KLIC: A Portable Implementation of KL1

ICOT
First Research Department

Tetsuro Fujise Takashi Chikayama
Kazuaki Rokusawa Akihiko Nakase

Contents

Motivation

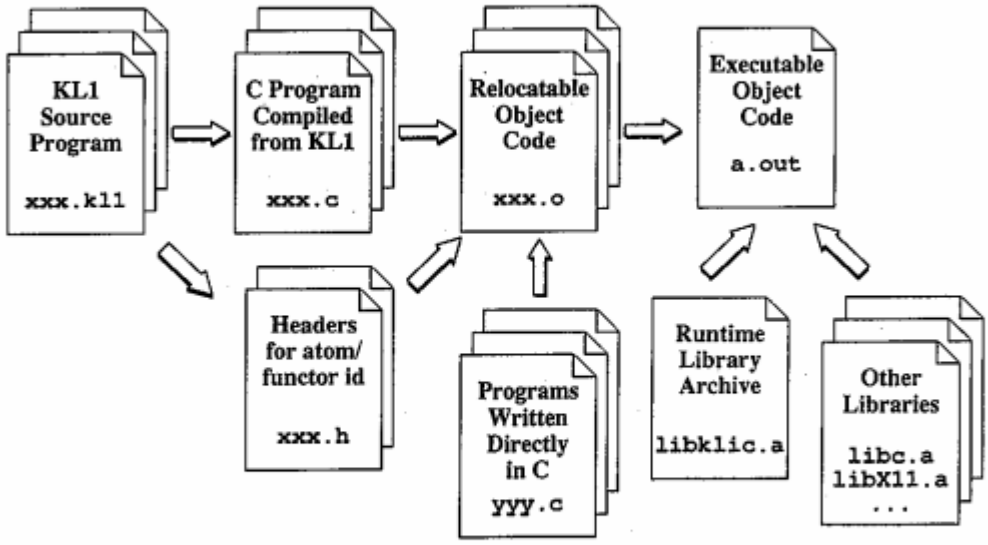
- KL1: A Concurrent Logic Programming Language
- KL1 impl. on PIMs
  - Efficient but not portable (only for dedicated HW)
  - Difficulty in making experimentations
    Much low-level optimization in impl. core

⇒ Portable and Extensible Implementation

C as the Intermediate Language
— Merits of Compilation —

- Portability
- System-dependent Low-Level Optimization by C Compiler
- Ease in Linkage with Programs in Other Languages

Compilation and Linkage of KL1 Programs



Efficiency Problems/Solutions (1)

- Costly function calls
  - One "module" as one function
- Inability to control register allocation
  - Global data are cached on local variables possibly on registers

Efficiency Problems/Solutions (2)

- Cost of provision for interrupts
  - Signal handlers only set a flag
  - The flag check can be combined with heap overflow checks
- Large object code size
  - runtime routines for exceptional cases

Portability of Sequential Core

Model	OS	Manufacturer
SparcStation	SunOS, Solaris	Sun Micro.
DEC 3000	OSF/1	DEC
RS 6000	AIX	IBM
HP 9000	HP-UX	HP
IRIS	IRIX	SGI
EWS 4800	EWS-UX/V	NEC
Luna 88k	Mach	OMRON
M-880	HI-OSF/1-MJ	Hitachi
S-3800	HI-OSF/1-MJ	Hitachi
IBM AT clone	MS-DOS, OS/2, Linux	IBM etc.

