OSCAR Multigrain Parallelizing Compiler for Multi-core Architectures

笠原博徳
Hironori Kasahara
Professor, Department of Computer Science
Director, Advanced Chip-Multiprocessor Research Institute
早稲田大学
Waseda University
http://www.kasahara.cs.waseda.ac.jp

Hironori Kasahara

＜Personal History＞

B.S. (1980,Waseda), M.S.(1982,Waseda), Ph.D.(1985,EE, Waseda). Res.Assoc. (1983,Waseda), Special Research Fellow JSPS (1985) ,Visiting Scholar (1985.Univ.California at Berkeley), . Assist. Prof. (1986.Waseda), Assoc. Prof.(1988,Waseda), Visiting Research Scholar(1989-1990. Center for Supercomputing R&D, Univ.of Illinois at Urbana-Champaign), Prof.(1997-,Dept. CS, Waseda). , IFAC World Congress Young Author Prize (1987), IPSJ Sakai Memorial Special Award (1997), STARC Industry-Academia Cooperative Research Award (2004)

＜Activities for Societies＞

IPSJ：Sig. Computer Architecture(Chair), Trans of IPSJ Editorial Board (HG Chair), Journal of IPSJ Editorial Board (HWG Chair), 2001 Journal of IPSJ Special Issue on Parallel Processing(Chair of Editorial Board: Guest Editor, JSPP2000 (Program Chair) etc.

ACM：International Conference on Supercomputing(ICS)(Program Committee)
Int'l conf. on Supercomputing (PC, esp. '96 ENIAC 50th Anniversary Co-Prog. Chair).

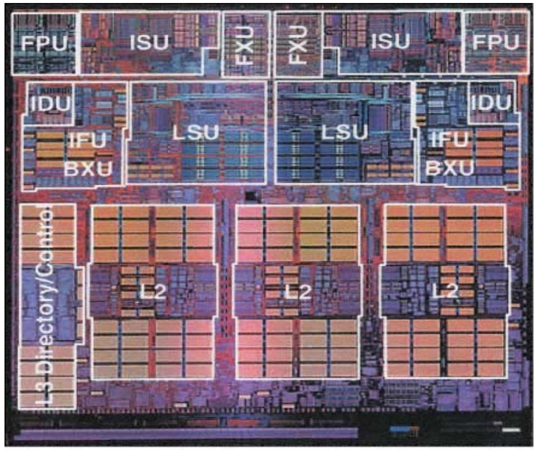
IEEE: Computer Society Japan Chapter Chair, Tokyo Section Board Member,
OTHER: PCs of many conferences on Supercomputing and Parallel Processing.

<Activities for Governments>

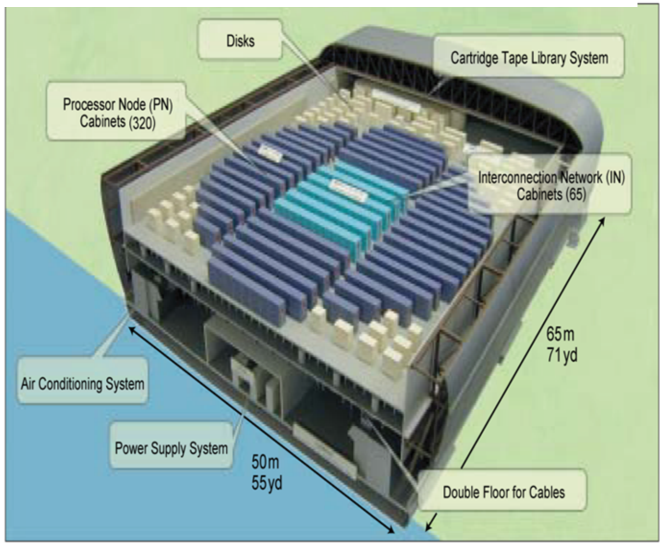
METI：IT Policy Proposal Forum(Architecture/HPC WG Chair),
Super Advanced Electronic Basis Technology Investigation Committee
NEDO:Millennium Project IT21 "Advanced Parallelizing Compiler"(Project Leader),
Computer Strategy WG (Chair).Multicore for Realtime Consumer Electronics Project Leader etc.
MEXT:Earth Simulator project evaluation committee,
JAERI: Research accomplishment evaluation committee, CCSE 1st class invited researcher.
JST: Scientific Research Fund Sub Committee, COINS Steering Committee ,
Precursory Research for Embryonic Science and Technology (Research Area Adviser)
Cabinet Office: CSTP Expert Panel on Basic Policy, Information & Communication Field Promotion Strategy , R&D Infrastructure WG, Software & Security WG

＜Papers＞ 151 Papers with Review, 20 Papers for Symposium with Review, 105 Technicar Reports, 154 Papers for Annual Convention, 49 Invited Talks, 74 Articles in Newspaper & Web, etc.

Multi-core Everywhere



IBM First Chip Multiprocessor (CMP) Power 4 processor
2 cores on a chip



Earth simulaor : 5120 vector processors Multiprocessor Supercomputer

Multi-core from embedded to supercomputers

- **Consumer Electronics (Embedded)**
  - **Mobile Phone, Game, Digital TV, Car Navi, DVD, Camera**
  - **IBM/ Sony/ Toshiba Cell, Fuijtsu FR1000, NEC/ARMMPCore&MP211, Panasonic Uniphier, Renesas SH multi-core(RP1)**
- **PCs, Servers**
  - **Intel Dual-Core Xeon, Core 2 Duo, Montecito**
  - **AMD Quad and Dual-Core Opteron**
- **WSs, Deskside & Highend Servers**
  - **IBM Power4,5,5+, pSeries690(32way), p5 550Q(8 way) ,**
  - **Sun N iagara(SparcT1,T2), SGI ALTIX350,**
- **Supercomputers**
  - **Earth Simulator:40TFLOPS, 2002, 5120 vector proc.**
  - **IBM Blue Gene/L: 360TFLOPS, 2005,**
    Low power CMP based 128K processor chips

High quality application software, Productivity, Cost performance, Low power consumption are important

