The Evaluation of PIM

ICOT
First Research Department

Kouichi KUMON Hiroyoshi HATAZAWA

Overview

- Evaluation of Single Processor Performance
  - Comparison among PIMs and KLIC Using small benchmarks including append and life
- Evaluation of Automatic Load Distribution
- Making Execution Models
- Evaluation of Inter-cluster Communication Cost

Difference between PIM and KLIC

- PIM:
  - Dedicated Hardware (tag handling insn)
  - MRB GC used
  - Large Memory: PIM/m(16MW) PIM/p(32MW)
- KLIC:
  - Developing Environment (UNIX)
  - Compiled into C-language

KL1 Processing Systems

- PIM/m: Distributed Parallel Processors
- PIM/{p,c,i,k}: Cluster-configured Parallel Processors
- KLIC: Running on Sparc10 Workstation
- KLIC/p: Sequential-core of KLIC for PIM/p
  - Almost the Same Object-Code Quality as Other WS.

Append RPS on Various Systems

System	clock (ns)	nrev1500 (KRPS)	nrev5000 (KRPS)	Ratio
PIM/m	65	600	411	.69
PIM/p	80	305	281	.92
PIM/c	66	55	56	1.02
PIM/i	240	65	65	1.00
PIM/k	100	76	76	1.00
KLIC(1)	28	1,151	1,173	1.01
KLIC/p	80	225	238	1.06
KLIC/p(2)	80	405	386	.95

(1) On SS10, time is measured by CPU time.

(2) The initial heap size is 10 mega words.

Append Code Analysis

- Main loop: 21 insns in KLIC, 28 insns in PIM/p
- MRB related insns: $> 20\%$
- Locking/Unlocking: 2 insns
- Non-negligible SHOEN management overhead

MRB Penalties

- Dereference and Type-checking
- Free List Management Cost
- GC Cost Proportional to Reduction
- Extra Move Instructions

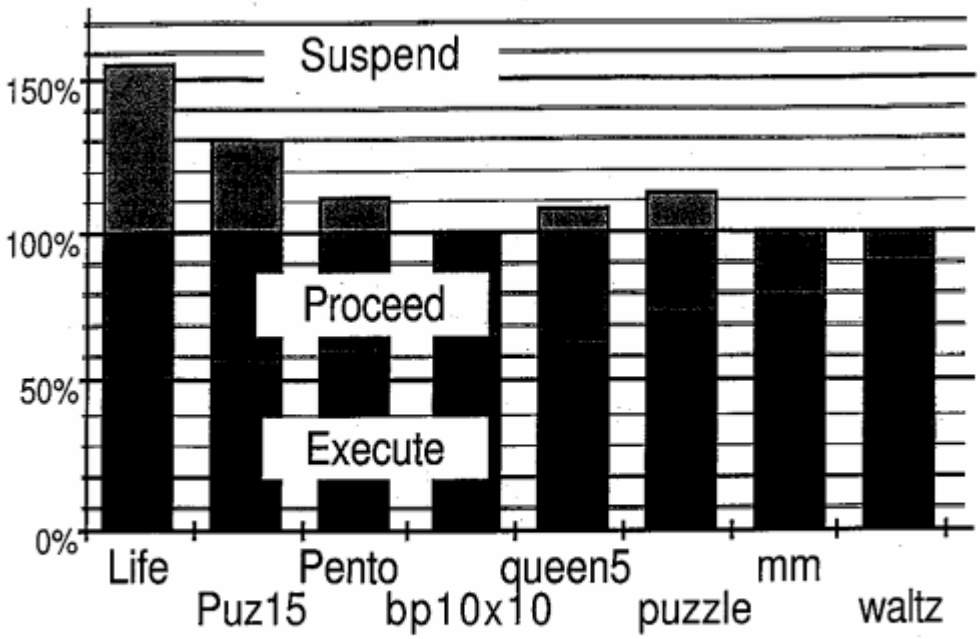
Characteristics of Append

- Core of Stream Operation
- # of Active-cell $\ll$ # of Consumed-cell
- Almost Execute operations only
- No Suspension

