[Research Paper]

구획공간 화재시뮬레이션에 대한 FDS 모델의 병렬계산 효율: 공유메모리 계산환경

민수경 · 김성찬^{*†}

경일대학교 소방방재학과 대학원생, *경일대학교 소방방재학과 교수

Efficiency in Parallel Computing of FDS Model for Compartment Fire Simulation: Shared Memory System

Su-Gyeong Min · Sung-Chan Kim^{*†}

Graduate Student, Dept. of Fire Safety, Kyungil Univ.,

*Professor, Department of Fire Safety, Kyungil Univ.

(Received February 24, 2021; Revised April 14, 2021; Accepted April 14, 2021)

요 약

본 연구는 효과적인 화재시뮬레이션 결과 도출을 위해, FDS 모델의 병렬계산모드와 계산영역 분할방식에 따른 계산효율을 비교하였다. 해석영역의 크기는 $12.0\text{ m} \times 3.8\text{ m} \times 3.0\text{ m}$ 이며 화원은 $0.4\text{ m} \times 0.4\text{ m}$ 의 n-Heptane 화재를 고려하였다. 해석영역은 0.1 m 격자크기를 적용하여 136,000개의 격자로 구성되며 각 계산의 효율은 다중 코어 단일 CPU와 공유메모리 환경에서 시뮬레이션 종료시간인 300 s까지의 wall clock time을 통해 평가된다. MPI 모드와 하이브리드 모드를 통한 병렬계산은 OpenMP 모드에 비해 매우 높은 속도향상지수(speed-up)를 나타냈으며 계산효율은 해석분할 방식에 따라 큰 차이를 보였다. 단일 해석영역에 대한 OpenMP 모드의 최대 속도향상지수는 1.5를 넘지 않았으며 스레드의 할당이나 해석영역의 분할에 있어서 최적조건이 존재함을 보였다. 본 연구는 제한된 계산자원과 시간을 활용하여 보다 효과적인 화재해석결과를 도출하는데 기여하고자 한다.

ABSTRACT

This study evaluates the computational efficiency based on the parallel processing mode and domain decomposition method of the FDS model to enhance the computational performance of fire simulation. A single compartment of dimensions $12.0\text{ m} \times 3.8\text{ m} \times 3.0\text{ m}$ is considered along with a rectangular fire source ($0.4\text{ m} \times 0.4\text{ m}$) fueled by n-Heptane. The computational domain was divided into 136,000 cells forming a grid size of 0.1 m , and the computational efficiency for each calculation was evaluated by the wall clock time for a simulation time of 300 s using a computational framework with 24 cores of a single CPU and a 256 GB shared memory system. The MPI and hybrid mode in FDS parallel offers a greater speed-up capability than the OpenMP mode, and the domain decomposition method used greatly affects the computational efficiency. The maximum speed-up with the OpenMP mode was less than 1.5 for a single computational domain, which indicates that there is an optimal condition for thread assignment and domain decomposition in the OpenMP mode. The present study is expected to contribute toward obtaining effective fire simulation results with limited computing power and time in fire protection engineering.

Keywords : Fire simulation, OpenMP, MPI, FDS, Fire model, Parallel computation

1. 서 론

Fire dynamics simulator (FDS)는 대표적인 필드 모델(Field model)로써 화재발생으로 인한 열과 물질전달의 비정상 유동(Unsteady flow)을 수치해석적으로 계산하는 화재전

용 해석프로그램이다. FDS모델은 격자를 기반으로 해석이 이루어지기 때문에 계산격자크기는 해석결과의 해상도와 신뢰성에 직접적인 영향을 미치는 중요한 인자가 될 수 있다⁽¹⁻³⁾. 따라서 해석영역의 크기와 계산격자 크기에 의해 계산용량은 결정되며, 계산용량이 증가함에 따라

계산에 필요한 시간과 자원(Computing power)이 늘어나기 때문에 사용자는 해석결과의 타당성과 계산비용을 고려하여 효율적인 격자크기와 계산방식을 선택해야 한다.

계산방식에 있어서 FDS 모델은 순차계산(Sequential computation)과 병렬계산(Parallel computation)을 모두 지원하고 있으며 특히, 병렬계산의 경우 OpenMP와 Message passing interface (MPI), OpenMP와 MPI를 결합한 하이브리드(Hybrid) 모드를 적용할 수 있다⁽⁴⁾.