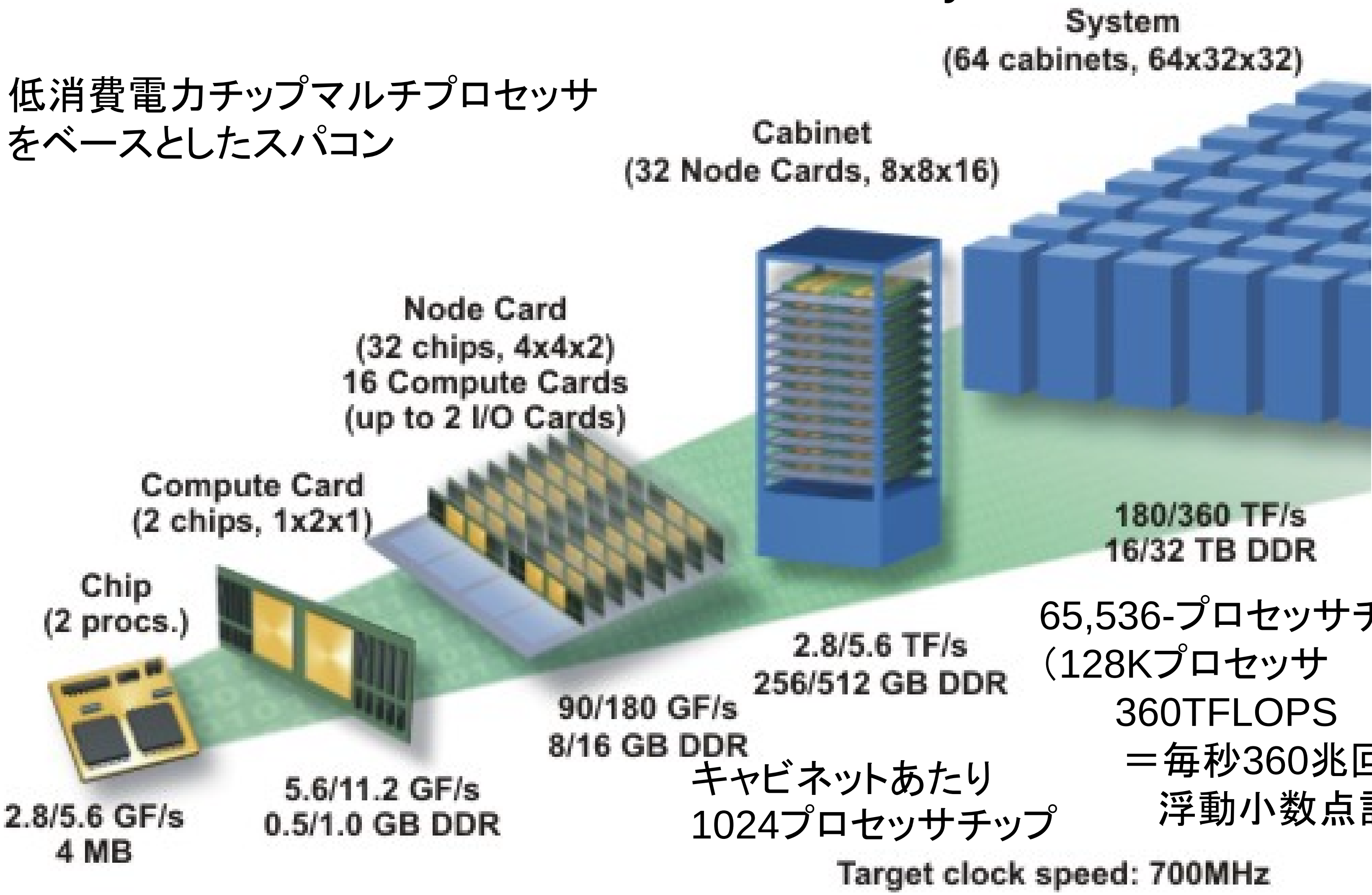
Ex, Mobile phones, Games

Compiler cooperated multi-core processors are promising to realize the above futures

IBM BlueGene/L

Lawrence Livermore National Laboratory2005/

低消費電力チップマルチプロセッサ
をベースとしたスパコン



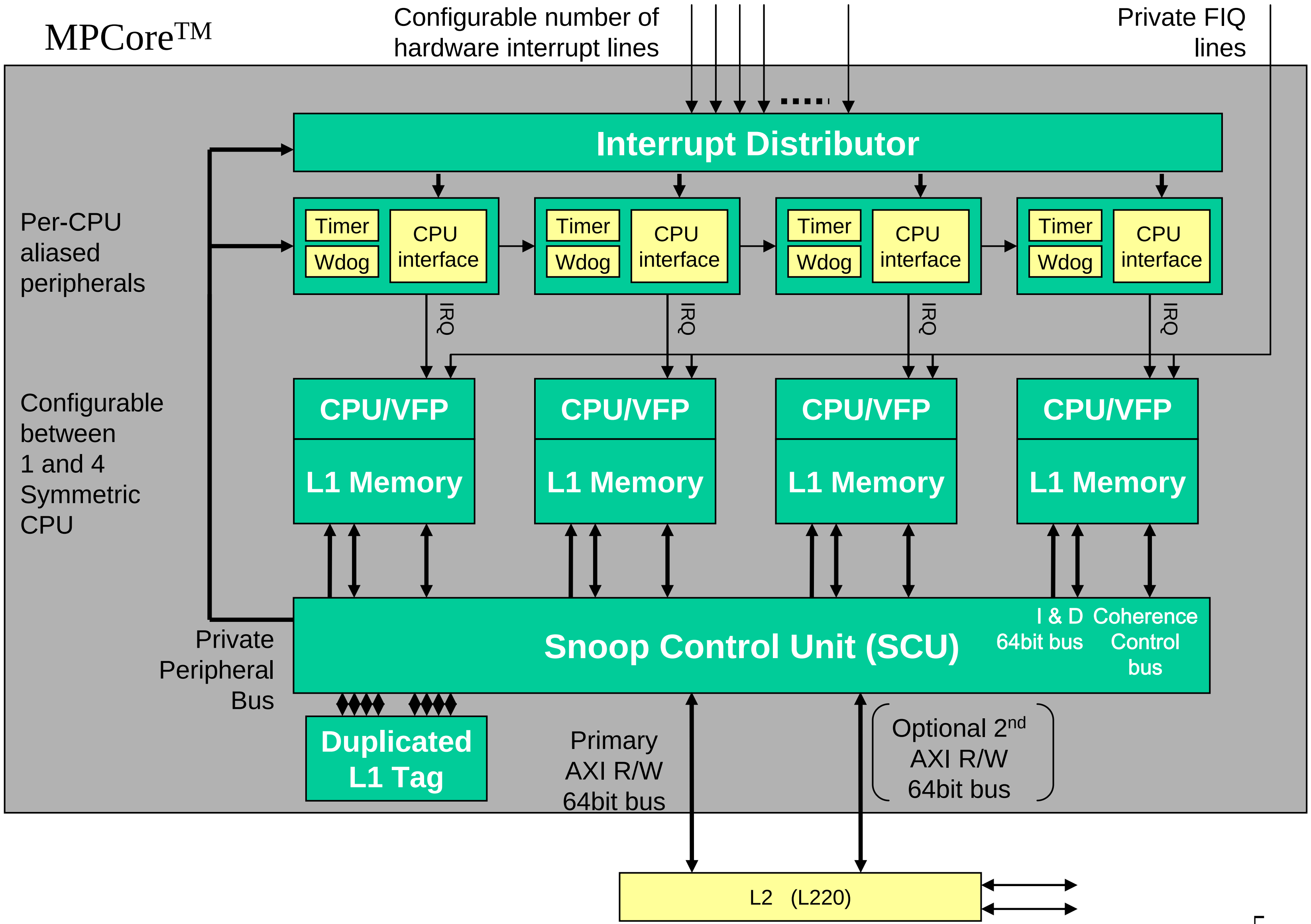
65,536-プロセッサチップ
（128Kプロセッサ
360TFLOPS
＝毎秒360兆回の
浮動小数点計算）

