

# FEM 및 FVM solver 결합 이용에 대한 파일 I/O 의 영향

고원진\*, 이기택\*\*, 유지형\*, 정형수\*\*,  
 \*한양대학교 미래자동차공학과(미래자동차-sw 융합전공)  
 \*\*한양대학교 컴퓨터소프트웨어학과

ifyoung0@hanyang.ac.kr, ktlee20@hanyang.ac.kr, jihyungyoo@hanyang.ac.kr, hyungsoo.jung@hanyang.ac.kr

## Effect of File I/O on a coupled FEM & FVM solver

Won-Jin Ko\*, Ki-Taek Lee\*\*, Ji-Hyung Yoo\*, Hyung-Soo Jung\*\*

\*Dept. of Automotive Engineering (Automotive-computer convergence), Han-Yang University

\*\* Dept. of Computer Science, Han-Yang University

### 요 약

전기 자동차 에너지 시스템을 개발하기 위해서는 열 유체-전자기장 연성해석이 필요하다. 이를 위해 열 유체와 전자기장 각각의 해석에 특화된 패키지들을 사용하고, 이 패키지의 실험 결과를 저장 장치를 통해 주고받는다. 하지만 이는 저장장치 I/O 횟수를 늘려서 전체 계산 과정을 느리게 만들 수 있다. 데이터 교환을 MPI 패키지를 이용하여 DRAM 을 통해 이루어지도록 만들 수 있지만, 이는 계산 결과의 지속성을 악화시킨다. 즉, 열 유체 및 전자기장 연산과정에서 속도와 결과의 지속성은 상충적인 관계 갖는다. 본 연구는 이러한 관계를 실험적으로 분석하고 데이터의 지속성을 완화했을 때 이에 의해 얻을 수 있는 성능 이익을 분석한다.

### 1. 서론

전기 자동차 에너지 시스템을 개발하는 데 있어서 장치의 동작과 이에 의해 발생하는 발열 그리고 장치의 동작과 발열의 상호 영향 모두가 고려되어야 한다. 이를 위해, 열 유체-전자기장 연성해석을 수행하고, 이때, finite volume method(FVM)과 finite element method(FEM) 기법을 이용한다. 하지만, FVM 은 열 유체 해석에 특화되어 있고 FEM 은 전자기장 해석에 특화되어 있어서, 정확한 결과를 얻기 위해 서로 호환되지 않는 두 기법을 동시에 풀어야한다. 이를 위해 FVM 과 FEM 기법을 위한 solver 패키지들은 상대 패키지의 출력을 입력으로 이용한다. 이 때, 패키지 간 계산 결과를 문서 파일 형태로 주고받을 수 있으나 이는 I/O 횟수를 늘려서 계산 과정을 매우 느리게 만든다. 중간결과를 MPI로 전달하는 라이브러리를 이용하여 I/O 횟수를 줄이고 계산 과정에 걸리는 시간을 줄일 수 있지만, 이는 계산 결과의 지속성을 악화시킨다. 즉, 연산 속도와 연산 결과의 지속성은 I/O 빈도에 대해 상충적인 관계를 갖는다. 본 연구는 이러한 상충 관계를 실험적으로 분석하고 데이터의 지속성을 완화했을 때 이에 의해 얻을 수 있는 성능 이익을 분석한다.

### 2. 본론

#### 2.1 연산 시간과 연산 결과 지속성의 상충관계

전기 자동차의 에너지 시스템은 많은 발열을 발생시킬 수 있으며, 이렇게 나타난 발열은 시스템 내 다른 요소들에 영향을 끼칠 수 있다. 그리고 이렇게 발열에 의해 영향 받은 다른 요소들은 다시 발열의 또 다른 원인이 되기도 한다. 이렇게 다양하고 복잡한 현상의 전기 자동차 에너지 시스템을 문제없이 개발하기 위해서는 열 유체와 전자기장을 연동하여 해석하는 연성해석을 필요로 한다. 이 두 현상들을 해석하기 위해서는 열 유체와 전자기장 현상 모두 연립 편미분방정식으로 이뤄진 지배방정식을 풀어야 하며, 이 때 사용해야하는 지배 방정식을 풀기 위한 기법은 각각 FVM 과 FEM 로 이 두 기법을 연동하여 현상의 해석을 진행할 수 있다. 두 기법 모두 열 유체-전자기장 연성해석이 가능하지만 FVM 은 열 유체 해석에 FEM 은 전자기장 해석에 특화되어 있어서 정확한 결과를 얻기 위해서는 서로 호환되지 않는 두 기법을 동시에 풀어야 한다.