Haarhoff 등은 공유 메모리 계산환경에서 단일 구획공간에 대해 OpenMP 모드와 MPI, 하이브리드 모드를 적용하여 계산서브루틴별 병렬화도와 속도향상지수(Speed-up)를 비교하였다. 해석결과 컴퓨팅 환경이나 계산격자별로 설정 스레드 수의 최적값이 존재함을 보였으며, MPI 모드의 계산 효율이 OpenMP에 비해 상당히 높은 것으로 평가되었다⁽⁵⁾.

Weisenpacher 등은 터널화재의 13~43백만개의 격자에 대해 최대 48개의 CPU 코어를 적용하여 OpenMP, MPI, 하이브리드 모드의 다양한 병렬계산을 수행하였다. MPI 모드의 경우 계산환경에 따라 최대 11배까지 차이를 보였으나 OpenMP 모드의 경우 MPI 모드에 비해 상대적으로 매우 낮은 속도향상지수를 보였다.

FDS 모델에서 적용되는 병렬계산모드는 CPU 혹은 스레드(thread)간의 데이터 전달방식, 해석영역 분할방법, 프로세스별 계산 부하 등의 다양한 설정환경에 따라 계산효율의 차이가 발생할 수 있다⁽⁶⁾.

화재시뮬레이션이 국내 소방엔지니어링 분야에 널리 적용되고 있는 성능위주소방설계 등의 대상 구조물은 내부 구조가 복잡하고 길이 스케일이 상대적으로 크기 때문에 충분한 해상도를 가진 계산 결과를 얻기 위해서는 상대적으로 많은 계산격자를 필요로 하며 고성능 계산환경을 요구한다. 병렬계산은 하드웨어적인 여건과 함께 전체 해석영역을 분할하여 계산 하중을 분산시킴으로써 계산 효율을 증대시킬 수 있다. 이를 위해 주어진 환경에서 FDS 모델의 병렬계산모드에 따른 계산효율을 분석하고 최적의 계산조건을 도출하는 것은 많은 계산격자에 대해 빠른 계산을 도모하는 필수적인 과정이다.

본 연구에서는 단일 구획공간에 대해 동일한 해석조건을 적용하여 FDS 모델의 병렬계산모드와 해석영역 분할방식에 따른 계산효율을 비교하고자 한다. 특히 화재공학 실무자들이 유용하게 활용할 수 있도록 다중 코어를 가진 단일 컴퓨터 환경에서 계산효율을 분석함으로써 제한된 계산 자원과 시간을 활용하여 보다 효과적인 화재해석 결과를 도출하는데 기여하고자 한다.

2. 본 론

2.1 FDS 모델의 병렬계산

일반적으로 CFD모델의 병렬계산은 전체 해석영역에 대해 여러 개로 분할된 개별 해석영역별로 프로세서를 적용

Figure 1. Memory system of parallel programming model.

하여 계산을 수행하는 방식이나 하나의 해석영역을 프로그램 내부적으로 영역을 분할하여 동시에 여러 프로세서에서 계산을 수행하는 것을 포함한다. 병렬계산의 컴퓨팅 환경은 크게 연산을 수행하는 프로세서(코어)와 연산에 필요한 데이터를 기억하는 메모리로 구성되며 프로세서별로 메모리를 가지는 분산 메모리 시스템(Distributed memory system)과 전체 메모리를 개별 프로세서들이 공유하는 공유 메모리 시스템(Shared memory system), 물리적으로 분산된 메모리를 논리적으로 공유하여 사용하는 분산 공유 메모리 시스템(Distributed shared memory system) 등으로 구분할 수 있다⁽⁷⁾.

FDS 병렬계산에서 OpenMP 모드는 Figure 1(a)와 같이 공유 메모리 시스템을 기반으로 계산에 적용되는 코어들이 하나의 공유 메모리를 통해 데이터를 기록, 전달하는 방식이다. 주로 다중코어를 가지는 단일 컴퓨터에 적용되며, 사용자가 계산영역을 분할하지 않아도 OpenMP 라이브러리를 통해 지정된 스레드(Thread)수만큼 해석영역을 분할하여 손쉽게 병렬계산을 수행할 수 있다⁽⁴⁾.

