



Java Technology Seminar  
JavaOne 2008 Wrap-up

# Java + You

JCO 회장 최상훈



# Every May...



- JavaOne Conference Season !!
  - Java Issue maker
    - Ajax, JPA, Groovy, (OSS) Java SE 6, EJB X.0, JavaFX ...

# 2008 JavaOne Issues



- JVM Optimization
- RIA
- Script Languages
- Open Source
- SOA
- ...

# Java + You



# JVM Optimization 1



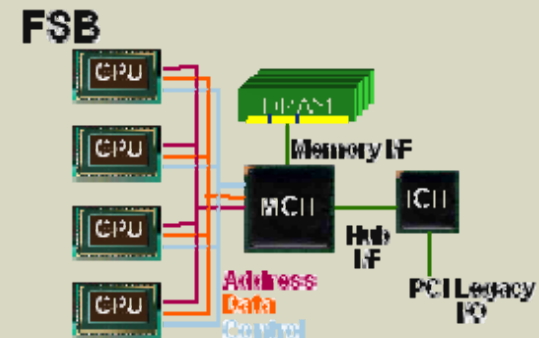
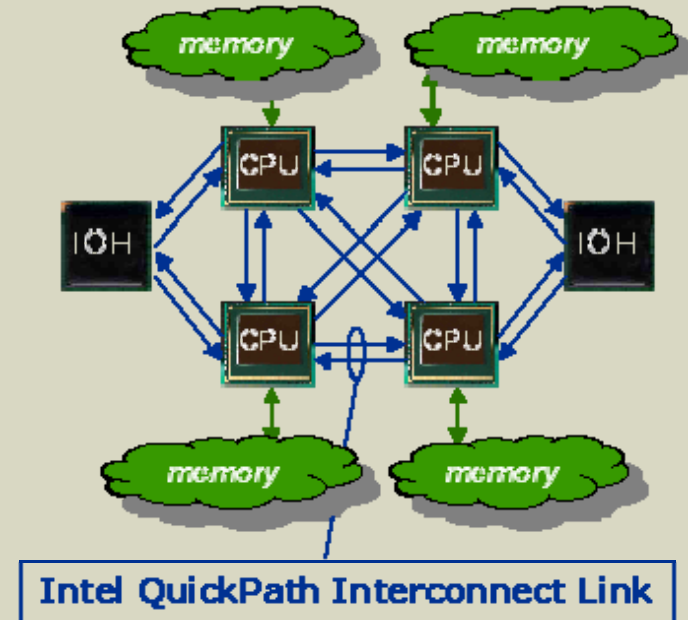
- Java + Intel Processor

		Java Language											
		java	javac	javadoc	apt	jar	javap	JPDA	JConsole				
		Security	Int'l	RMI	IDL	Deploy	Monitoring	Troubleshoot	Scripting	JVM TI			
		Deployment			Java Web Start				Java Plug-in				
		AWT			Swing			Java 2D					
		Accessibility	Drag n Drop	Input Methods		Image I/O	Print Service	Sound					
		IDL	JDBC		JNDI		RMI	RMI-IIOP					
		Beans	Intl Support		Input/Output	JMX	JNI	Math					
		Networking	Override Mechanism		Security	Serialization	Extension Mechanism	XML JAXP				Java SE API	
		lang and util	Collections	Concurrency Utilities		JAR	Logging	Management					
		Preferences API	Ref Objects	Reflection	Regular Expressions	Versioning	Zip	Instrumentation					
		Java Hotspot Client VM					Java Hotspot Server VM						
		Solaris		Linux		Windows		Other					
JDK													
JRE													
lang and util Base Libraries													
Java Virtual Machine													
Platforms													

# JVM Optimization 2



- Intel QuickPath Interconnect
- Scalable solution
  - Much higher link bandwidth than Front Side Bus (FSB)
  - Many system topologies with more than four processors supported