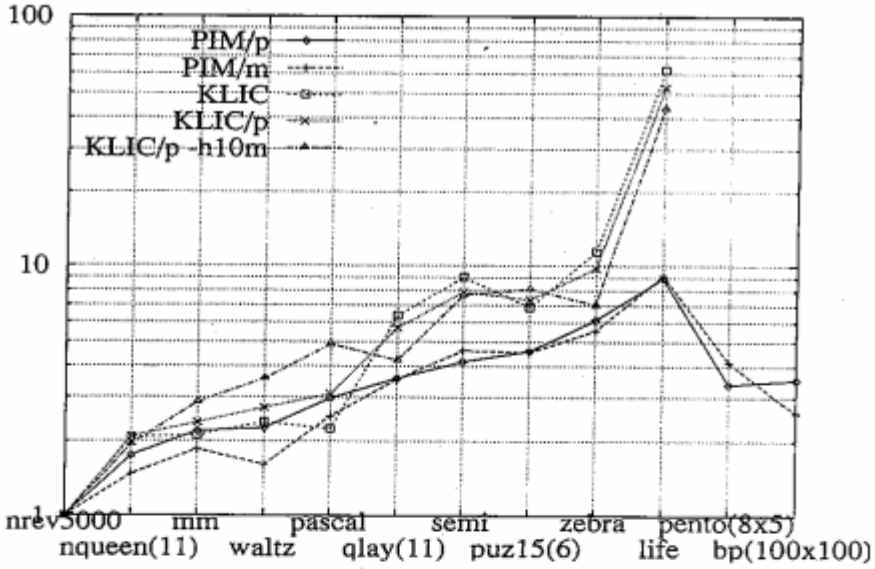
Classification of Executed Instructions

Category	PIM/p		KLIC/p
Type-check	6	$<$	7
MRB	6	$\gg$	–
Write	2	$<$	3
SHOEN	3	$>$	–
Exclusive	2	$>$	–
Deref	2	$>$	1
Allocation	2	$<$	3
Read	1	$<$	2
Slit-check	1	$<$	3
Move	1	$=$	1
Other	2	$>$	1
Total	28	$>$	21

Characteristics of Benchmarks



Relative Machine Performance



Life Program Analysis

- PIM Performance ≫ KLIC Performance
- Reasons for the Superior PIM Performance:
  - Goal Record Reclamation
  - Low Suspension/Resumption Overheads
  - Infrequent Stop-and-Collect GC

Parallel Execution Performance in a Cluster

- Automatic Load Distribution Available
- Performance Dependence on Distribution Method
- Making Execution Models for Simple Program `TP1`
- Measurement of Speed-up Ratio
- Estimation of Goal Distribution Overhead

Load Distribution Method

- Original VPIM Method:
  - One Goal Queue
  - No Preparation during Reduction
  - Goals Distributed by Busy Processor
- Modified VPIM Method:
  - Two Goal Queues: Local and Global
  - Preparation during Reduction
  - Distribution by Idle Processor

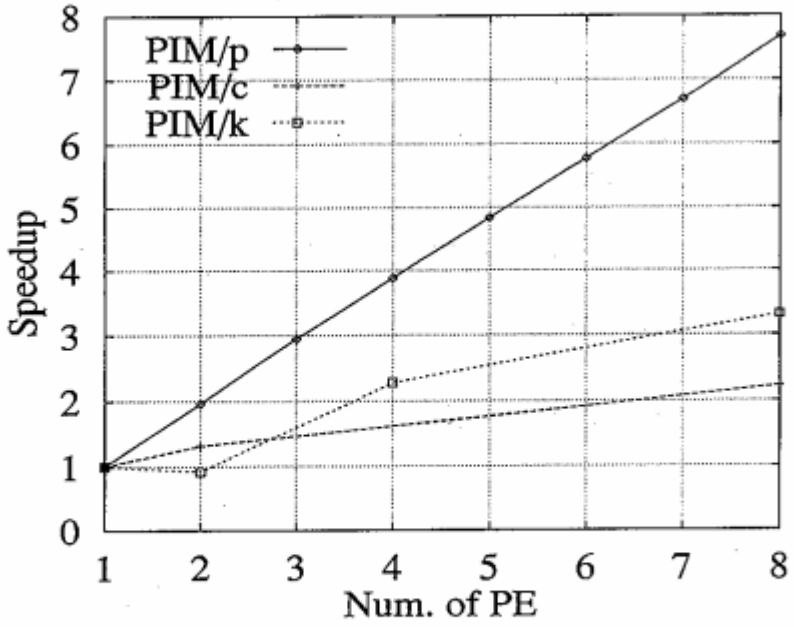
Characteristics of Distribution Methods (1)

- Original VPIM Method in PIM/c and PIM/k
  - Simple Goal Management Scheme
  - Low Overhead in Sequential Execution
  - Distribution Overheads on both Idle and Busy Processors
  - Good for Low Distribution Frequency

Characteristics of Distribution Methods (2)

- Modified VPIM Method in PIM/p
  - Complicated Goal Management
  - Effective for High Distribution Frequency