キャビネットあたり
1024プロセッサチップ

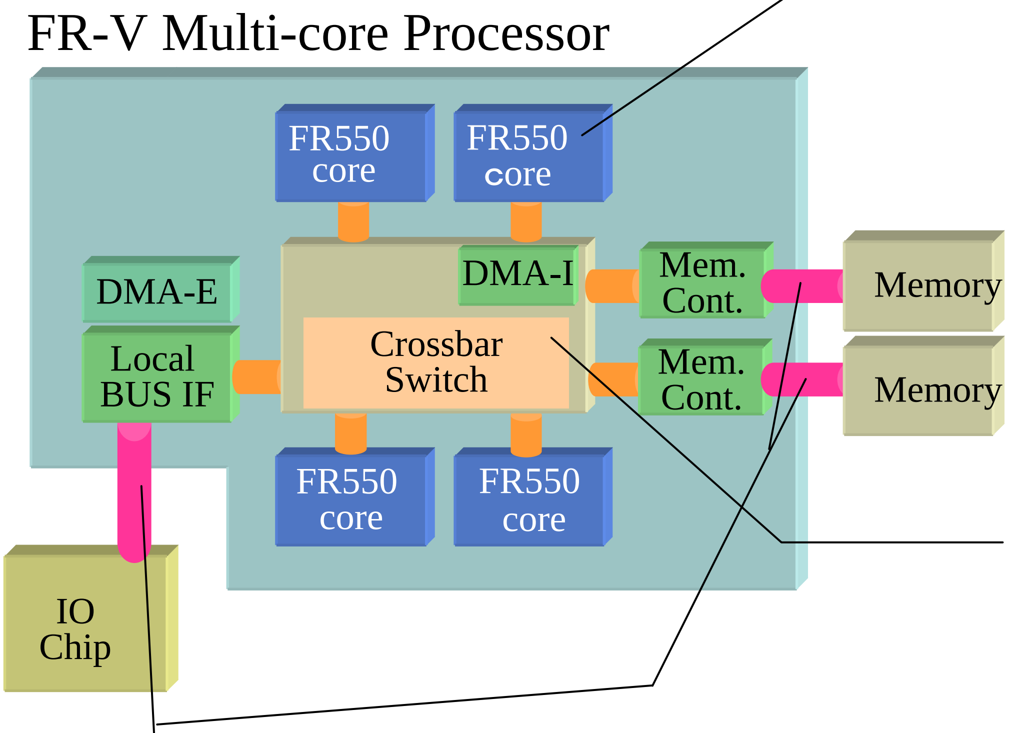
1プロセッサチップ上に2プロセッサ集積

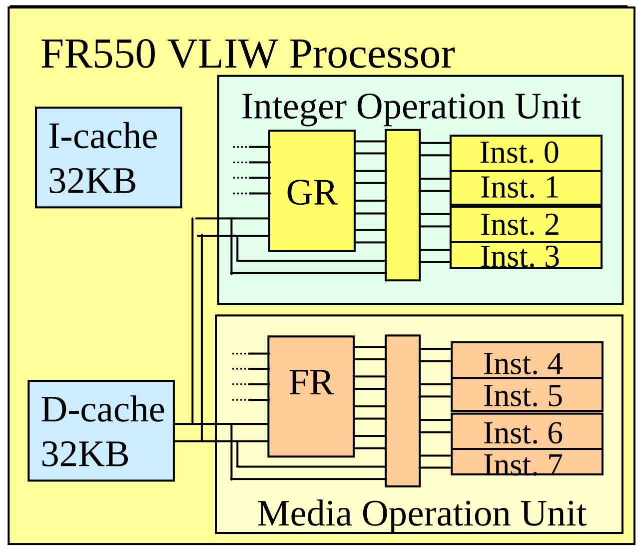
MPCore™

ARM and NEC Collaboration



Fujitsu FR-1000Multicore Processor



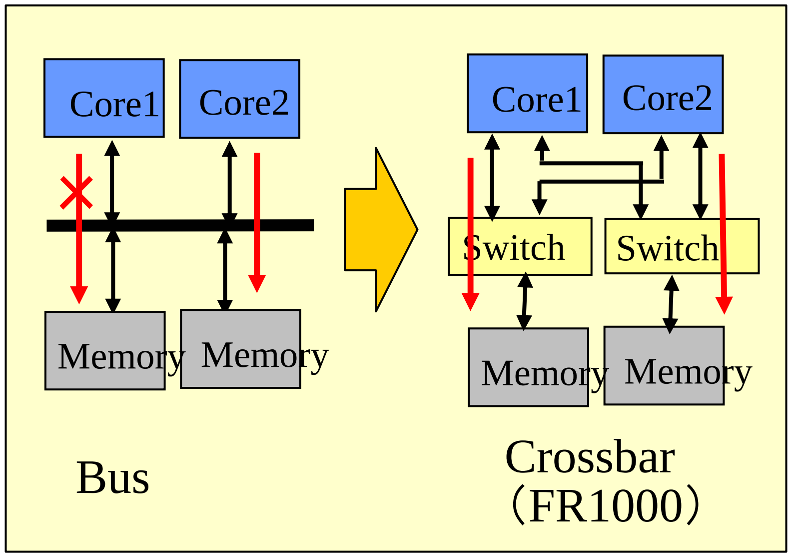


Fast I/O Bus

•Memory Bus: 64bit x 2ch / 266MHz

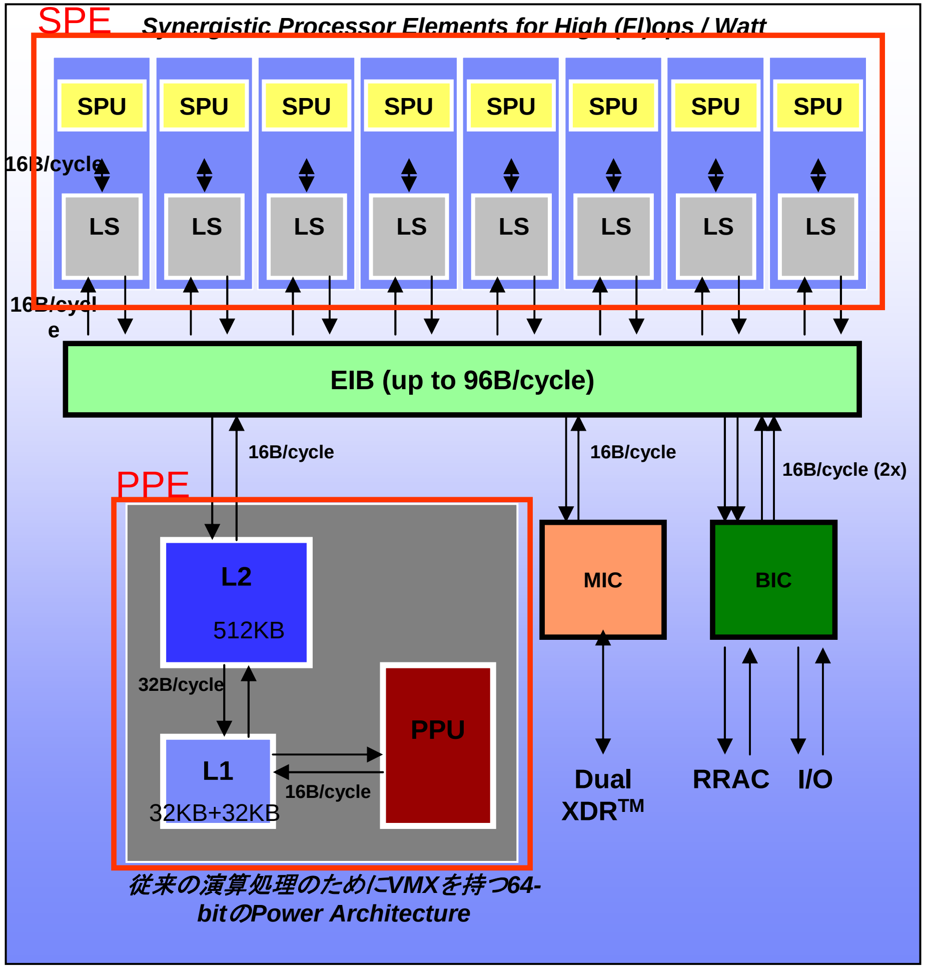
