고성능 클러스터와 분산 병렬 파일 시스템을 이용한 유전체 데이터 전처리 작업의 효율적인 병렬화 기법

변은규, 문지협, 곽재혁
한국과학기술정보연구원
e-mail : ekbyun@kisti.re.kr

An Efficient Parallelization Mechanism for Preprocessing of Genome Sequence Data on HPC environment

Eun-Kyu Byun, Ji-hyeob Mun, Jae-Hyuck Kwak
Korea Institute of Science and Technology Information

요 약

차세대 염기서열 분석법이 생성한 유전체 원시 데이터를 기존의 방식대로 하나의 서버에서 분석하기 위해서는 수십 시간이 필요할 수 있고 이러한 시간을 최대한 줄여야 하는 응급 상황도 존재한다. 따라서 본 연구에서는 고속의 네트워크로 연결되고 병렬 파일 시스템을 공유하는 서버 클러스터를 활용하여 분석 시간을 크게 단축 시킬 수 있는 유전체 데이터 분석의 전처리 프로세스의 병렬화 방법을 제안한다. 기존의 검증된 분석도구를 기반으로 프로세스의 병렬화, 데이터의 분배 및 병렬 병합 기법을 개발하였고 실험을 통해 성능을 향상 시킬 수 있음을 증명하였다.

1. 서론 및 배경

차세대 염기서열 유전체 분석법(Next Generation Sequencing, NGS)의 발전으로 인해 유전체 정보를 보다 저렴한 가격과 적은 시간을 들어 읽어 낼 수 있게 되었다.[1] 이렇게 얻은 유전체 데이터를 분석하여 질병의 진단, 예방 등에 활용할 수 있다. 이러한 분석 과정은 유전체 정보를 기계를 통해 읽어내는 작업 뿐 아니라 수백 기가바이트에 달하는 데이터를 분석하는 과정이 필요하고 기존의 단일 서버를 사용하는 방법을 사용했을 때 수십 시간이 걸리는 경우도 일반적이다. 이러한 처리 시간은 일상적인 검사나 연구의 경우에는 크게 문제가 되지 않지만 응급 환자에게 적합한 처치를 해야 하는 상황에는 치명적일 수 있다. 따라서 더 많은 전산 자원을 활용하더라도 이 시간을 단축 할 수 있는 기술이 필요하다. 본 논문에서 많은 유전체 데이터를 분석에서 공통적으로 쓰이는 데이터 전처리 프로세스를 병렬화 하는 방법을 제안한다. 병렬화된 분석 프로세스는 고속의 네트워크로 연결되고 병렬 파일 시스템을 공유 하는 복수의 서버들 위에서 동작하여 실행시간을 크게 단축 시킬 수 있다.

NGS 기법을 통한 유전자 분석 과정은 여러 단계의 데이터 처리 프로세스로 구성된 파이프라인으로 이루어진다. 먼저 세포로부터 염기서열을 읽어야 한다. 하지만 각 염색체 혹은 RNA 염기서열 전체를 한꺼번에 직접 읽어내는 것이 불가능하다. 따라서 시퀀서(Sequencer)라는 장비를 이용하여 유전체를 작은 다수의 리드(Read)로 조각 내고 이를 증폭하여 화학 및 광학적인 방법을 사용하여 염기쌍(base pair)들을 순서대로 읽어내어 원시 데이터 파일을 생성한다. 분석을 위해서는 조각나기 전의 원래 염기서열 정보를 리드 데이터들로부터 재조합 해야 한다. 이러한 과정을 데이터 전처리 과정이라 하며 대부분의 유전정보 분석 기법에서 공통적으로 필요로 한다. 전처리 결과의 기술 방법 또한 SAM, BAM 이라는 파일 형식으로 표준화되어 있다.[2]

가장 먼저 각 리드들이 실제로 유전자중 어느 위치에 해당하는 지를 알아야 한다. 이를 위해 참고 게놈(reference genome)의 염기서열에 리드들의 데이터를 비교하여 높은 확률로 일치하는 위치를 찾는 과정을 거친다. 이 과정을 지역 정렬(alignment)이라 하며 많은 계산 시간과 메모리를 필요로 하지만 전처리 중 필수적인 단계이다. 지역 정렬을 위해서는 여러 알고리즘 중 Burrow-Wheeler Aligner(BWA)[3]가 널리 쓰이고 있다.