Cluster Performance Measurement



Load Distribution Cost in a Cluster

	Distribution Cost(μs)	Equivalent Reductions
PIM/p	17∼22	3∼4
PIM/c	239∼275	14∼16
PIM/k	94∼116	9∼11

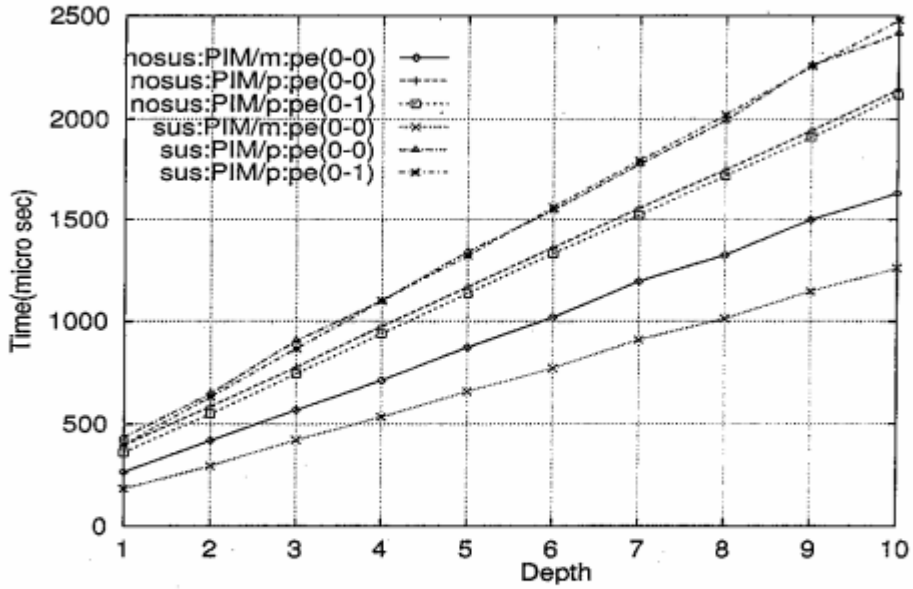
Distributed System Performance

- Throw Goal
- Meta-Level Management
- Data Transfer
  - Lazy Transfer Scheme

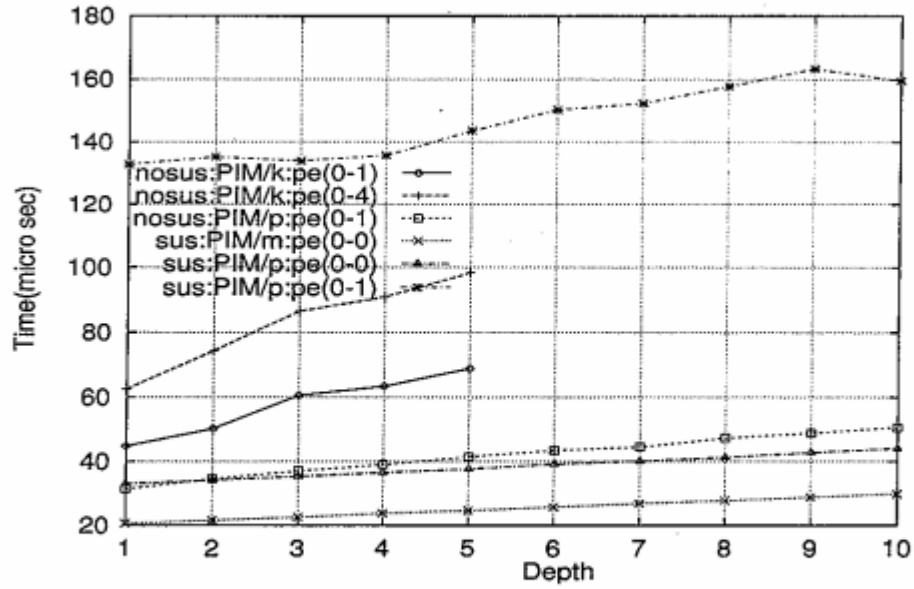
Performance Measurement

- Communication Cost Measurement
- Cost Dependency on Transferred Data Structure
  - Data Width
  - Data Depth

Inter-cluster Comm. Cost vs. Data Depth



Inner-cluster Comm. Cost vs. Data Depth



Conclusion

- KLIC has almost the same performance as PIM.
- Suspension processing in PIM is much faster.
- Random load distribution results in shorter execution threads.
- PIM/p handles frequent distribution better.
- KLIC has larger communication overheads.