•System Bus: 64bit / 178MHz

(周波数は現FR-Vの2倍）

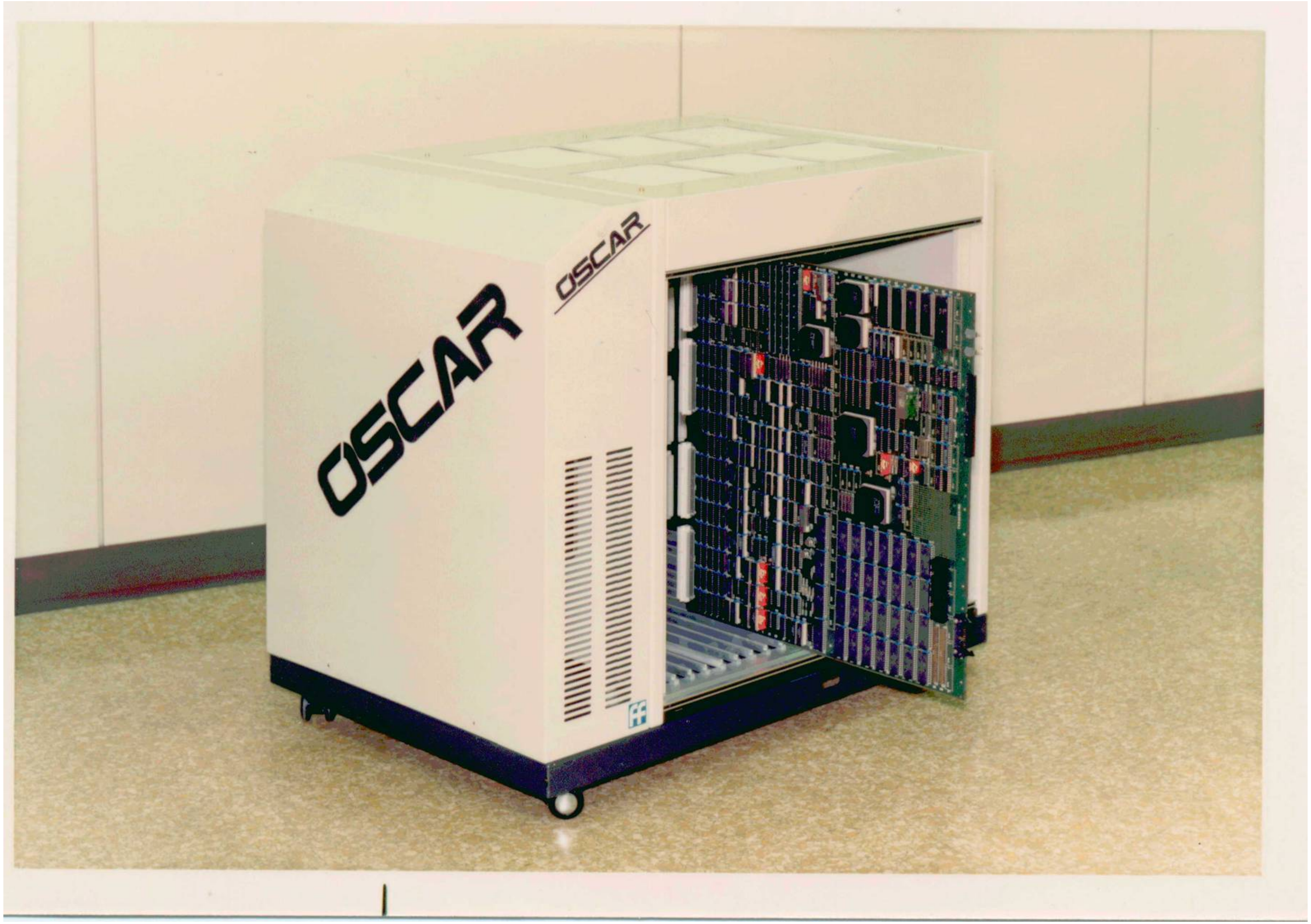


CELL Processor Overview

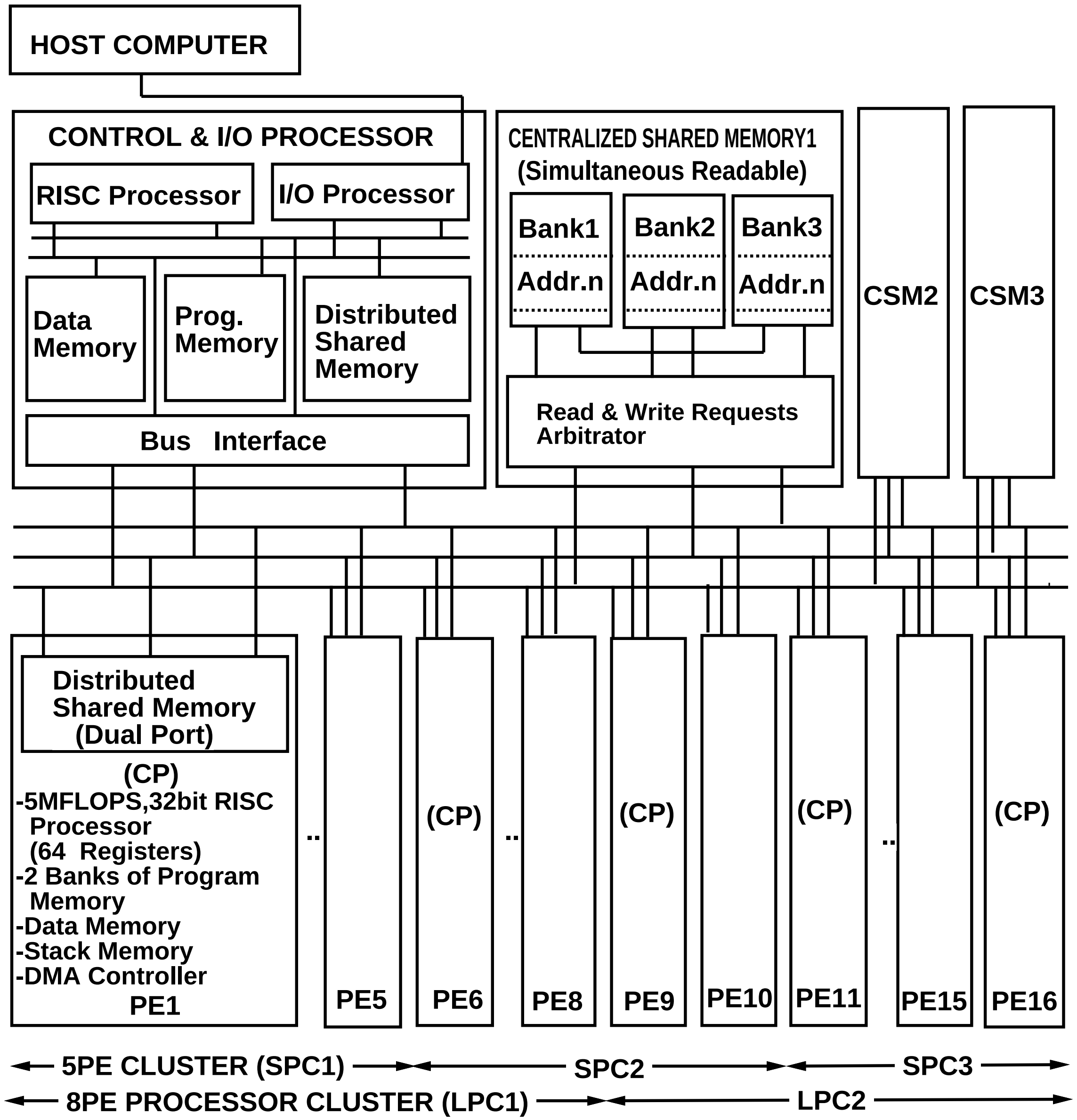
- **Power Processor Element (PPE)**
  - PowerCore processes OS and Control tasks
  - 2-way Multi-threaded
- **Synergistic Processor Element (SPE)**
