Caching Strategies for Data-Intensive Web Sites

Presented by
Hossein S. Attar

Outline

- Introduction on Site Management Systems
- Introduction on Caching
- Weave
- Conclusion
Site Management Systems

- Traditional Dynamic Web Applications
- Site Management System Approach
  - Idea: separation of structure and content from graphical presentation
  - Benefits
    - Easier development and maintenance
    - Different views over data
    - Explicit site structure, needed by performance-improvement solutions

Declarative Site Specification in Weave

- A Web site:
  - Description of site structure and content
    - Site schema
    - Site class
  - A set of XSLT programs
    - XSLT = eXtensible Stylesheet Language for Transformations
    - Evaluates, rearranges, and reassembles the information in the XML source document
define class CustomerNat (SNK) 
  {instances using Q0 ;
  data nation_name using Q1 ;
  link Customer to Customer(SCK) using Q2 ;
  define query Q0 as select nationkey as SNK from Nation ;
  define query Q1 as select name as nation_name from Nation where nationkey=SNK ;
  define query Q2 as select custkey as SCK, name as anchor from Customer where nationkey=SNK ;
Caching

- Overview
- Materialization Strategies
  - Purely dynamic (no materialization)
  - HTML page materialization
  - XML fragment materialization
  - Query-result materialization
    - Materialized Views
    - Cache Functions
- No universally good strategy

Weave

- A Web site management system with caching capability
- 3-tier customizable cache system
- WeaveRPL and policy files
- Statistics Manager
Conclusion

- A site-management system for easy development and maintenance
- 3-tier caching with complex runtime policies
- Extensible with automatic policy generation
- Support for distribution and replication