다음으로 중복 검출 단계가 필요할 수 있다. 시퀀서에서 NGS 생성하는 과정 중 중합 효소 연쇄반응 단계에서 같은 데이터가 여러 차례 발생할 수 있다. 이러한 데이터가 분석에 왜곡을 가져올 수 있으므로 중복으로 판단되는 리드를 표시해 둘 필요가 있다. 데이터를 변경하는 것이 아니라 플래그를 추가하는 것이므로 분석 응용의 필요에 따라 활용 여부를 결정할 수 있다.

마지막으로 분석 도구가 원하는 위치의 유전체 정보에 빠르게 접근 할 수 있도록 색인화 하는 작업이 필요하다. 이를 위해서 각 리드를 참고 게놈 내에서의 위치를 나타내는 전역 좌표를 기준으로 정렬해야

한다. 그 이후에 ascii 형식으로 되어 있는 파일을 binary 형식으로 인코딩하고 압축하는 과정을 마친 후 색인 정보를 생성해야 한다.

본 연구에서는 가장 널리 쓰이는 유전체 원시 데이터인 일루미나사의 시퀀서에서 생성하는 paired-end 리드 데이터를 대상으로 삼았다.[4] 이 원시 데이터를 지역 정렬, 중복 검출, 색인 하는 전처리 과정 각각을 병렬화 하고 그에 필요한 데이터 전달 기법을 개발하였다. 데이터의 신뢰도를 높이기 위해 프로세스의 병렬화, 데이터의 분배 및 병합 기법은 추가로 개발하였고 각 단위 프로세스에서의 데이터 처리는 기존의 검증된 도구들이 수행하도록 하였다. 데이터 처리 도구로는 BWA, samblaster[5], samtools[6] 가 각각의 단계에서 사용되었다. 작업 및 데이터 병렬 처리의 실제 구현에는 서버 간의 소켓 통신, 병렬 파일시스템인 LustreFS 를 이용한 MPI-IO 및 Hadoop on Luster 기술 등을 활용하였다.

유전체 분석 파이프라인을 구성하는 각 단계를 병렬화하는 연구 또는 특정 분석을 위한 전체 파이프라인을 병렬화하는 연구 결과는 발표 된 적이 있다.[9] 그러나 다양하게 쓰일 수 있는 NGS 데이터의 전처리 과정 전체를 고성능 클러스터와 병렬파일시스템을 활용하여 병렬화하고 최적화하는 것은 우리의 지식 범위 내에서는 처음 시도되었다.

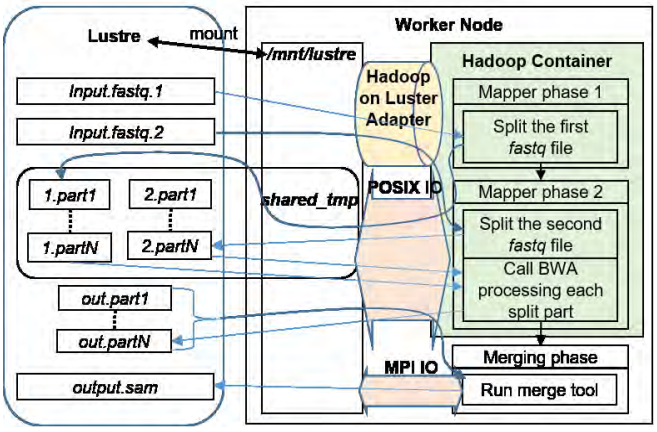
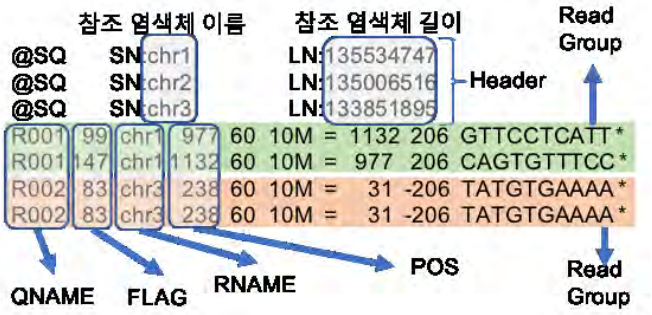


