

# X-Hyper320TKU 임베디드 디바이스 상에서의 CVM 성능비교 분석

박윤정<sup>o</sup> 민덕기

건국대학교 컴퓨터정보통신공학과

sm6280p@konkuk.ac.kr, dkmin@konkuk.ac.kr

## Comparative Performance Analysis of CVM on X-Hyper320TKU Embedded Device

Yunjung Park<sup>o</sup> Dugki Min

School of Computer Science Engineering, Konkuk University

### 1. 서론

최근 유비쿼터스 환경을 위한 다양한 임베디드 디바이스가 등장함에 따라 이를 지원하기 위한 다양한 애플리케이션과 이들의 성능에 많은 관심을 가지게 되었다. CVM은 스마트 디바이스, PDA, 셋탑박스과 같은 자원이 제한되었지만 32bit 이상의 프로세서를 가진 임베디드 디바이스에서 자바 애플리케이션 실행을 위한 가상 머신이다. 본 논문에서는 이러한 CVM 가상 머신이 임베디드 디바이스 상에서 어떠한 성능을 보여주는지를 기존의 JVM과의 비교를 통해 테스트하여 그 결과를 분석해 보고자 한다.

### 2. 실험 설계

CVM과 JVM의 성능 비교 테스트는 자바 애플리케이션 처리 능력을 통해 측정하도록 한다. 테스트 애플리케이션은 가장 많이 사용되는 기능인 I/O, Thread, Network의 성능을 측정할 수 있는 애플리케이션을 선택하였다. I/O 처리 성능 테스트는 다양한 용량의 텍스트 파일을 열고 이를 읽어 들이는 시간을 측정하므로 파일 크기에 따라 얼마나 효율적으로 파일을 처리 할 수 있는지를 측정하였다. Thread 처리 성능 테스트는 애플리케이션 내에서 여러 개의 Thread가 동시에 작업을 실행할 때에 이를 처리하는 데에 얼마나 시간이 걸리는지를 측정하는 테스트이다. Thread 처리 성능은 애플리케이션 내의 Thread 수와 각 Thread가 처리하는 작업횟수를 각각 다르게 하여, 각 요소에 따라 가상 머신에서 작업 처리 시간이 어떻게 달라지는지를 측정하였다. Network 처리 성능을 테스트는 TCP와 RMI 네트워크 통신을 통하여 외부 에코 서버에 메시지를 전달하고 받는데 걸리는 응답 시간을 측정하였다.

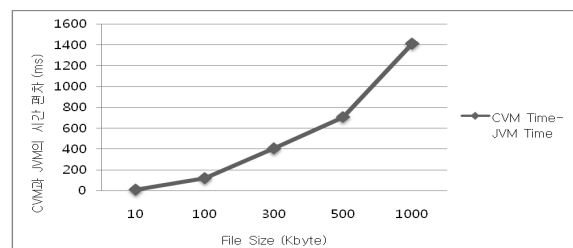
### 3. 실험 결과

첫 번째 I/O 처리 성능에 대한 테스트는 10번씩 수행되었으며, 파일 용량에 따른 CVM과 JVM의 파일 처리 시간의 평균 시간의 값은 [표 1]과 같다. [그림 1]은 각각 다른 파일 크기를 처리할 때 발생하는 CVM과 JVM의 처리시간 편차를 나타내고 있다.

표 1. 용량에 따른 CVM과 JVM의 파일 처리 시간

File Size	CVM	JVM
10 Kbyte	81.1 ms	73.3 ms
100 Kbyte	764.6 ms	644.9 ms
300 Kbyte	2339.8 ms	1936.1 ms
500 Kbyte	3910.4 ms	3204.6 ms
1000 Kbyte	7835.1 ms	6425.2 ms

그림 1. 파일 크기별 가상머신 간의 파일 처리시간 편차



[표 1]과 [그림 1]에 나타난 결과로 볼 때 CVM은 파일 처리 부분에서 JVM에 비해 낮은 성능을 보여주며, 특히 처리해야 할 파일 크기가 커질수록 CVM의 성능이 급격하게 저하됨을 알 수 있다. 이는 CVM은 처리량이 많을수록 File I/O 태스크에서 낮은 성능을 나타내고 있다는 것을 보여준다.

둘째로 Thread 처리 성능에 대한 테스트는 실행계획대로 각 조건마다 5번씩 이루어졌으며, 아래 [표 2,3]에는 이러한 테스트 결과에 대한 평균을 작업 수행 시간이다.

