Internals of App Servers (Middlewre Systems)

Introductory Lecture

- Ramesh Loganathan

Course: Expected Learning

Understand

- Understand essence of middleware and distributed object technology
- App Server Architecture overview
- Deep dive on a contemporary distributed app platform
- Understand the design and architecture of the platform
- Understand underlying subsystems/technologies
- Implement the platform

• Via

- Lectures & Invited Talks
- Seminars
- Hackathons
- Course project
 - A complete functional project building a Middleware App Platform

Course Outline

- Lectures 4 or 5 3-hour classes per Module
 - M1- Understand essence of middlewares and distributed object technology
 - M2- App Server architecture
 - M3- Lifecycle of a Web Service request
 - M4-Things "in" the Internet
 - M5- Project problems Discussions
 - M6- Project problems Discussions
 - M7- Project problems Discussions
 - M7- Project problems Discussions
- Labs L1 to L8 (16 hrs)
- Seminars 6 hrs (6 groups, 1 hr each)
- Mini Project review 3-4 hrs

Labs.. Based on the project picked for course

- L1- Server clusters setup
- L2- One client talking to two servers
- L3-TBD
- L4-TBD
- Mini Project

Seminars

- Seminar Topics based on Project
 - Topics will be assigned thru the course.
 - Teams will be formed in the beginning.
- Possible areas:
 - IoT apps, standards & platforms
 - SOA related standards
 - .net internals vs. J2EE internals
 - Cluster considerations when developing J2EE apps

Project

- Start thinking about possible projects
 - Should be an Application Platform (to build and deploy apps)
 - Should be distributed (multi-server) environments
 - Should be contemporary
- Recent year projects
 - Microservcies latform
 - IoT App Platform
 - IoT Basics
 - Realtime cloud analytics engine (on a distributed platform)
 - Multi-tenanted SOA app platform
 - SOA Internals/ Mediation and Control
 - Distributed Long-running web transactions

Assessment

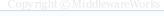
		WEIGHTED					
Quiz Total	Lab Total	Hackthons (1+2)	Requirement & Design Doc Marks	Pre-Demo	Final Demo	End Sem	Total
5	15	10	15	10	15	30	100

Beginnings..

- Form Teams- for seminar and project
 - 2 teams. 5-6 each.
- Self Driven Java Learning
 - Use Labs & TA's help
- Start thinking about Seminar ideas
 - As lectures progress seminar topics will be finalised
- Start thinking about project ideas
 - will be finalized in 2 weeks
- Through the course, interaction is required
 - Very difficult to share deeper perspective thru a 1-way talk
 - More the questions, merrier the learning
- To encourage this, we will have grading for CP! ☺

Questions?

RameshL@progress.com



Moving to Server Helps...

- "Power" components no more limited by client capability
- Pooling, sharing of scarce resources now possible from server-side
- Replication and distribution of application
- High scalability
- Design efficiency only limit to performance
- Support hundreds to millions of concurrent users
 - Incremental support from server-side for increasing number of concurrent users

...Multi-tier?

- Simplify the enterprise application scenario
- Remodel application development in three tiers
 - Tier I: Presentation Logic Layer
 - Tier II: Business Logic Layer
 - Tier III: Data Access Logic Layer
- Move business and data access logic to same/separate servers
- Keep only presentation logic on Client

...But new challenges appear

- Middleware systems are complex
- To build a proprietary middle-tier framework, considerations are:
 - Multi-threading
 - Resource sharing
 - Replication
 - Load balancing
 - Various services to be able to work with each other
- Compatibility with widely heterogeneous middleware services

Solution: the application server

- Application Server Sits in the middle tier
- Automates complex multi-tier features
- Manages, recycles scarce system resources
 - processes, threads, memory, db links, network sessions
- Load balances to share processing across multiple systems
- Access to infrastructure services
 - naming/directory, txn, persistence, security
- Specific runtime environment

Where it fits in

