



Web Programming with SMLserver

Niels Hallenberg

joined work with Martin Elsman

IT University of Copenhagen

PADL'2003 Talk, January 13, 2003

What is SMLserver?



SMLserver is a Web server platform for Standard ML programs

FEATURES/INITIAL REQUIREMENTS:

- ◆ SMLserver has a rich interface for writing Web applications, including:
 - Easy access to form data and request header information, including cookies
 - Access to a variety of RDBMSs through an efficient generic interface that supports reuse of database connections
 - Support for type safe data caching, fetching of foreign Web pages, filtering and scheduling of script execution.
- ◆ Programs are compiled into bytecode files, which are loaded only once but may be executed many times
- ◆ A multi-threaded execution model allows multiple requests to be served simultaneously



STATIC TYPING:

- ◆ Web applications are exposed to users early
- ◆ Web applications change rapidly
- ◆ Static typing can ease maintenance and improve robustness

HIGHER-ORDER FUNCTIONS:

- ◆ High degree of code reuse and shallow interfaces
- ◆ Good for separation of development tasks (design vs. application logic)

A RICH MODULES SYSTEM:

- ◆ Name space management
- ◆ High degree of code reuse

WHY THE ML KIT:

- ◆ No reference tracing garbage collection and no tags, but region based memory management

What do SMLserver scripts look like?



```
val time_of_day =
  Date.toString (Date.fromTimeLocal(Time.now()))
val _ = Ns.Conn.return
  `<HTML>
    <BODY><H1>Hello world!</H1>
      The current date and time is ^time_of_day
    </BODY>
  </HTML>`
```

ML SERVER PAGES:

```
<HTML>
  <BODY><H1>Hello world!</H1>
    The current date and time is
    <?MSP= Date.toString (Date.fromTimeLocal(Time.now())) ?>
  </BODY>
</HTML>
```



EXAMPLE PROJECT FILE:

```
import ../lib/lib.pm
in
  local
    ../demo_lib/Page.sml
  in
    [
      (* SMLserver Script Files *)
      demo/time_of_day.sml
      demo/time_of_day.msp
    ]
  end
end
```



MOTIVATION:

- ◆ caching of session information
- ◆ caching of information obtained from foreign Web sites (e.g., currency rates, stock quotes)

SMLserver provides the function

$$\text{Cache.memoize} : (\alpha, \beta) \text{cache} \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta$$

Given an appropriate cache c , the application

```
Cache.memoize c f
```

returns a function f' which is a memoized version of f .



A cache is obtained by the function

$$\text{Cache.get} : \alpha \text{ Type} * \beta \text{ Type} * \text{name} * \text{kind} \rightarrow (\alpha, \beta) \text{cache}$$

Cache types ($\alpha \text{ Type}$) are constructed using base types and type constructors:

BASE TYPES:

Int: int Type

Real: real Type

TYPE CONSTRUCTORS:

Pair: $\alpha \text{ Type} \rightarrow \beta \text{ Type} \rightarrow (\alpha, \beta) \text{ Type}$

Option: $\alpha \text{ Type} \rightarrow \alpha \text{ option Type}$

List: $\alpha \text{ Type} \rightarrow \alpha \text{ list Type}$

CACHE KINDS SUPPORTED:

Caches based on size and on time



GENERIC DATABASE INTERFACE:

- ◆ Support for Oracle, Postgresql, and MySQL

DATABASE POOLING:

- ◆ A handle identifies a connection to an RDBMS
- ◆ SMLserver maintains a configurable number of pools (of handles)
- ◆ A database handle is owned by at most one script at a time
- ◆ Handles are requested and released by the database functions in such a way that no deadlocks appears
- ◆ The programmer needs not know about handles, unless transactions are used



```
signature NS_DB =
  sig
    val dml      : quot -> unit
    val foldr    : ((string->string)*'a->'a)->'a->quot->'a
    val qqg     : string -> string
    ...
  end
```

NOTES:

- ◆ Quotations are used for embedding SQL
- ◆ Function `qqg` escapes quotes (') for SQL string embedding
- ◆ Function `dml` for executing `insert` and `update` statements
- ◆ Function `foldr` folds over the rows returned by a SQL statement (similar to `List.foldr`)



DATA MODEL:

```
create table guest (  
    email    varchar(100),  
    name     varchar(100),  
    comment  varchar(2000)  
);
```

ADDING A NEW GUEST:

```
val _ = Db.dml  
    `insert into guest (name,email,comment)  
    values (^(Db.qqqq n),^(Db.qqqq e),^(Db.qqqq c))`
```

NOTES:

- ◆ Function `qqqq` escapes quotes (') for SQL string embedding



SHOW CONTENT OF THE GUEST BOOK:

```
fun layoutRow (g: string->string, acc:quot) =
  `<li> <i>^(g "comment")</i>
  -- <a href="mailto:^(g "email")">^(g "name")</a>
  <p>` ^^ acc

val rows = Db.foldr layoutRow ``
  `select email,name,comment from guest`

val _ = Page.return "Guest Book"
  (`<ul>` ^^ rows ^^ `</ul>` ^^
  `<form action=guest_add.sml>...</form>` )
```

