CS1ADS — Algorithms and Data Structures

Aims and Goals of this Session

- Introduce distributed programming and RMI.
- Some issues in portability.
- Introduce a final few thoughts and signposts for further Java learning.

Parallel and Distributed Systems

- As you will discover further in CS3CCN Computer Communications and Networks and CS3DSM Distributed Systems, we can have concurrency beyond that of multi-tasking on a single processor.

Parallel Systems

- Parallel systems are where we have many processors connected together as bus speeds and with no communications errors.
- Communication is at very high speed and the processors are tightly coupled, perhaps even sharing the same system clock.

Distributed Systems

- Distributed systems are computer systems connected together using network technology.
- Communications is relatively slow and can be error prone.
- Such systems are loosely coupled.

Parallel and Distributed Systems

- The two are essentially distinguished by the granularity of activity.
- Parallel systems are based on fine-grain concurrency.
- Distributed systems are based on course-grain concurrency.

Networks

- Networks these days usually means the Internet which uses the TCP/IP protocols.
- There are other protocols, e.g. ATM.

Sockets

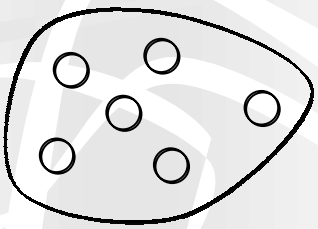
- Ports and sockets are the mechanisms of communication over the Internet.
- This is very low-level and should be eschewed.

Remote Method Invocation

- Java introduces the notion of distributed object communicating over the Internet: Remote Method Invocation (RMI).
- The idea here is that objects can exist on different machines and nonetheless communicate with each other and indeed be considered components of a single program.

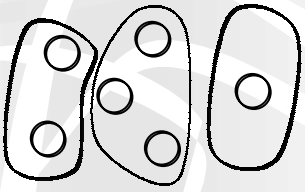
Remote Method Invocation

- Multi-tasking on a single processor.



Remote Method Invocation

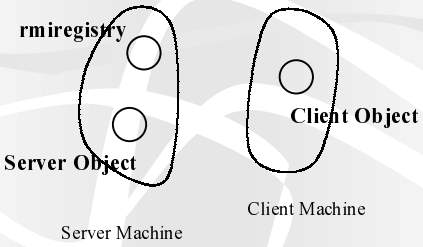
- Distributing objects across many processors.



Remote Method Invocation

- RMI is a distributed Java system support package that can be used for applications and applets.
- RMI can by used for both distributed applications and distributed applets.

Remote Method Invocation



Remote Method Invocation

- RMI is an abstraction at the object level.
- RMI is higher level than sockets.
- RMI is object based rather than network based
- RMI should be used as it wraps up socket management.

Abstraction

- RMI is a lesson in abstraction and package construction.
- From low-level networks we can construct low-level abstraction: InetAddress, Socket.
- On top of this we can build RMI.

Abstraction

- Abstraction is the very foundation of programming.
- Abstraction is the very foundation of software engineering.

Abstraction

- Collections
- Java Generic Library (JGL).
- Java Foundation Classes.
- Java Media Framework.
- JavaMail.
- JavaSound.
- Java2D, Java3D.

JavaBeans

- A protocol for a defining components that can be manipulated within a visual development environment.
- Instead of programming things explicitly we use a WYSIWYG GUI editor.

Enterprise Java Beans

- EJB is an increasingly important technology.

Portability

- A fundamental touchstone of abstraction is portability.
- Although we have a large number of operating systems, we want to avoid having to reprogram systems when we move from one operating system to another.

Portability

- Java supports portability by abstracting away from the operating system.
- Java provides a virtual operating system interface.

Portability

- Portability means that the same program source can be executed, hopefully without observable change in behaviour on different systems.
- Java extends this and lets the same compiled code be portable.

Portability

- Java achieves portability by defining a virtual machine (the Java Virtual Machine — JVM).
- Currently this is implemented as an interpreter of JVM instruction.
- In the future we could see JVM chips.

Politics

- Microsoft want to ensure total hegemony of the operating system market.
- They want control of Java and to make people use Windows specific features.
- This will bind people and applications to Windows.

Politics

- Sun want to maintain freedom for Solaris to exist and hence are pushing absolute portability of code: Pure Java.

Politics

- Java programs should be fully portable across all platforms.
- Do not use operating system specific features.
- Fight exclusivity imperialism.

Politics

- If people want to be operating system specific, use C++ or operating system specific languages, e.g. Visual Basic.

Java vs. C++

- Roughly:
  - C++ is for real-time applications.
  - Java is for everything else and especially UI related applications. NB Distinguish client-side from server-side when dealing with distributed applications.

Java vs. JavaScript

- Java is heavy-weight.
- Use JavaScript and animated GIFs for making Web pages pretty, not Java.
- Use Java for whole applications that are initiated across Web or for *servlets*.

Summary of this Course

- Done lots more Java programming.
- Have become more proficient.

Summary of this Course

- Have looked at various data structures and algorithms.
- Have looked at various algorithms for searching and sorting data.
- Have looked at various design issues, particularly with respect to packages in Java.

Summary of this Course

- Life-long learning means:
  - You never finish learning new techniques in programming.
  - All code really needs re-vamping in the light of new programming features, techniques, etc.

End of this Course