일반적으로 FVM 과 FEM 기법 모두 특화된 solver 패키지들을 활용하여, 연립 편미분방정식을 푸는 동작을 수행한다. 그리고 이 때 더 정확한 결과 값을 얻기 위해 계산 결과를 서로 교환하여 결과가 수렴이

될 때까지 계산을 반복한다. 즉, 이 두 solver 패키지는 각각의 계산이 수렴할 때까지 상대 패키지의 계산 결과를 입력으로 받아들여서 계산하는 과정을 반복하여 더 실제 값과 가까운 근사치를 얻는 작업을 수행한다. 그리고 이 과정에서 패키지 간 결과물을 텍스트 파일로 주고받는다.

하지만 이러한 텍스트 파일을 이용한 데이터 전달 과정은 저장장치에 I/O 를 유발할 수 있고, 이 때 데이터 전달을 위해 생성되는 텍스트 파일의 용량은 수십 메가바이트에서 수 기가바이트이다. 그리고 이러한 작업은 근사값을 얻기 위해 계산을 반복하는 과정에서 매번 계산을 끝내고 다시 시작할 때, I/O 를 발생시킨다. 즉, 전체 계산 과정에서 I/O 가 소모하는 시간의 비중을 늘릴 수 있다.

이를 해결하기 위해 중간결과를 MPI 로 각각의 패키지 간 전달하는 라이브러리인 EOF-library 를 이용할 수 있다[3]. 즉, 지연 시간이 긴 저장장치를 이용한 결과 교환 방식보다, DRAM 을 이용하여 결과 값을 교환하는 방식이 더 이점을 가질 수 있다. 하지만 DRAM 을 통해 데이터를 교환하는 것은 연산의 결과를 저장장치에 기록하지 않는다는 것을 의미하며, 연산이 에러에 의해 중단되는 경우 DRAM 을 통해 교환된 데이터는 복구가 불가능해진다. 즉, 계산 결과 값들의 영속성이 깨질 수 있다. 이는 연산 속도와 결과의 영속성이 상충하는 관계를 갖는다는 것을 의미한다.

본 연구는 이 두 상충 관계를 분석하기 위해 결과 값 교환 과정에서 저장장치를 통한 교환의 빈도를 조절하고 빈도에 따른 연산 속도를 측정한다. 그리고 이 측정 값을 통해 데이터 결과의 영속성을 완화할 시, 이에 따라 어느 정도의 계산 성능의 이득을 얻을 수 있는지 확인한다.

## 2.2 실험결과 및 분석

본 장에서는 FVM 을 풀기 위한 패키지로 OpenFOAM 을 이용하고, FEM 을 풀기 위한 패키지로 Elmer 를 이용한다. 이 두 패키지를 연동하여 실험 수행 후, 이 때의 실험 결과를 분석한다[1], [2]. 실험은 같은 횟수의 연산을 반복하면서 이 때 수행되는 I/O 횟수를 조절하는 방식으로 진행이 되었다. I/O 횟수가 변화함에 따라 변경되는 연산 시간을 측정하였고, 이를 통해 I/O 횟수의 변화가 연산 속도에 미치는 영향을 확인하였다. 또한 동일한 계산을 계산 노드 로컬 디스크 상의 디렉토리와 NFS 를 이용하여 연결된 디렉토리에 수행하여 저장장치 I/O 만 발생했을 때의 연산의 속도를 확인하였고 네트워크를 통한 데이터 전

송이 연산 속도에 미치는 영향을 분석하였다.

### 2.2.1 실험환경

본 실험은 아래 표 1 에서의 실험 환경에서 수행이 되었다. 실험은 infiniband 로 계산 노드와 계산의 결과 값이 저장되는 노드들을 연결한 환경에서 수행이 되었다. 즉 데이터 계산 노드가 데이터 저장 노드로부터 데이터를 읽어 온 후에, 이 데이터를 계산하고 다시 계산 결과 값을 데이터 저장 노드에 적어 놓는 방식으로 실험이 수행되었다.