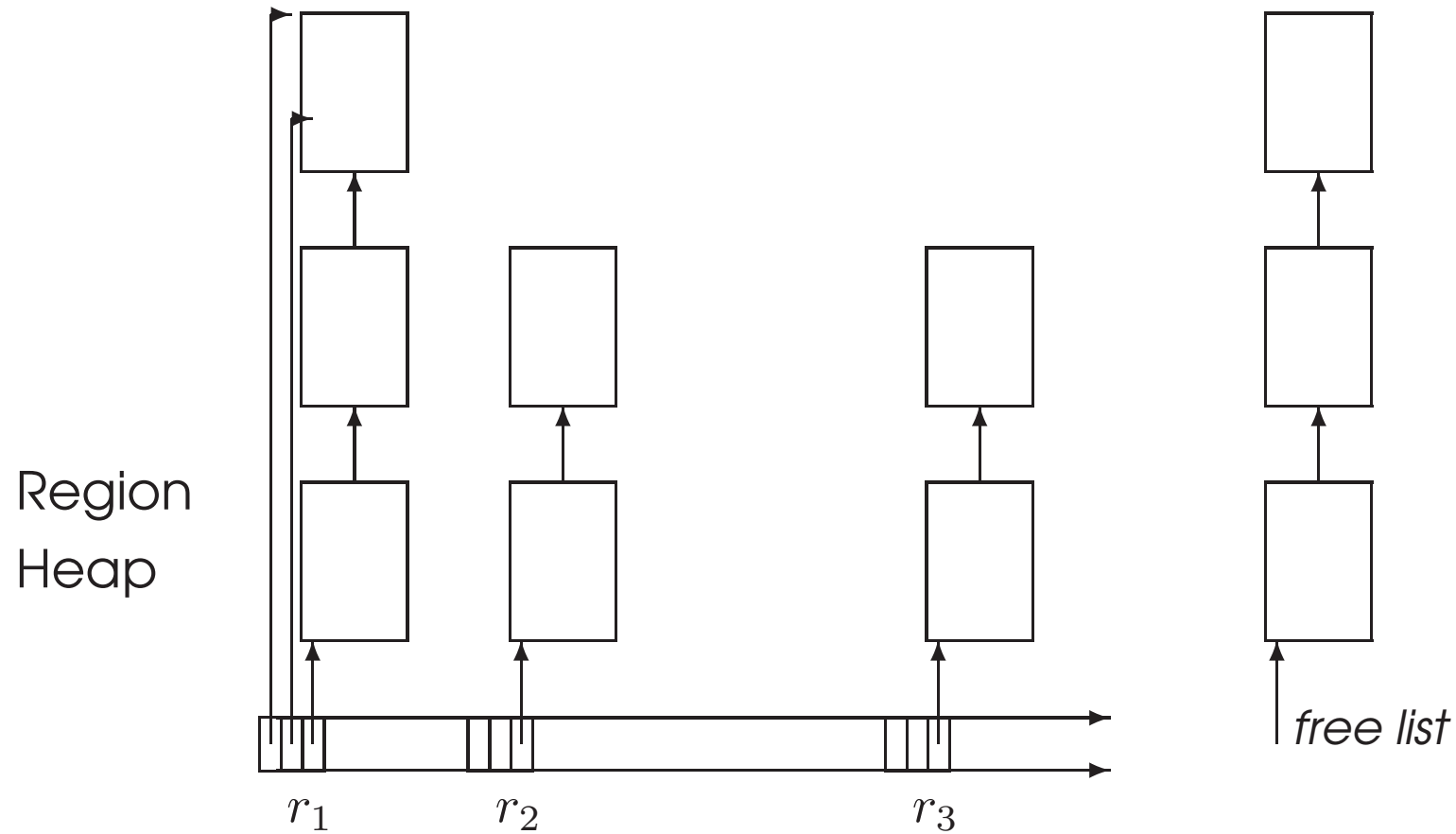
NOTES:

- ◆ Library function `Page.return` sends a page back to the client
- ◆ The first argument to `foldr` is a function for formatting a row



- ◆ Memory allocation and deallocation directives are inserted in the program at compile time
- ◆ Memory is allocated from a *free list* of region pages
- ◆ A region is a list of **region pages** — a **free-list** contains region pages currently unused
- ◆ Activation records stored on the runtime stack contain: **temporary variables** and **region descriptors**