MPI는 Figure 1(b)와 같이 분산 메모리 시스템을 기반으로 계산에 적용되는 코어들이 물리적으로 분산된 메모리를 통해 프로세서별로 계산을 수행하고 네트워크를 이용하여 데이터를 전달하는 방식이다. 여러 대의 컴퓨터를 연결한 클러스터나 슈퍼컴퓨터에 적용되며, 사용자가 분할, 할당한 계산영역에 대해 프로세서가 단일 스레드를 이용하여 병렬계산을 수행한다.

하이브리드 모드는 분산 공유 메모리 시스템을 기반으로, 사용자는 MPI를 이용하여 계산영역을 분할하고 OpenMP를 통해 분할·할당된 계산영역에 대해 라이브러리를 적용하여 지정한 스레드 수만큼 다중 코어 계산을 수행하는 방식이다. 다중코어를 가지는 다수의 컴퓨터를 네트워크를 통해

Table 1. Computer Specification Used in This Study

CPU model	AMD Ryzen Threadripper 3960X
No. of Core	24 cores
No. of Threads	48 threads
CPU clock speed	3.8 GB
RAM	258 GB
Operating system	Windows 10 education

Figure 2. Computational domain and dimensions of the fire compartment.

연결한 클러스터 환경에 주로 적용된다.

본 연구는 Table 1과 같이 다중코어(24코어) 단일 CPU와 256 GB의 공유 메모리 시스템을 적용하여 3가지 병렬계산 모드를 적용하였다. FDS 해석에서 스레드당 할당되는 메모리(OMP_STACKSIZE)는 해석대상에 따라 계산속도에 영향을 미칠 수 있으나 본 해석에서는 16 MB로 고정하고 계산을 수행하였으며 할당 메모리에 따른 영향은 고려하지 않고 병렬계산모드와 영역분할방식에 따른 영향을 중심으로 계산효율을 평가하고자 한다.

최적의 계산조건 도출을 위해 계산시작에서 정상적인 종료까지의 실제 프로그램의 물리적 실행시간을 의미하는 Wall clock time (WCT)을 이용하여 속도향상지수(Speed-up, S)를 계산하고 계산의 효율성을 평가하였다⁽⁸⁾. 속도향상지수는 단일 프로세서 대비 다중 프로세서의 계산속도비를 의미하며, 계산속도가 빨라질수록 속도향상지수는 증가하게 된다.

$$S(N) = \Delta t_1 / \Delta t_N$$

여기서 N 은 프로세서 수, Δt_1 은 하나의 프로세서를 통해 계산에 소요된 시간, Δt_N 은 N 개의 프로세서를 통해 소요된 시간이다.

2.2 화재시뮬레이션 대상 및 조건

본 연구에서는 병렬계산모드 및 해석영역 분할방식에

Figure 3. Comparison of heat release rate between experimental data and FDS MPI calculation with 8 domains.

따른 FDS 모델의 계산 효율성을 평가하기 위해 Figure 2와 같이 직육면체의 단일 구획공간을 해석영역으로 선정하였다.

해석공간의 크기는 길이 12 m, 폭 3.8 m, 높이 3 m이며 바닥면을 제외한 구획공간 외부는 해석영역에 포함되지 않는다.

계산에 사용된 FDS 버전은 6.7.4로서, 격자크기는 한 변의 길이가 0.1 m인 정방형 격자로 구성하였으며, 전체 격자수는 136,000개이다. 화원은 구획실 안쪽 벽면에서 2.1 m 떨어진 정중앙에 위치하며, 크기는 0.4 m × 0.4 m × 0.1 m로 연료는 n-heptane을 가정하였다. 화재 시나리오는 실험에서 측정된 발열량을 주요 시간대에서 추출하여 입력하였으며, 시뮬레이션 시간은 300 s로 모두 동일하다. Figure 3은 측정값을 기반으로 입력된 발열량과 8개의 계산영역을 적용한 FDS MPI 모드의 계산 발열량을 나타내며, 모든 시뮬레이션 조건에서 측정값과 유사한 계산 발열량을 보였다.

Table 2는 병렬계산모드별로 계산에 적용된 해석영역 분할방식과 그에 따른 전체계산영역 수를 나타낸다. OpenMP 모드는 x축 방향의 수평분할된 계산영역 수와 계산에 할당된 스레드 수(OMP_NUM_THREADS)에 대해 효율성을 비교하였다⁽⁹⁾. MPI 모드와 하이브리드 모드는 x축 방향의 수평분할, z축 방향의 수직분할, x축 방향과 z축 방향을 모두 고려한 복합분할조건에 대해 계산속도를 비교하였다.