  - 8 SPE offers high performance
  - Dual issue RISC Architecture
  - 128bit SIMD(16-way)
  - 128 x 128bit General Registers
  - 256KB Local Store
  - DedicatedDMA engines



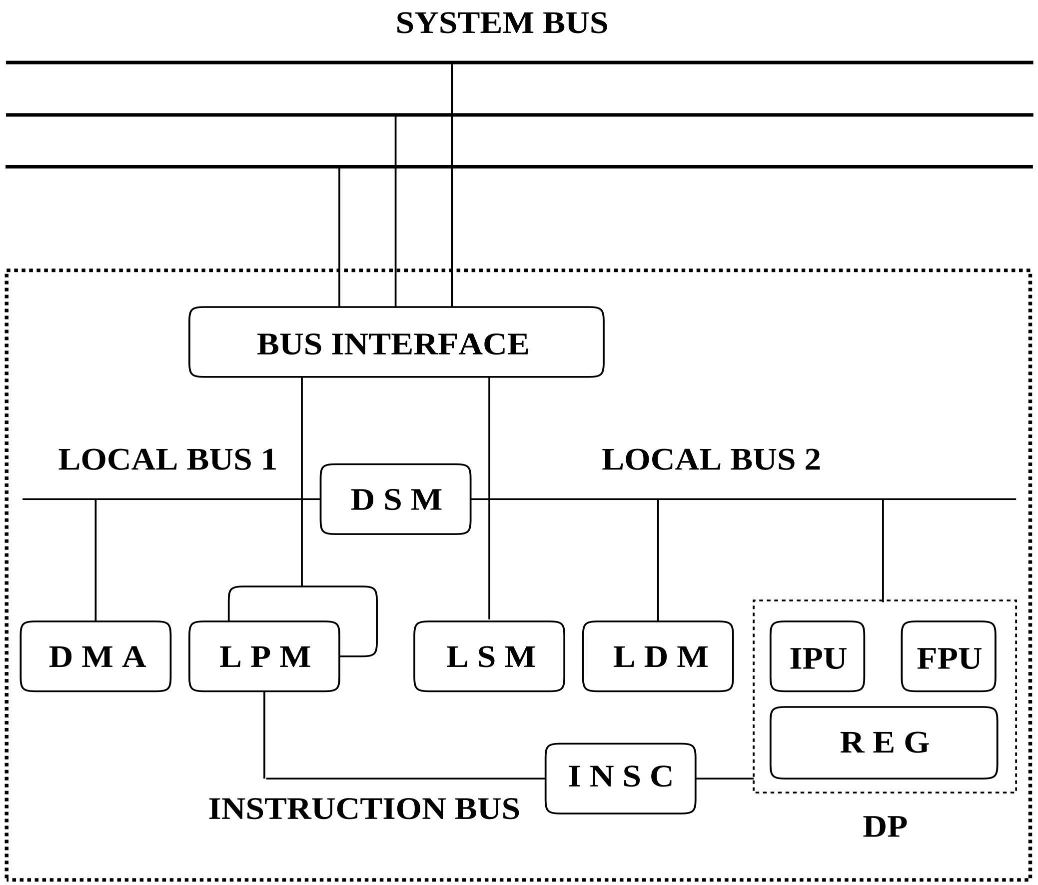
1987 OSCAR(Optimally Scheduled Advanced Multiprocessor)



OSCAR(Optimally Scheduled Advanced Multiprocessor)

