하고 동작하지 않는 나머지 자원들은 대기 상태에 있게 되는데 step



[그림 4.2] 프로그램의 아키텍처

안의 동작에 대해서도 동작단위 total-task를 만들어 step 단위의 병행처리[5]와 같이 실행한다면 step단위 병행 실행 시 보다 플라스미드 DNA를 추출하는데 있어 많은 시간을 줄일 수 있게 될 것이다.

total-task로 스케줄링 된 task들을 실행하는 프로그램의 개발에 있어서는 [그림 4.3]와 같이 사용자가 원하는 일을 할 수 있도록 task를 만들고 시스템을 병행 제어할 수 있도록 전체 task를 step단위로 스케줄링 하는 first_application과 first_application에서 스케줄링 된 step을 받아 다시 한 번 동작단위로 스케줄링하고 이를 수행할 수 있도록 시스템의 제어를 담당할 second_application으로 나누어 개발하여 두 개의 독립 프로그램이 멀티-태스킹을 지원하는 운영체제 위에서 두



[그림 4.3] 프로그램의 아키텍처

언어로의 개발이 효과적일 것이다. 또한 개발될 합리적인 멀티-태스킹 프로그램이 기존의 시스템에 이식될 경우에는 하드웨어 제어 블록의 프레임워크를 Adapter 설계 패턴 [4]이나 Bridge 설계 패턴 [4]에 따라 개조, 확장하는 방법을 제안하며 이는 확장된 클래스를 통하여 어떤 시스템에 이식되더라도 하드웨어에 종속적인 하드웨어 제어블록만을 쉽게 분리하여 수정함으로써 사용 가능하게 될 것이다.

5. 결론

이 연구에서는 HTBAS의 필요성과 실험 중 비교적 과정이 간단하며 Alkali lysis method로 실험방법이 보편화되어 있는 플라스미드 추출 과정을 살펴보고 이를 자동화한 Prep. machine을 가지고 현재까지 개발되어 상용화 된 HTBAS들이 지니고 있었던 멀티-태스킹에 있어서의 순차적 단순 반복이라는 문제점을 지적하고 개선된 합리적인 멀티-태스킹을 제시하였다.

합리적인 멀티-태스킹에서는 전체 task를 step과 step내의 동작들을 두 개의 독립 application으로 나누어 계층적 스케줄링을 하도록 설계방향을 제시하였는데 이는 기존의 시스템에 비해 처리 시간을 1/4가량으로 단축시킬 수 있을 것으로 기대하며 기존 시스템에 새로 개발될 프로그램을 이식할 경우에는 하드웨어 제어 블록의 프레임워크를 Adapter 설계 패턴이나 Bridge 설계 패턴에 따라 개조, 확장하는 방법을 제안하여 확장된 클래스를 통하여 어떤 시스템에 이식되더라도 하드웨어에 종속적인 하드웨어 제어블록만을 쉽게 분리하여 수정함으로써 사용 가능할 것으로 기대된다.

이 연구에서는 현재까지 개발되어 상용화 된 HTBAS들의 멀티-태스킹에 있어서의 문제점의 지적과 멀티-태스킹의 개선 방향만을 제시하였지만 향후 실질적인 멀티-태스킹의 설계와 구현을 하고 시스템에 이식하여 전체 시스템의 성능을 증명할 예정이다.

참고문헌

- [1] 박재균, "나노바이오공학의 이론과 응용," 바이오매거진, 생명공학연구원, 2004
- [2] Birnboim, H.C. and Doly J, "A rapid alkaline 추출 procedure for screening recombinant Plasmid DNA," Nucleic Acids Res 7, 1513-1523
- [3] Lee KO et al. The Korean Journal of Biomedical Laboratory Sciences 1996; 2: 223-229
- [4] E. Gamma, et al., "Design Patterns: Elements of Reusable Object-Oriented Software," Addison-Wesley, 1995
- [5] Al-Rawi, Chazi A, "Iterative processing: From applications to parallel implementations," Stanford University, 2003
- [6] John Regehr, Alastair Reid, "Evolving real-time systems using hierarchical scheduling and concurrency analysis," In proceedings of the 24th IEEE real-time systems symposium (RTTS2003), pages 25-26, Mexico, December 3-5 2003