3. 결 과

3.1 병렬계산 반복성

계산 효율의 비교에 앞서 해석결과의 반복성(Repeatability)을 확인하기 위해 동일한 계산모델 및 해석조건에 대한 반복계산을 수행하였다. 이때 동일한 계산환경을 제공하기 위해 개별 단독 계산을 수행하였으며, 운영체제에서 제공하

Table 2. Summary of Case for Decomposition of Computational Domain

Parallel mode	Case	Number of division		Total No. of domain
		x axis	z axis	
OpenMP	1	1	1	1
	2	2	1	2
	3	4	1	4
	4	8	1	8
	5	12	1	12
MPI / hybrid	1	1	1	1
	2	2	1	2
	3	4	1	4
	4	8	1	8
	5	12	1	12
	6	15	1	15
	7	20	1	20
	8	24	1	24
	9	1	2	2
	10	1	4	4
	11	1	8	8
	12	1	10	10
	13	1	2	2
	14	2	2	4
	15	4	2	8
	16	6	2	12
	17	8	2	16
	18	10	2	20
	19	12	2	24

는 기본적인 프로세스를 제외한 별도의 추가적인 프로그램 실행 없이 FDS 프로그램만 실행하여 계산시간을 평가하였다.

Figure 4는 해석영역을 x축 방향으로 8분할한 계산조건에서의 OpenMP 모드와 하이브리드 모드의 2회 반복계산 결과를 나타낸다. OpenMP 모드의 경우 4개의 스레드를 이용하여 계산을 수행하였으며, 하이브리드 모드는 분할영역당 2개의 스레드를 할당하여 계산하였다. 격자구성 및 초기해석조건에 대한 설정이 진행되는 초기계산단계를 제외하고 OpenMP 모드의 경우 반복계산에 따른 계산시간 차이는 최대 0.8%를 넘지 않았다. 하이브리드 모드의 경우 2.1% 정도로 OpenMP 모드에 비해 상대적으로 높은 계산시간의 차이를 보였으나, 전체 계산시간을 고려할 때 두 병렬계산 모드의 반복성은 매우 높은 것으로 평가할 수 있다.

3.2 OpenMP 모드의 계산 효율

OpenMP 모드의 계산효율을 파악하고자 해석영역을 분할하지 않은 단일 계산영역과 해석영역을 분할한 다중 계

Figure 4. Comparison of wall clock time for verifying repeatability of OpenMP and hybrid mode of FDS model.

산영역에 대해 시뮬레이션을 수행하였다. Figure 5(a)는 단일 계산영역에서 스레드 수에 따른 계산소요시간(WCT) 변화를 나타낸 것으로, 시뮬레이션 시간별로 소요되는 계산시간은 계산 초기를 제외하고 높은 선형성을 유지하였다. 계산소요시간은 할당된 스레드가 가장 적은 1개에서 9,740 s로 최댓값을 보였으며, 8개의 스레드가 할당된 경우 약 30% 줄어든 6,869 s로 최솟값을 보였다. 그러나 32와 48개의 스레드를 적용한 경우, 계산소요시간은 오히려 증가하여 각각 8,205 s, 8,768 s를 나타냈다. Figure 5(b)는 OpenMP 모드에서 단일 계산영역에 적용된 스레드 수에 따른 속도향상지수를 비교한 것이다. 8 이하의 스레드 적용 시 스레드 수에 따른 속도향상지수는 급격히 증가하여 1.42의 최댓값을 보였으나 8 이상의 스레드 수에서 속도향상지수는 점진적으로 감소하여 컴퓨터의 최대 스레드인 48개를 할당한 경우 속도향상지수는 1.11까지 감소하였다.

Figure 6은 OpenMP 모드에서 해석영역 분할에 따른 계산영역 수와 할당 스레드 수(NUM_OF_THREADS)에 따른 계산소요시간과 속도향상지수를 나타낸다. 고려한 스레드 수는 OpenMP 모드의 단일 계산영역 결과에서 최대 계산효율을 보인 8까지만 고려하였다.