# JVM Optimization 3

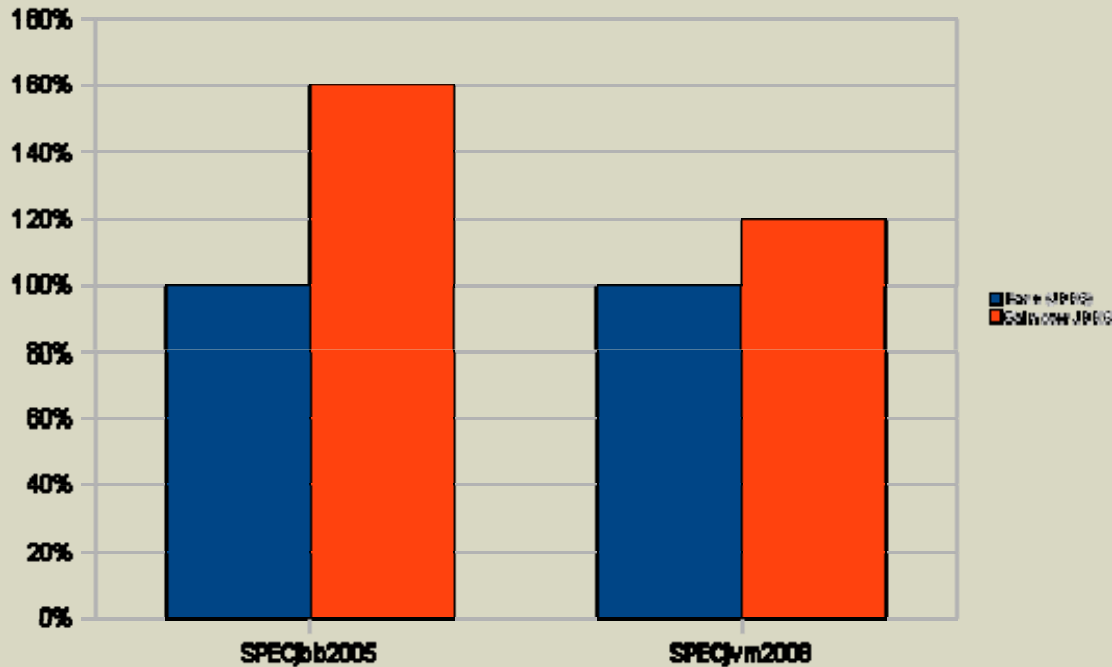


- Code Generation & Processor Specific Optimizations
  - Profiling Support: triggering method compilation, inlining, devirtualization..
  - Register allocation and assignment
  - Use of Itanium processor relative calls instead of indirect calls
- Allocation prefetch
- Tune register allocation to reduce memory traffic

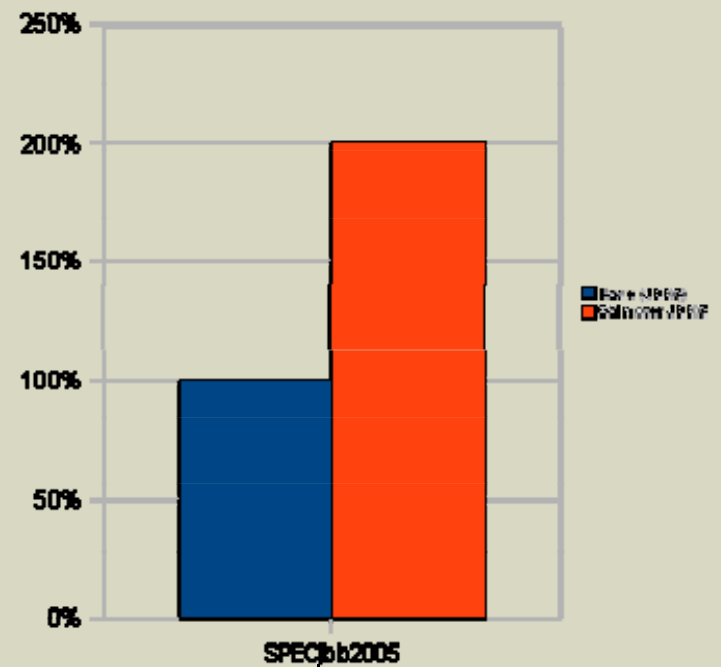
# JVM Optimization 4



Performance Gain on Intel Xeon processors



Performance Gain on Intel Itanium processors



# JVM Optimization 5



- S/W Transactional Memory VS. H/W ...

```
static void move(Queue q1, Queue q2) {  
    synchronized (q1) {  
        synchronized (q2) {  
            v = q1.dequeue();  
            q2.enqueue(v);  
        }  
    }  
}
```



```
static void move(Queue q1, Queue q2) {  
    atomic {  
        v = q1.dequeue();  
        q2.enqueue(v);  
    }  
}
```

SDK



Market

# Q & A



연락처 :  
jco5th@gmail.com



**이 저작물은 크리에이티브 커먼즈 코리아 저작자표시-비영리-  
동일조건변경허락 2.0 대한민국 라이선스에 따라 이용하실 수 있습니다.**

This work is Licensed under Creative Commons Korea Attribution 2.0 License.