The Region based memory model

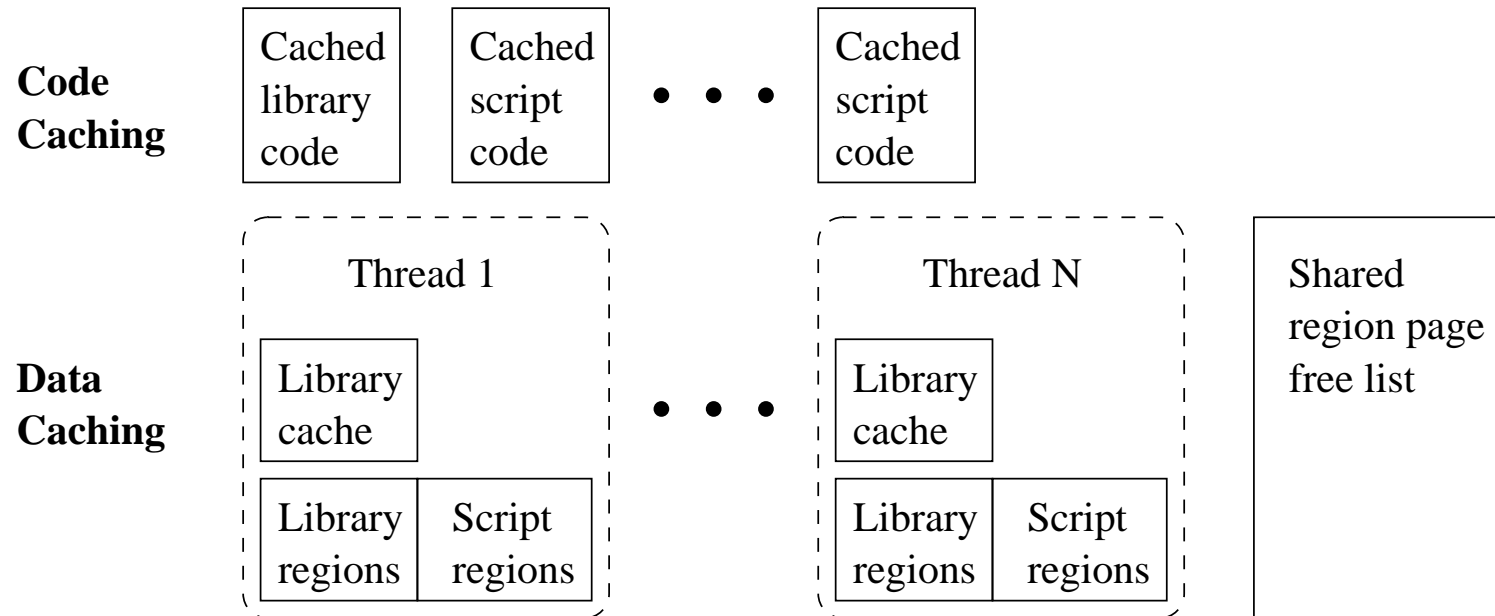


Machine Stack with Activation Records

The SMLserver memory model



EFFICIENT THREAD SAFE MEMORY MODEL:



WHEN SERVER BOOTS:

- ◆ load library code
- ◆ execute library code
- ◆ copy "Library regions" into "Library cache"

WHEN EXECUTING SCRIPT:

- ◆ load script code
- ◆ execute script code
- ◆ deallocate script regions
- ◆ restore "Library regions"

Measurements with ApacheBench



Program	Requests / second			
	MosML MSP	AOLserver TCL	Apache PHP	SMLserver MSP
hello	55	724	489	1326
date	54	855	495	1113
db	27	558	331	689
guest	25	382	274	543
calendar	36	27	37	101
mul	50	185	214	455
table	21	59	0.7	93
log	8	12	0.4	31

- ◆ ApacheBench (v. 1.3d) uses eight threads (60 seconds each)
- ◆ 850Mhz Pentium 3 Linux box (384Mb RAM)

Measurements with ApacheBench



Program	Requests / second		
	SMLserver MSP	No script caching	With library execution
hello	1326	916	349
date	1113	744	337
db	689	516	275
guest	543	356	249
calendar	101	69	80
mul	455	300	241
table	93	84	75
log	31	30	28

- ◆ Caching of loaded script bytecode improves performance (3 - 53 pct).
- ◆ Execution of library code degrades performance (10 - 74 pct).



At The IT University of Copenhagen, SMLserver is used for

- ◆ a course evaluation system
- ◆ an alumni system
- ◆ an online application system
- ◆ an employee database
- ◆ a survey and questionnaire system

CONCLUSION:

SMLserver scales to large Web sites



- ◆ The Bigwig project at BRICS (Brabrand, Møller, and Schwartzbach). Type safe generation of HTML code.
- ◆ The *HaskellDB Project* and work by Buneman and Ohori: *Polymorphism and Type inference in Database Programming*, TODS'96. Type safe construction of SQL.
- ◆ The Haskell WASH/CGI library by Thiemann: *Wash/CGI: Server-side Web scripting with sessions and typed compositional forms*, PADL'02.
- ◆ Using continuations to implement the interaction between clients and Web servers. Queinnec: *The influence of browsers on evaluators or, continuations to program web servers*, ICFP'00 and Graunke et al.: *Automatically restructuring programs for the web.*, ASE'01
- ◆ The ML Server Pages implementation for Moscow ML by Sestoft