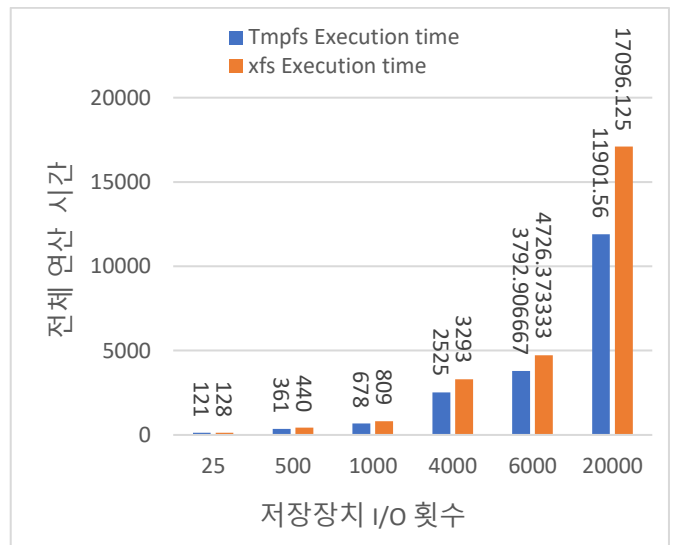
계산 노드는 데이터 교환을 위해 데이터 저장 노드를 NFS 서버로 사용하였다. 즉 데이터가 저장되는 노드에서 디렉토리에 NFS 를 마운트 하여 공유 디렉토리를 클러스터 내에 공개하고 이 디렉토리에 계산 노드가 결과 값을 작성하고 데이터를 읽어오는 방식으로 실험이 수행되었다.

또한 데이터 저장 노드는 저장장치를 효율적으로 이용하기 위해, RAID 0 레벨로 여러 저장 장치들을 묶어서 병렬로 이용하였다.

<표 1> 실험환경

CPU	2 x Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz
Local Machine RAM	96GB
Remote Machine RAM	136GB
RAID controller	Dell PERC H310
RAID level	0
Network card	Mellanox Technologies MT27500 Family [ConnectX-3]

### 2.2.2 저장장치 I/O 횟수 변화 실험



(그림 1) 저장장치 I/O 횟수에 따른 연산 시간의 변화

그림 1 은 저장장치 I/O 횟수에 따른 연산 시간의 변화에 대한 그래프이다. 실험 측정을 위해, 원격 저장장치에 데이터가 저장되는 횟수를 조절하면서 전체 반복 계산의 연산 시간을 측정하였다. 실험을 통해 원격 저장장치에 I/O 를 발생시키지 않고 같은 노드 내에서 MPI 라이브러리를 통해 데이터를 교환할 경우 데이터의 지속성을 완화하여 연산 속도를 더 빠르게 만들 수 있다는 것을 확인했다. [3]

또한, 이 때 네트워크가 데이터 쓰기 속도에 미치는 영향을 파악해보기 위해, tmpfs 에 존재하는 원격 디렉토리와 SSD 에 설치된 xfs 에 존재하는 원격 디렉토리에서 실험을 수행하였다. 그림 1 에서 확인할 수 있듯이, tmpfs 로의 읽기와 쓰기는 메모리에서 발생하는 것임에도 불구하고, MPI 를 사용해서 데이터를 교환하는 것보다 많은 시간을 소모하는 것을 확인할 수 있다. 이는 데이터 저장 노드의 SSD 에서 발생하는 I/O 외에도 네트워크를 통한 데이터 전송에서 발생하는 지연이 상당히 크다는 것을 보여준다.

### 3. 결론

저장장치에서 발생하는 I/O 횟수에 따른 열 유체-전자기장 연성해석의 계산 시간 비교 분석을 진행하였고, 이를 통해 데이터의 지속성과 연산 속도의 상충적인 관계를 확인할 수 있었다. 위 전산실험을 통해 얻은 자료는 향후 다양한 전산해석에 관한 실험 및 연구를 진행할 때 기초자료로 사용될 수 있을 것으로 기대한다.

#### 참고문헌

- [1] OpenCFD Group, OpenFOAM, Git repository <https://develop.openfoam.com/Development/openfoam>
- [2] Elmer FEM, Elmer, GitHub repository <https://github.com/elmercsc/elmerfem>
- [3] EOF Consulting LLC, EOF-library, Github repository <https://github.com/jvencels/EOF-Library>