Generic Objects

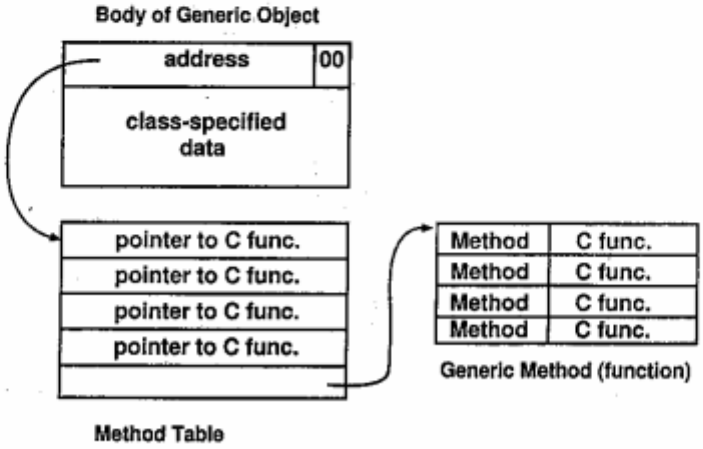
- Framework for system extension
- No need to touch impl. core for extension
- For various purposes
  - Some of built-in data types
  - Non-local memory references

    Different parallel impl. schemes upon the same core
  - Foreign language I/F better than subroutines
- 3 categories depending on how methods are called

Representation of Body

Method tables for:

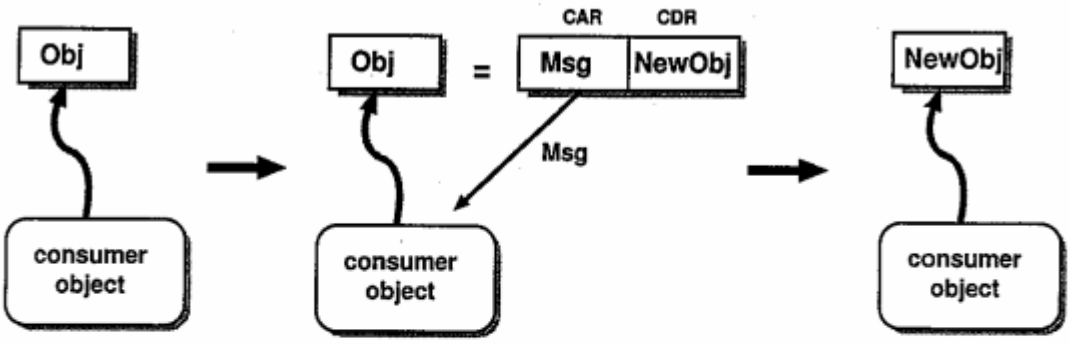
- Standard: unification, GC, encode, ...
- Class-specific: dispatches on method names



Three Categories of Generic Objects

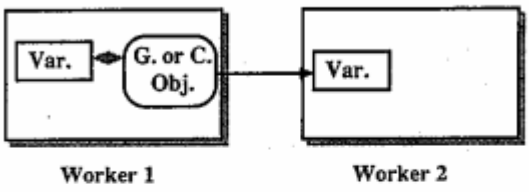
- Data Objects: Normal "immutable" data
  Manipulated thru explicit "method" calls
  = simple data
- Consumer Objects: Data-driven processes
  Associated with a var.; awaken by unification
  = suspended goals
- Generator Objects: Demand-driven processes
  Associated with a var.; awaken by dereference
  no KL1 counterparts

Consumer Object as Process

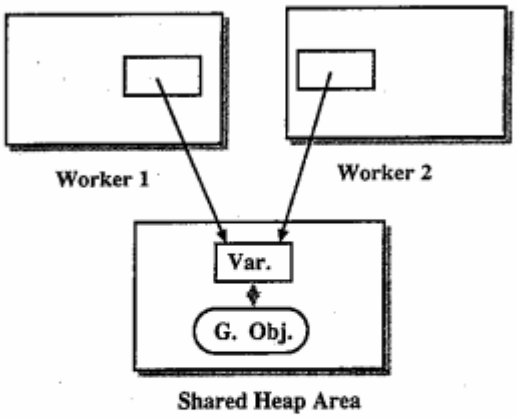


Obj = [Msg| NewObj]