표 2. CVM에서의 Thread 수와 작업량 별 수행 시간 (단위: ms)

작업량	Thread			
	2	5	10	30
100	0.3	0.24	0.18	0.26
1000	2.2	2.24	2.26	2.75
10000	21	25.32	39.08	40.7
1000000	3577.9	6066.68	12253.32	38816.64

표 3. JVM에서의 Thread 수와 작업량 별 수행 시간 (단위: ms)

작업량	Thread			
	2	5	10	30
100	0.4	1.08	0.67	0.70
1000	5.2	5.8	4.48	3.58
10000	5.5	10.04	8.28	5.99
1000000	273.3	407.8	1358.66	2020.87

위의 결과로 볼 때에 CVM에서는 작업량이 적을수록 높은 성능을 보이고 있으며, 작업량이 많아질수록 급격하게 CVM의 성능이 저하되고 있음을 알 수 있다. JVM은 작업량이 적을 때에 CVM에 비해 낮은 성능을 보여 주지만, 작업량이 많아짐에 따라 완만하게 성능이 저하될 뿐만 아니라 작업량이 많아짐에 따라 CVM 보다 높은 성능을 보여 주었다. 이러한 성능 차는 작업량이 많아질수록 현저하게 나타나고 있다.

세 번째 Network 처리 성능 테스트는 RMI와 TCP를 통해 이루어 졌다. 각 테스트는 10번씩 이루어 졌으며 이에 대한 결과는 아래 [그림 2,3]과 같다.

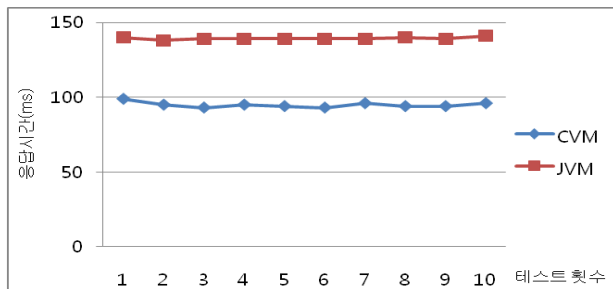


그림 2. RMI 통신 시의 응답시간

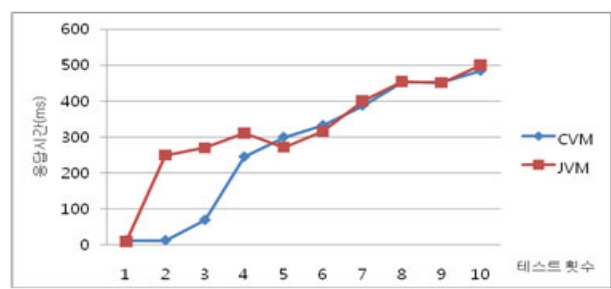


그림 3. TCP/IP 통신 시의 응답시간

위의 결과로 볼 때에 RMI, TCP 통신에서 CVM은 비교적 좋은 성능을 보여주지만 통신 방법에 따라 성능에 차이가 있음을 알 수 있다.

#### 4. 결론

CVM과 JVM의 성능 평가 실험의 결과를 통하여, 다음과 같은 몇 가지 결론을 내릴 수 있다.

- (1) 작업 처리량이 적을수록 CVM은 JVM보다 좋은 성능을 보여 주었다.
- (2) 작업 처리량이 많아질수록 CVM은 JVM보다 낮은 성능을 보여 주었다.
- (3) 작업 처리량의 증가량이 많아질수록 CVM의 성능 저하는 급격하게 나타났다.
- (4) 네트워크 부분에서는 대체적으로 CVM이 JVM보다 좋은 성능을 보여 주었다.

이러한 결과로 볼 때 임베디드 디바이스를 타겟으로 하는 JavaME의 Connected Device Configuration (CDC)의 CVM 가상 머신의 성능은 X-Hyper320TKU 임베디드 디바이스 상에서 애플리케이션의 작업 처리량에 따라 현저하게 차이가 나며, 임베디드 디바이스에서 JVM이 사용 가능하다면 애플리케이션에서 주로 사용하는 기능과 예상 작업 처리량에 따라서 CVM과 JVM을 선택하여 사용하는 것이 보다 향상된 애플리케이션의 성능을 얻을 수 있다고 할 수 있다.