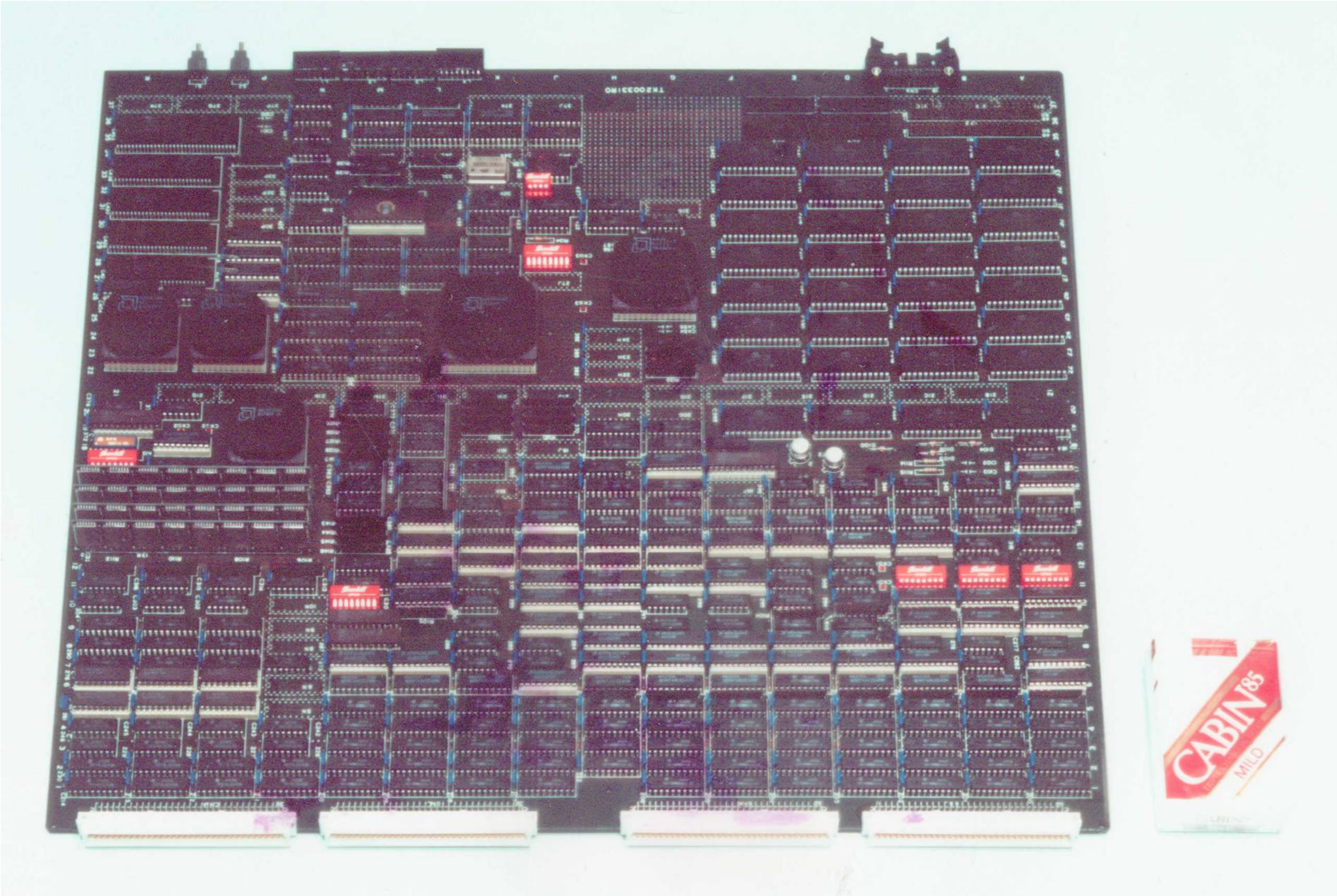


OSCAR PE (Processor Element)

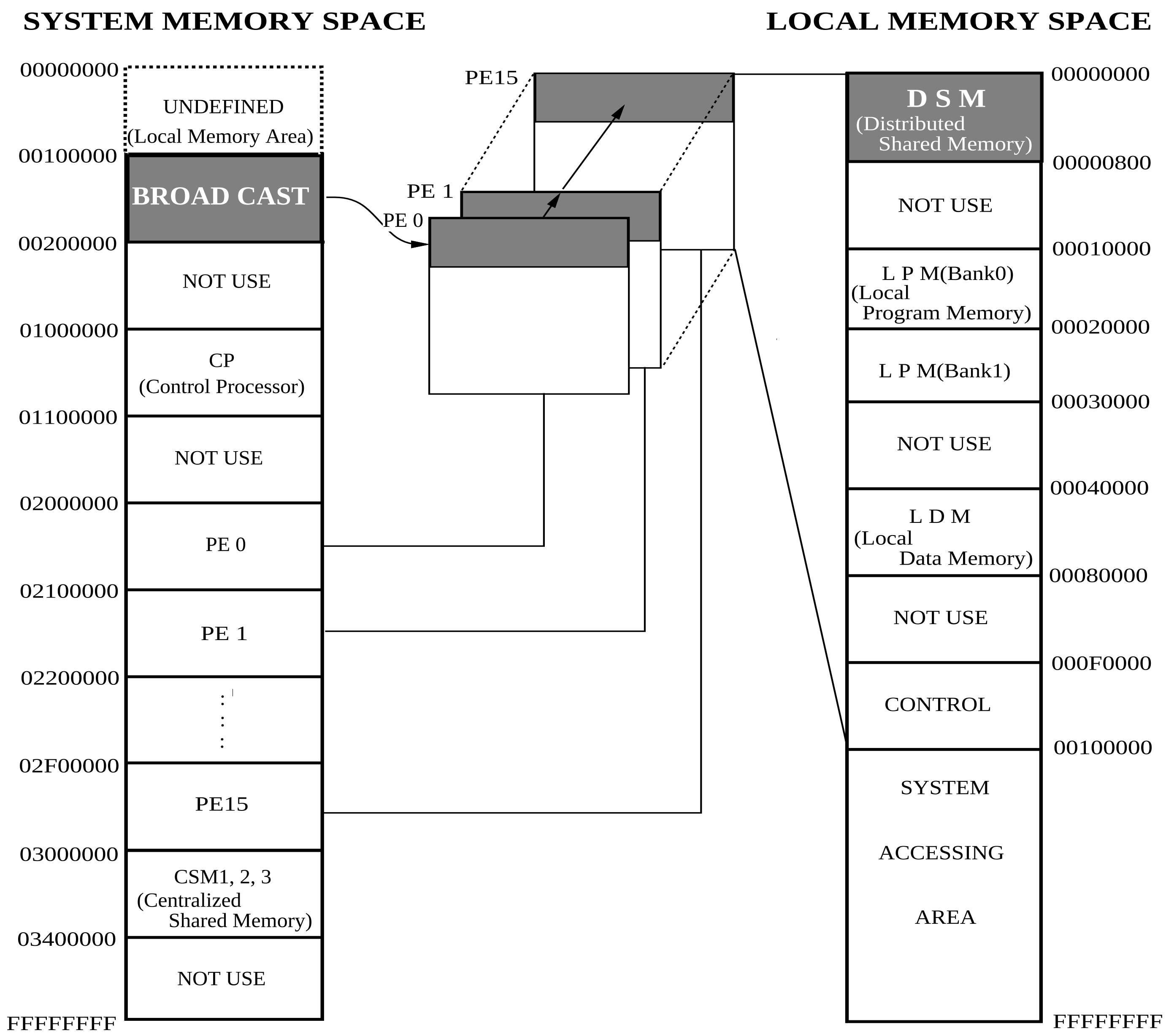


D M A : DMA CONTROLLER
L P M : LOCAL PROGRAM MEMORY (128KW * 2BANK)
I N S C : INSTRUCTION CONTROL UNIT
D S M : DISTRIBUTED SHARED MEMORY (2KW)
L S M : LOCAL STACK MEMORY (4KW)
L D M : LOCAL DATA MEMORY (256KW)
D P : DATA PATH
I P U : INTEGER PROCESSING UNIT
F P U : FLOATING PROCESSING UNIT
R E G : REGISTER FILE (64 REGISTERS)