그림 1 KBigBWA 의 실행과정

2. 유전체 데이터 전처리 과정의 병렬화 기법

본 연구팀은 선행연구를 통해 KBigBWA 라는 이름의 BWA 의 병렬화 기법을 제안하였다[8]. KBigBWA 는 Hadoop 을 이용하여 BWA 를 병렬화 한 기존의 BigBWA[9]의 문제점을 개선하여 성능을 크게 향상시켰다. Hadoop 을 통해 데이터를 자동으로 분할하고 분산된 노드에서 병렬로 처리할 수 있지만 한 쌍의 입력 파일을 합쳐서 HDFS 로 전송하는 비용이 매우 크다. 그 문제를 해결하기 위해 입력, 출력, 중간 데이터를 공유 파일시스템인 Lustre 에 두고 Hadoop on Lustre 어댑터를 이용하여 접근하게 하였다. 또한 MapReduce 를 2 단계로 개선하여 paired read 를 처리할 수 있게 하였으며 분산되어 생성된 SAM 파일들을 병렬 IO 를 이용하여 빠르게 하나의 파일로 합치는 기능도 개발하였다. 그림 1 에 KBigBWA 의 실행 단계가 나타나 있다.

지역 정렬 단계가 완료되면 SAM 형식의 파일이 생성된다. 그림 2 의 예제와 같이 SAM 파일에는 참조 게놈의 염색체의 이름과 길이 정보가 헤더에 나타나고 각 리드의 정렬(alignment) 정보가 매 줄 마다 기록되게 된다. 이 중 RNAME 값과 POS 값을 이용하면 해당 리드가 참조 염색체 중 어느 위치에 정렬 되어 있는지 알 수 있다. 헤더에 적힌 참조 염색체의 순서와 길이 값을 누적하면 각 참조 염색체의 오프셋을 계산할 수 있고 이 값과 POS 값을 더해서 정렬된 위치의 전역 좌표를 결정할 수 있다. 중복 검출과 색인 단계에는 이 전역 좌표 값을 사용한다.



(그림 2) SAM 파일 예제

그림 2 에서 QNAME 값은 시퀀서에서 읽은 각 조각의 ID 를 나타낸다. Paired-end read 의 경우 데이터는 각 유전자 조각을 양쪽에서 일정한 크기 만큼 읽어내어 생성한 데이터이므로 동일한 QNAME 에서 리드 정렬 정보가 최소한 2 개 존재한다. 또한 참조 게놈에 정렬 되는 위치가 하나 이상인 경우도 있을 수 있다. 이경우 가장 확률이 높은 정렬 위치의 쌍을 Primary line 이라고 한다. 이렇게 같은 QNAME 을 가지는 여러 개의 리드 정렬 정보의 집합(이후 Read Group 으로 표현함)은 동일한 염기서열 조각에서 생성된 데이터들로 중복 검출 시 함께 처리되어야 한다.

Samblaster 는 SAM 파일을 입력으로 받는데 이 때 같은 Read Group 에 속한 리드들은 반드시 연속된 라인으로 함께 입력되어야 한다. 또한 samblaster 알고리즘은 Read Group 의 primary line 의 전역 좌표 값을 비교하여 중복을 판단한다. 따라서 중복 검출의 누락을 방지하기 위해 이 조건을 고려하여 병렬화를 위해 데이터를 재분배 기법을 설계하였다. 각각의 노드에서 병렬로 실행된 BWA 는 SAM 형식으로 결과를 출력하는데 Read Group 이 연속되어 기록된다. 이 결과물을 읽어서 Read Group 별로 구분하고 primary line 의 전역 좌표 2 개 중 작은 값을 Key 값으로 read group 의 정렬 정보 전체를 value 값으로 결정한다. 그 Key 값을 기준으로 데이터를 분배(partitioning)하여 samblaster 를 수행하는 노드에 N-to-M 전송한다. 이 과정을 통해

각각의 samblaster 프로세스에는 참조 게놈의 특정한 범위에 정렬된 유전자 조각들이 모이게 된다. 입력된 SAM 형식의 데이터에서 중복으로 판단된 리드의 FLAG 값만을 수정한 내용이 결과로 출력된다.