할당 스레드 수에 상관없이 해석영역 수에 따른 계산소요시간은 전체적으로 유사한 경향을 보였으며 해석영역 수가 2인 경우 최소 계산소요시간을 나타냈고 이후 해석영역 수가 증가함에 따라 계산소요시간은 증가하였다. 이는 OpenMP 모드의 경우 계산영역을 분할할 시 영역분할에 따른 계산효율의 증가 이외에도 해석영역 간 정보전달과정에 계산효율이 감소하는 등 복합적인 요인이 작용하기 때문으로 생각된다. 계산에 할당된 스레드의 수에 따른 계산소요시간은 해석영역 수에 따라 일부 차이는 있으나 전체적으로 스레드 수가 4이상인 경우 큰 차이를 보이지 않았다. 따라서 사용자가 계산영역을 분할하지 않고 OpenMP

(a) Wall clock time

(b) Speed-up

Figure 5. Comparison of the wall clock time and speed-up according to the number of threads used in OpenMP calculation for a single domain.

모드를 통해 병렬계산을 수행할 경우, 할당 스레드 수의 증가에 따라 컴퓨팅 자원의 사용률은 증가하나 계산효율은 크게 향상되지 않는 것으로 나타났다. 또한 계산영역 분할에 따른 계산시간 단축의 효과가 미미하며, 본 연구의 경우 해석영역을 4 이상 분할한 경우 계산소요시간이 오히려 증가하는 것으로 나타났다.

3.3 MPI 모드의 계산 효율

MPI 병렬계산모드는 사용자가 직접 해석영역을 분할하고 분할된 해당영역을 개별 프로세서에 할당하기 때문에 분할 방식에 따른 영향을 함께 비교하였다. Figure 7에서 보는 바와 같이 MPI 모드를 통한 병렬계산 수행 시, 수평이나 수직방향의 단일방향 분할보다 혼합방향 분할이 효율적이며 본 해석 대상의 경우 계산영역은 8개 이하로 분할하는 것이 가장 효율적으로 나타났다.

(a) Wall clock time

(b) Speed-up

Figure 6. Comparison of wall clock time and speed-up according to the number of domain and threads for OpenMP calculation.

3.4 하이브리드 모드의 계산 효율

하이브리드 병렬계산모드는 사용자가 분할, 할당한 계산영역에 대해 다중 스레드를 적용하여 계산을 수행한다. 본 연구에서는 동일한 계산영역에 대해 2개의 스레드를 적용하여 계산을 수행하였다.

Figure 8은 하이브리드 모드 적용 시 분할방식별 계산영역 수에 따른 계산소요시간을 비교하여 나타낸다. 해석영역의 분할방식이나 병렬계산모드에 관계없이 단일 영역에서 영역분할이 증가함에 따라 계산소요시간은 급격히 감소하였으나 해석영역 수 8 근처에서 계산소요시간은 더 이상 감소하지 않았으며 이후 해석영역증가에 따라 오히려 계산소요시간은 점진적으로 증가하는 경향을 보였다. 이는 영역분할에 따른 계산속도 향상에 비해 분할된 영역 간 정보전달이나 영역 간 계산시간차이에 따른 대기시간에 기인한 것으로 판단된다.

(a) wall clock time

(b) speed-up

Figure 7. Comparison of wall clock time and speed-up according to the number of domain along the direction of domain decomposition for MPI calculation.

Figure 8. Comparison of wall clock time according to the number of domain along the direction of domain decomposition.

Figure 9. Comparison of wall clock time of MPI and hybrid mode for 8 sub-domains according to the direction of domain decomposition.

본 연구에서 MPI와 하이브리드 모드의 계산효율이 가장 높은 영역분할 수 8인 경우에서 x축, z축, x + z축 방향의 분할방식에 따른 계산소요시간을 Figure 9에 비교하였다. 전체적으로 동일 해석조건에서 하이브리드 모드와 MPI 모드의 계산소요시간은 유사한 결과를 나타냈고, MPI 모드를 적용한 경우가 분할방식에 관계없이 계산소요시간이 감소하나 계산방식에 따른 차이는 크지 않는 것으로 나타났다.

4. 결 론

병렬처리기술을 이용한 효과적인 화재시물레이션 해석 결과 도출을 위해, 본 연구에서는 다중 스레드를 갖는 단일 CPU와 공유메모리 계산환경에서 단일 구획공간에 대한 FDS 모델의 병렬계산모드와 계산조건에 따른 계산소요시간, 속도향상지수 등을 비교하여 얻은 결론은 다음과 같다.