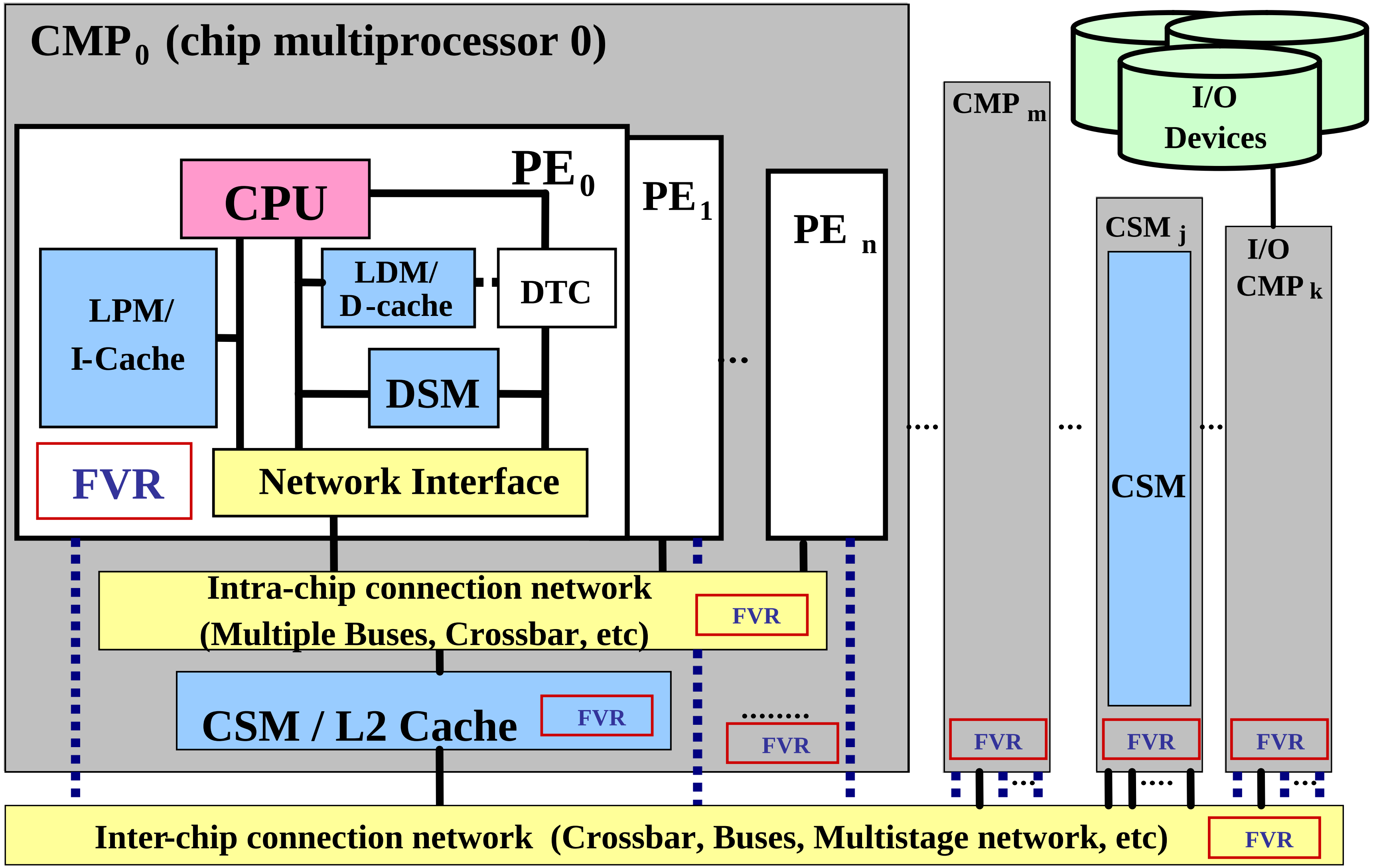
1987 OSCAR PE Board



OSCAR Memory Space



OSCAR Multi-Core Architecture



CSM: central shared mem.

DSM: distributed shared mem.

DTC: Data Transfer Controller

LDM : local data mem.

LPM : local program mem.

FVR: frequency / voltage control register

National Project by METI/NEDO

Multi-core for Real-time Consumer Electronics

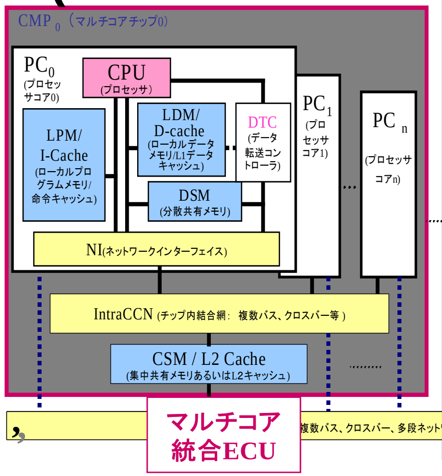
<Goal> **R&D of compiler cooperative multi-core processor technology for consumer electronics like Mobile phones, Games, DVD, Digital TV, Car navigation systems.**

<Period> **From July 2005 to March 2008**

<Features>
- **Good cost performance**
- **Short hardware and software development periods**
- **Low power consumption**
- **Scalable performance improvement with the advancement of semiconductor**
- **Use of the same parallelizing compiler for multi-cores from different vendors using newly developed API**

API：**Application Programming Interface**

経済産業省 / NEDOリアルタイム情報家電用マルチコア (2005.7～2008.3)**



新マルチコアプロセッサ
- 高性能
- 低消費電力
- 短HW/SW開発期間
- 各チップ間でアプリケーション共用可
- 高信頼性
- 半導体集積度と共に性能向上

開発マルチコアチップは情報家電へ

**日立,富士通,ルネサス,東芝,松下,NEC

Needs for Parallizing Compiler Cooperative Chip Multiprocessor

- **Limitation of ILP (Instruction Level Parallelism) in microprocessors**
- **Limitation of loop level parallelism in high end multiprocessor systems**
  - **Mature data dependence analysis & restructuring techniques by loop parallelizing compilers**
- **Needs for scalable architectures with increasing transistors integrated on a chip with better cost performance, low power consumption, short development period and high application soft productivity (ease of use).**

➢ **Multigrain parallelizing compiler cooperative Chip Multiprocessor (CMP)**

Vectorization

Loop Distribution and Vectorization for Innermost Loop

do I = 2,N

C(I)=A(I)+B(I) :S1

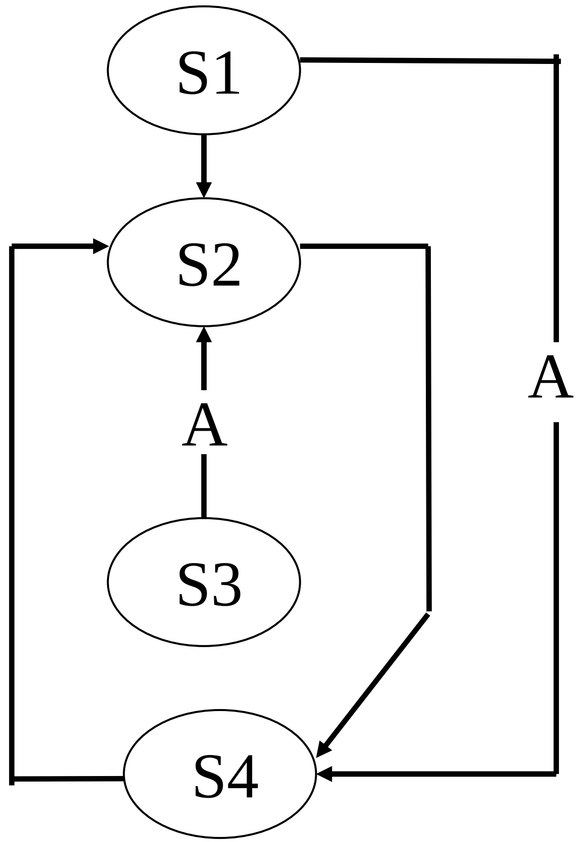
D(I)=C(I)+A(I-1) :S2

E(I)=D(I+1)+X :S3

A(I)=D(I) ∗ B(I) :S4

end do

(a) A Do loop



(b) Data dependence graph for (a)