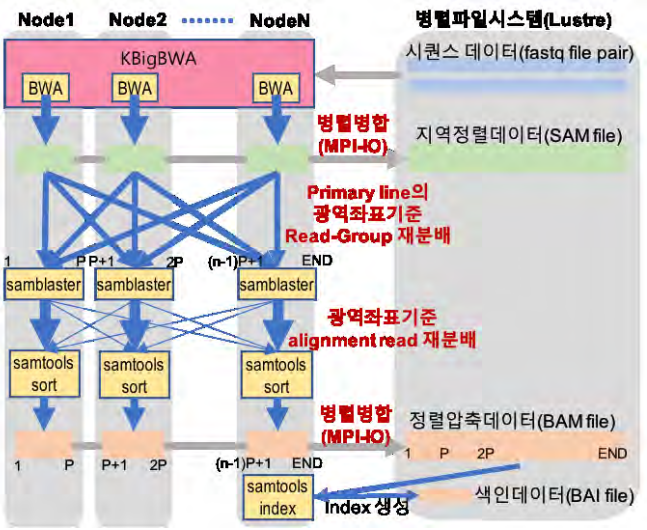
Primary line 의 광역 좌표를 기준으로 파티션을 결정하였기 때문에, non-primary 리드들의 좌표가 파티션과 일치하지 않는 경우가 생길 수 있다. Samblaster 수행 후 이러한 리드들에 대해 광역 좌표를 기준으로 맞는 위치의 파티션에 전송하고, 현재 파티션과 일치하는 리드들은 로컬 노드 내에서 다음 단계의 입력으로 직접 전달한다. 몇 가지 데이터 셋을 통해 측정해 본 결과 이 실제로 이렇게 재분배 되는 비율은 1%전후로 적은 양이어서 소요시간이 매우 짧았다. 설령 대부분이 데이터가 재분배되는 경우에도 samblaster 별렬화를 위해 수행했던 재분배 프로세스에서 소요되는 시간과 비슷한 수준의 시간이 소요 될 것이므로 전체 수행 시간에 큰 영향을 주지 않을 것으로 판단된다.

색인단계에서는 먼저 모든 리드 정보를 전역 좌표를 기준으로 오름차순 정렬(sort)을 해야 한다. 수억 개의 리드 정보를 한꺼번에 정렬 하는 것은 엄청난 시간을 필요로 한다. 그러나 이미 samblaster 병렬화 단계에서 전역 좌표를 기준으로 데이터가 분배되어 있다. 이렇게 분배되어 있는 데이터 집합들 각각을 병렬로 정렬(sort)하고 난 후, 각각의 데이터 집합들을 순서에 맞게 연결시키기만 하면 전체 정렬(sort)이 완성된다.

Samtools 의 sort 기능을 활용하면 sort, encoding, compress 가 한꺼번에 수행되고 결과물로 BAM 파일이 생성된다. 이렇게 각각의 samtools 프로세스에서 생성된 BAM 파일을 하나의 전체 BAM 파일로 연결하는 기법을 개발하였다. BAM 파일은 gzip 형식으로 으로 압축된 블록들로 구성되며 첫 블록은 헤더의 내용을 담고 있고 마지막 블록은 EOF 블록이다. BAM 파일을 정상적으로 합치기 위해서는 연결부위의 헤더와 EOF 블록을 잘라내야 한다. 손질된 파일의 연결은 MPI-IO 를 이용하여 N-to-1 병렬 쓰기 작업을 통해 이루어진다. 따라서 최종 파일의 크기가 크더라도 여러 노드가 동시에 쓰기를 수행하기 때문에 소요 시간이 크지 않다.