- 본 연구의 해석조건과 계산환경하에서 OpenMP 모드를 통한 계산 수행 결과, 단일 스레드 대비 계산에 적용된 다중 스레드 수에 따라 최대 속도향상지수는 단일 계산영역의 경우 1.42, 다중 계산영역은 1.07로 크지 않았으며, 8 이상의 스레드 할당 시 오히려 계산효율은 점진적인 감소를 보였다. 따라서 OpenMP 모드의 경우 계산영역 분할 없이 스레드 수의 할당을 통해 병렬계산을 수행하는 것이 효율적이며 격자크기, 해석영역크기 등의 계산조건에 따라 최적 스레드 수를 도출하는 과정이 필요할 것으로 판단된다.

- MPI 모드를 통한 병렬계산 수행 결과 OpenMP 모드에 비해 계산효율은 매우 높게 나타났으며 해석영역의 분할 방식에 따라 계산 효율은 큰 차이를 보였다. 특히 단일 CPU와 공유메모리 환경에서는 전체해석영역의 크기 및 격자수 등의 계산조건에 따른 해석영역의 최적분할 방식을 검토할 필요가 있다.

- 하이브리드 모드를 통한 병렬계산 수행 결과, MPI 모드와의 계산소요시간의 차이는 크지 않았으며 해석영역당 할당되는 스레드 수는 계산속도 향상에 큰 영향을 미치지

않는 것으로 나타났다.

본 연구는 제한된 해석영역과 화재시나리오 및 계산환경이긴 하나 단일 구획공간에 대해 화재해석 시, OpenMP 모드보다는 MPI 모드나 하이브리드 모드 적용이 효율적인 결과를 제공하였으며, 계산영역분할을 통한 병렬계산과정에도 계산영역 간 정보전달에 계산시간이 소요되기 때문에 최적 영역분할 수와 분할방식을 통해 계산 효율의 향상이 필요함을 보이고 있다. 병렬계산의 효율은 계산환경에 따라 차이가 있을 수 있기 때문에 본 연구 결과를 보편적으로 받아들이기 보다 개별 시스템에 적합한 최적 계산조건을 설정하는 것이 중요하다. 향후 다중 CPU 및 분산메모리 계산환경하에서 계산효율을 추가로 평가하여 제한된 계산자원과 시간을 활용하여 보다 효과적인 화재해석 결과를 도출하는데 기여하고자 한다.

후 기

본 연구는 소방청 소방대응력 향상을 위한 연구개발사업(2018-NFA002-005-0104000-2020)의 연구비 지원으로 수행되었습니다.

References

1. H. S. Yun and C. H. Hwang, "Analysis of Prediction Results and Grid Size Dependence According to Changes in Fire Area", Fire Science and Engineering, Vol. 33, No. 6, pp. 9-19 (2019).
2. W. C. Park, "Selection of Grid Size in Fire Simulation for Large Scale Buildings by Using FDS", Fire Science and Engineering, Vol. 26, No. 5, pp. 67-72 (2019).
3. S. C. Kim and S. H. Lee, "Assessment of Grid Sensitivity in the FDS Field Model to Simulate the Flame Propagation of an Electric Cable Fire", Journal of the Korean Society of Safety, Vol. 23, No. 4, pp. 30-35 (2008).
4. K. McGrattan, S. Hostikka, R. McDermott, J. Floyd, M. Vanella, "Fire Dynamics Simulator Technical Reference Guide, Volume 1: Mathematical Model", NIST Special Publication 1018-1, 6th Ed. (2019).
5. D. Haarhoff and L. Arnold, "Performance Analysis and Shared Memory Parallelisation of FDS", Proceedings of Fire and Evacuation Modeling Technical Conference, (2014).
6. P. Weisenpacher, L. Halada, J. Glasa, and V. Sipkova, "Parallel Model of FDS Used for a Tunnel Fire Simulation", Proceeding of the International Conference Parallel Numerics 11, pp. 96-105 (2011).
7. K. Kavi, H. S. Kim, B. Lee and A. R. Hurson, "Shared Memory and Distributed Shared Memory System: A Survey", Advances in Computers, Vol. 53, pp. 55-108 (2000).
8. P. S. Pacheco, "An Introduction to Parallel Programming", Elsevier Inc (2011).
9. OpenMP, "OpenMP Application Programming Interface", OpenMP Architecture Review Board, (2018).