Optimization Services (OS)

-- Providing Solvers As Services
-- Distributed/Parallel Computing System
-- Enterprise Computing Infrastructure
-- The Operations Research Internet/Intranet

Jun Ma

Industrial Engineering and Management Sciences
Northwestern University

IEMS Advisory Board, 03/06/2006
Optimization Services as Computing Utility Services

Intelligent User View

[Diagram showing a network of computers and services connected through Grid, Workstation, Registry, Intelligent Agent, Soap/XML, WSDL, Web Page and Service Server, IDLE COMPUTER, NEOS, USER, Computing Socket, model/problem, and model/problem.]
Centralized Vs Decentralized Enterprise Systems
Simplified System View

Centralized

Decentralized
Motivation

M \times N \text{ drivers}

M + N \text{ drivers}
Motivation

1. Tightly-coupled implementation
2. Various operating systems
3. Various communication/interfacing mechanisms
4. Various programming languages
5. Various benchmarking standards

- The key issue is communication, not solution!
- ... and Optimization Services is intended to solve all the above issues.
Optimization Services as Computational Internet/Intranet
Sketch View
Why is it the OR Internet/Intranet Analogy
Standard -- Optimization Services Protocol (OSP)
Technical Background

GET /xt/services/ColorRequest HTTP/1.0
Content-Length: 442
Host: localhost
Content-Type: text/xml; charset=utf-8
SOAPAction: "/getColor"

<soap:Envelope>
  <soap:Body>
    <OSP Protocol:
      String solve(String instance)
      input string instance follow OSI-L
      -- output string follow OSrL
    </OSP>
  </soap:Body>
</soap:Envelope>

SOAP is usually wrapped under HTTP

The 7-layer OSI Model
The 4-layer Internet model
Business Values
Business Values

Solve more types of computational problems more efficiently

Easily deploy enterprise computing system within a company, with intelligent components in scheduling computational jobs, registering and finding computing services, routing maintenance

Provide computational software as services on dedicated servers

Let all computational software communicate with each, independent of platforms and implementations.

Save costs on expensive software licenses

Make full use of limited computational solvers