마지막으로 합쳐진 BAM 파일에 대하여 samtools 의 index 기능을 이용하여 색인 파일을 생성한다. 이 과정은 전체 수행시간대비 수행시간이 크지 않고 samtools 의 알고리즘 상 기존의 도구를 그대로 사용하면서 병렬화 하기 불가능 하였기 때문에 본 논문의 연구 범위에서는 병렬화 작업을 진행하지 않았다.

그림 3 은 본 논문에서 제안한 NGS 데이터의 전처리 병렬화 기법의 전체 수행 과정을 정리하여 나타낸다. 파란 화살표는 데이터의 전달 과정을 나타내며, 회색 화살표는 MPI-IO 를 이용한 병렬 쓰기를 나타낸다. 노란 박스로 표시된 부분이 병렬화 한 기존 도구들이며, 붉은 색으로 표시한 부분이 새롭게 개발한 기능들이다. 시퀀서에서 생성한 원시 데이터가 fastq 파일의 쌍으로 주어지면 KBigBWA 가 Hadoop on Lustre 를 이용해 지역 정렬 데이터를 각각의 노드에 생성한다. 이 데이터는 Read Group 별로 primary line 의 광역 좌표를 기준으로 재분배되고 각각의 노드에서 samblaster 를 이용해 중복 리드를 검출한다. 이후 각 리드의 광역 좌표를 기준으로 위치 조정을 다시 수행한 후 각각의 노드에서 samtools sort 를 이용해 정렬 압축된 BAM 파일을 생성한다. 이들을 병렬 병합 기능을 이용해 하나의 BAM 파일로 합치고 index 파일까지 생성하면 NGA 데이터의 전처리 과정이 완료된다. 결과물로 지역정렬데이터 SAM 파일 하나 지역 정렬, 중복 검출, 정렬 및 압축이 모두 완료된 BAM 파일 하나와 그 index 파일 하나가 생성되고 이는 향후의 유전체 분석에 활용될 수 있다.



(그림 3) NGS 데이터 전처리 병렬화의 전체 수행 과정

3. 성능

위에서 설명한 병렬화 테스트베드 위에서 구현하고 실행하여 성능을 측정하였다. 테스트베드로는 FDR Infiniband 로 연결된 여덟 대의 계산 노드와 2 대의 스토리지 노드로 구성된 Lustre 파일시스템을 사용하였다. 각 계산 노드에는 듀얼 소켓 10 core Xeon E5-2650 CPU 와 80GB 메모리가 설치되어 있으며, 각각의 스토리지 노드에는 40 개의 하드디스크가 4 개의 RAID6 스토리지 타겟으로 구성되어 있다.

아래 표 1 은 크기가 208GB 인 유전체 리드 원시 데이터를 전처리 하는데 소요되는 시간을 보여주고 있다. 표의 왼쪽 부분은 기존의 방법과 동일하게 하나의 노드에서 BWA, samblaster, samtools 를 연속으로 실행할 때의 소요 시간을 나타낸다. 이 때, 멀티 스레드 옵션을 활성화하여 20 개의 코어를 최대한 활용하도록 하였다. 오른쪽의 결과는 본 논문에서 제시한 병렬화 기법을 이용하여 8 대의 노드에서 병렬로 수행하였을 때의 결과를 나타낸다. BWA 는 KBigBWA 를 통해 Hadoop on Lustre 를 통해 병렬로 수행되며 samblaster 와 samtools 는 GNU parallel 을 이용하여 각각의 노드에서 작업을 나누어 처리한다. 각 단계 사이에서의 데이터 재배치 및 전송은 IPoIB 를 통해 이루어지고, 파일을 병합하는 과정은 MPI-IO 를 이용해

병렬화 하였다. 결과물로 생성된 SAM 파일의 크기는 252GB 이며 BAM 파일의 크기는 46GB 이다. 각 단계에서 40 개의 작업 프로세스가 병렬로 데이터 처리 작업을 처리한다. 멀티스레딩이 가능한 BWA 와 samtools 는 각 프로세스마다 4 개의 스레드를 사용하여 모든 노드의 코어를 최대한 사용하였다.