Parallel Implementation

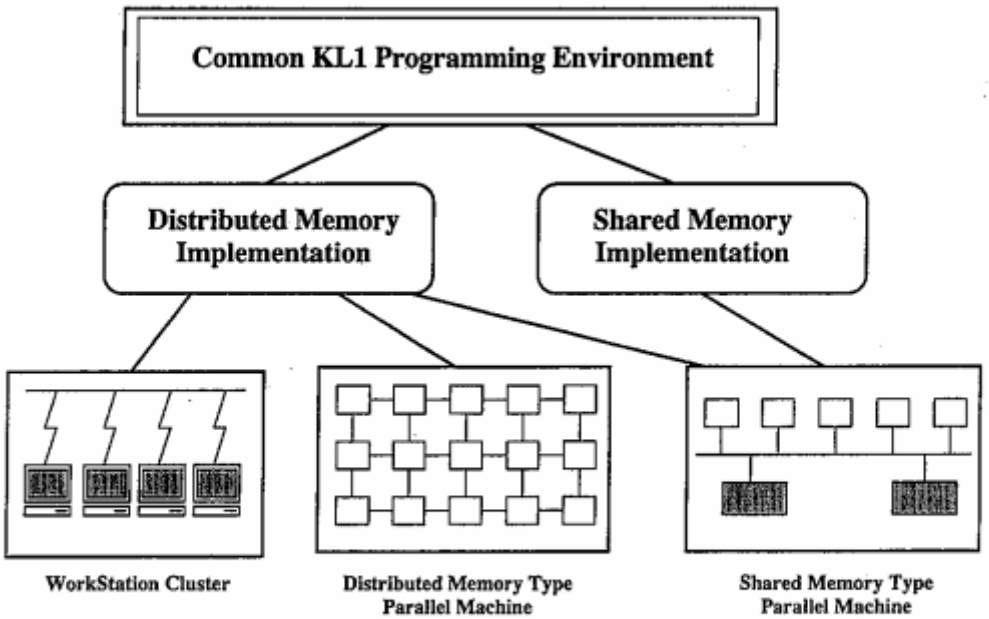


Distributed Memory Implementation

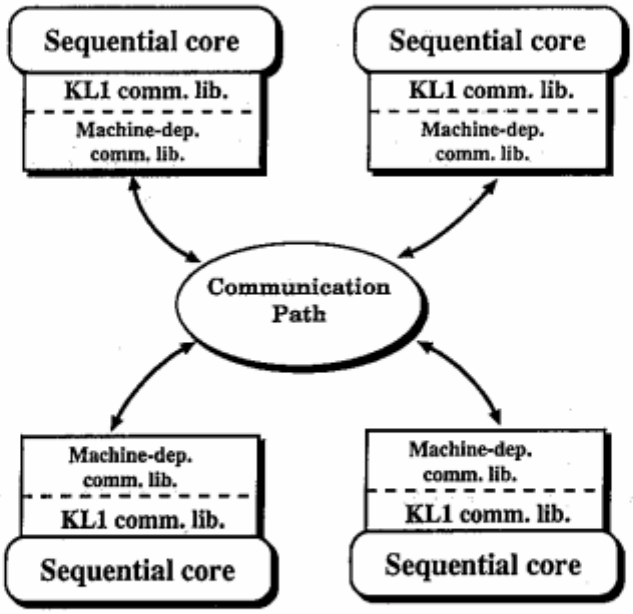


Shared Memory Implementation

Target Architecture



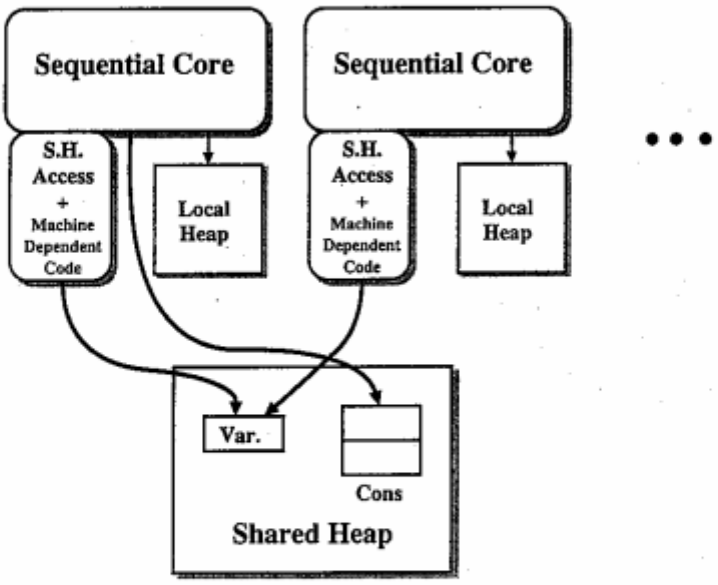
Distributed Memory Implementation



The Machine-Dependent Communication Library

- Initiation
- Message passing
  - General purpose message-passing library
  - Shared memory
  - Machine-specific communication path

Shared Memory Implementation



Portability of Parallel Implementation (1)

Model	OS	Dist.Impl.	Shared Impl.	Manufacturer
SparcStation	SunOS	PVM		Sun Micro.
	Solaris	PVM		
SparcCenter	Solaris	PVM	O	Sun Micro.
	Solaris	*S.M.**		
DEC 7000	OSF/1	PVM	O	DEC
RS 6000	AIX	PVM		IBM
Paragon	OSF/1	PVM		Intel
CM5	SunOS	CMAML		TMC
AP 1000†	‡	‡		Fujitsu
Cenju-3†	Mach	MPI		NEC