실험 결과를 요약하면 8 배의 자원을 투입하여 4 배 이상의 성능 향상을 이룰 수 있었다. 3 단계 모두 병렬화를 통해 의미 있는 성능 향상을 이루었고 데이터 재배치 과정에서 추가로 발생하는 시간은 병렬화를 통해 얻는 이득에 비해 충분히 작은 양이라고 판단된다. 또한 두 방식으로 생성된 결과물을 비교하였을 때 데이터 처리 순서의 변경으로 인해 발생하고 최종 분석에 영양을 미치지 않는 오차를 제외하고는 동일한 결과물을 생성하는 것을 확인하였다. 분석의 종류에 따라서 실험에서 사용한 데이터 보다 수 배 이상 큰 데이터를 실제로 사용하는 경우도 존재하기 때문에 병렬 자원을 투입하여 소요시간을 단축시키는 것의 이점이 더 부각 될 수 있을 것이다.

<표 1> 208GB 원시 데이터의 총 전처리 소요 시간 비교

	기존 방법 1 Node, 20 cores		**병렬화 기법 적용** 8 Nodes * 20 cores	
지역정렬	bwa mem (20 thread)	5854s	KBigBWA on Lustre (40 Workers * bwa mem with 4 threads)	1508s
중복검출	samblaster	1468s	Data shuffle over IPoIB socket + 40 * samblaster parallel execution + Data filter	345s
색인 (인코딩, 압축포함)	samtools sort	4435s	40 * samtools sort parallel execution	526s
			Parallel file merge with MPIIO	24s
	samtools index	615s	samtools index	615s
총 시간	**12372 초**		**3018 초**	

4. 결론

본 연구를 통해 병렬 자원을 활용하여 유전체 시퀀스 데이터의 전처리 과정의 병렬화 기법을 제안하여 소요 시간을 크게 감소시킬 수 있음을 보였다. 이 기법을 이용하여 시급성이 요구되는 상황에서 고성능의 병렬 자원을 이용하여 유전 변이 검출 프로세스의 효율성을 크게 증가시킬 수 있을 것으로 기대된다. 향후 유전체 분석 파이프라인의 다른 단계의 병렬화 기법을 추가로 개발하고 전체 파이프라인을 효율적으로 실행하기 위한 사용자 인터페이스의 개발이 필요하다.

이 논문은 2018 년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No. R0190-18-2012, 빅데이터 처리 고도화 핵심기술개발 사업 총괄 및 고성능 컴퓨팅 기술을 활용한 성능 가속화 기술 개발)과 한국과학기술정보연구원 주요사업의 지원을 받아 수행된 연구임

참고문헌

[1] S. Goodwin, J. D. McPherson, and W. R. McCombie, “Coming of age: ten years of next-generation sequencing technologies.”, Nature Review Genetics, 17(6):333–351, May 2016.

[2] “Sequence Alignment/Map Format Specification” , The SAM/BAM Format Specification Working Group, https://samtools.github.io/hts-specs/SAMv1.pdf

[3] H. Li and R. Durbin, “Fast and accurate long-read alignment with Burrows-Wheeler transform.”, Bioinformatics. 26(5):589–595, 2010.

[4] An introduction to Next-Generation Sequencing Technology, Illumina, Inc., https://www.illumina.com/documents/products/illumina_sequencing_introduction.pdf

[5] Faust, Gregory G., and Ira M. Hall. “SAMBLASTER: Fast Duplicate Marking and Structural Variant Read Extraction.” Bioinformatics 30(17): 2503–2505, 2014.

[6] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R. and 1000 Genome Project Data Processing Subgroup, "The Sequence Alignment/Map format and SAMtools". Bioinformatics. 25 (16): 2078–2079, 2009.

[7] Decap, Dries, Joke Reumers, Charlotte Herzeel, et al. “Halvade: Scalable Sequence Analysis with MapReduce.” Bioinformatics. 31(15): 2482–2488, 2015.

[8] Byun, Eun-Kyu et al. “Accelerating Genome Sequence Alignment on Hadoop on Lustre Environment.” 2017 IEEE 13th International Conference on e-Science (2017): 436-437, 2017.

[9] J. M. Abuín, J. C. Pichel, T. F. Pena and J Amigo, “BigBWA: approaching the Burrows-Wheeler aligner to Big Data technologies”, Bioinformatics 31(24):4003-4005, 2015.