SR 2001†	HI-UX	Express		Hitachi

† porting. ‡AP 1000 has a private OS and a communication path.
**S.M.* means shared memory.

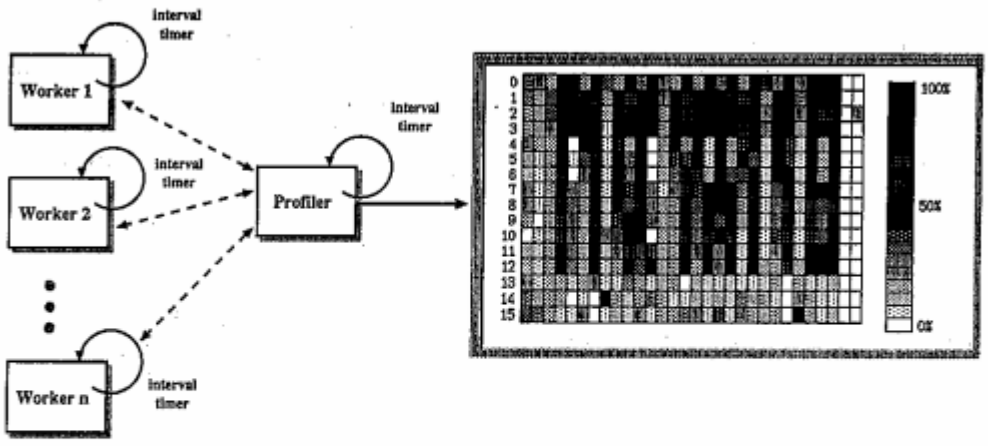
Portability of Parallel Implementation (2)

Code Lines

	KL1	C	Asm
Seq. Core	3,400	10,000	
Dist.Mem. Impl.			
KL1 Comm.		1,900	
Machine-Dep.		1,300	
Shared Mem. Impl.			
Shared Heap Acc.		3,300	
Machine-Dep.			150

KLIC: ver.1.511; Comm.Path: PVM; System: SparcCenter 2000

Runtime Monitor



Concluding Remarks

- KLIC can be ported on various systems easily
- Generic objects allowed extensible implementation
- Parallel implementations retain efficiency of the sequential core through generic objects

  ⇒ **These are based on the non-strict argument passing feature of KL1**
- Various efforts are on-going to make the system more useful in parallel software research