C(2:N)=A(2:N)+B(2:N) :S1

E(2:N)=D(3:N+1)+X :S3

do I=2,N

D(I)=C(I)+A(I-1) :S2

A(I)=D(I)*B(I) :S4

end do

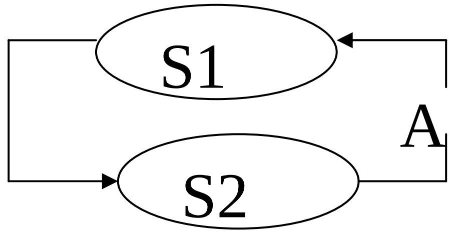
(c) Vectorized code

Scalar Expansion

An Example of Loop Restructuring

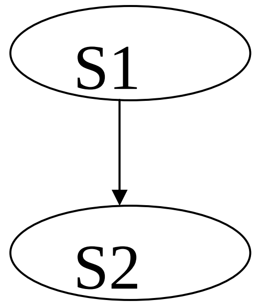
```
do I=1,N
   T=A(I)*B(I)        :S1
   C(I)=T+B(I)        :S2
end do
```

(a)




```
do I=1,N
   TEMP(I)=A(I)*B(I) :S1
   C(I)=TEMP(I)+B(I) :S2
end do
T=TEMP(N)
```

(b)




```
TEMP(1:N)=A(1:N)*B(1:N)
C(I)=TEMP(1:N)+B(1:N)
T=TEMP(N)
```

(c)

Doall Loop

with Scalar Privatization

```
real a(n),c(n)
do i=1,n
    x=a(i)*2.0
   c(i)=x+1.0
end do
```

```
real a(n),c(n)
doall i =1,n
    real x
    x=a(i)*2.0
    c(i)=x+1.0
end doall
```

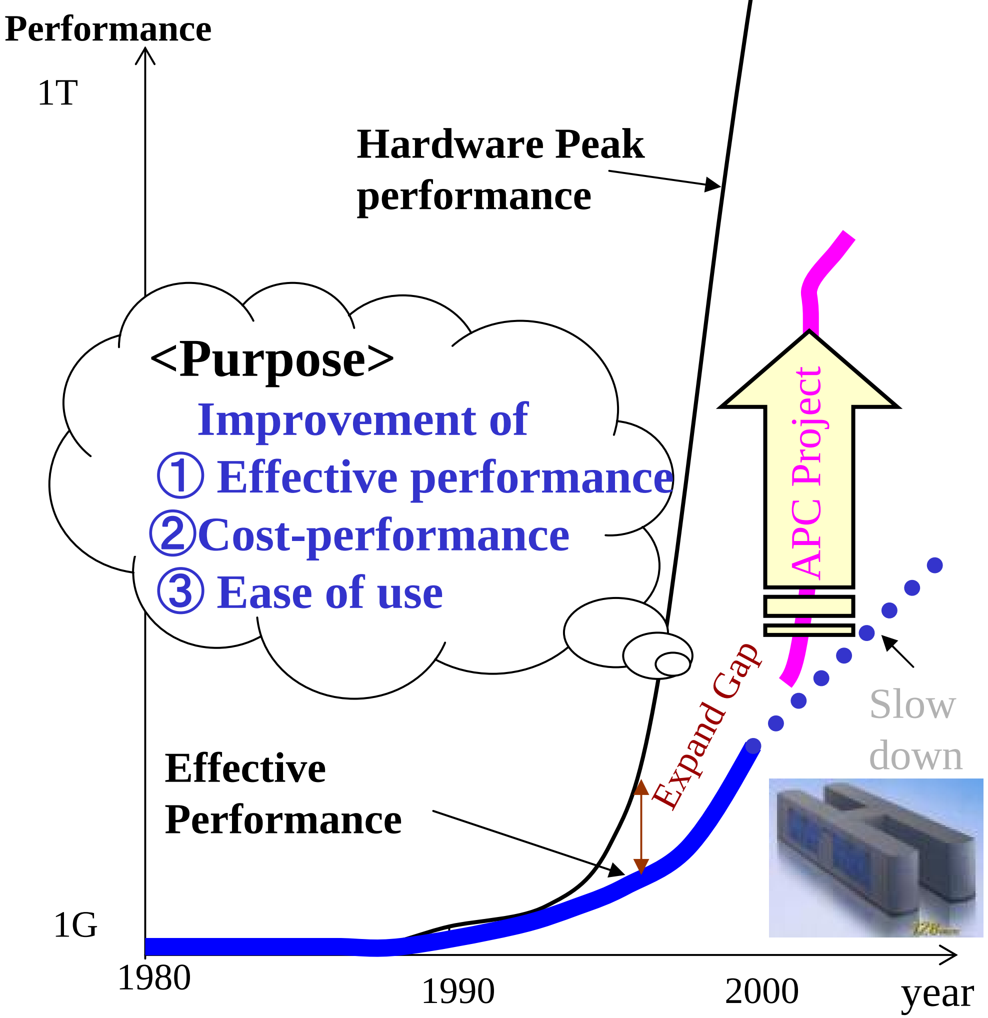
A loop can be parallelized **Doall with privatization**

OSCAR Parallelizing Compiler

- **Improve effective performance, cost-performance and productivity and reduce consumed power**
  - **Multigrain Parallelization**
    - **Exploitation of parallelism from the whole program by use of coarse-grain parallelism among loops and subroutines, near fine grain parallelism among statements in addition to loop parallelism**
  - **Data Localization**
    - **Automatic data distribution for distributed shared memory, cache and local memory on multiprocessor systems.**
  - **Data Transfer Overlapping**
    - **Data transfer overhead hiding by overlapping task execution and data transfer using DMA or data pre-fetching**
  - **Power Reduction**
    - **Reduction of consumed power by compiler control of frequency, voltage and power shut down with hardware supports.**

METI/NEDO Advanced Parallelizing Compiler Technology Project

Millenium Project IT21 2000.9.8 –2003.3.31
Waseda Univ., Fujitsu, Hitachi, AIST



Theoretical maximum performance vs.
Effective performance of HPC

Background and Problems

①Adoption of parallel processing as a core technology on PC to HPC
② Increase of importance of software on IT
③ Need for improvement of cost-performance and usability

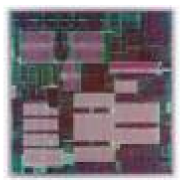
Contents of Research and Development

① R & D of advanced parallelizing compiler
Multigrain, Data localization, Overhead hiding
② R & D of Performance evaluation technology for parallelizing compilers

Goal: Double the effective performance

Ripple Effect

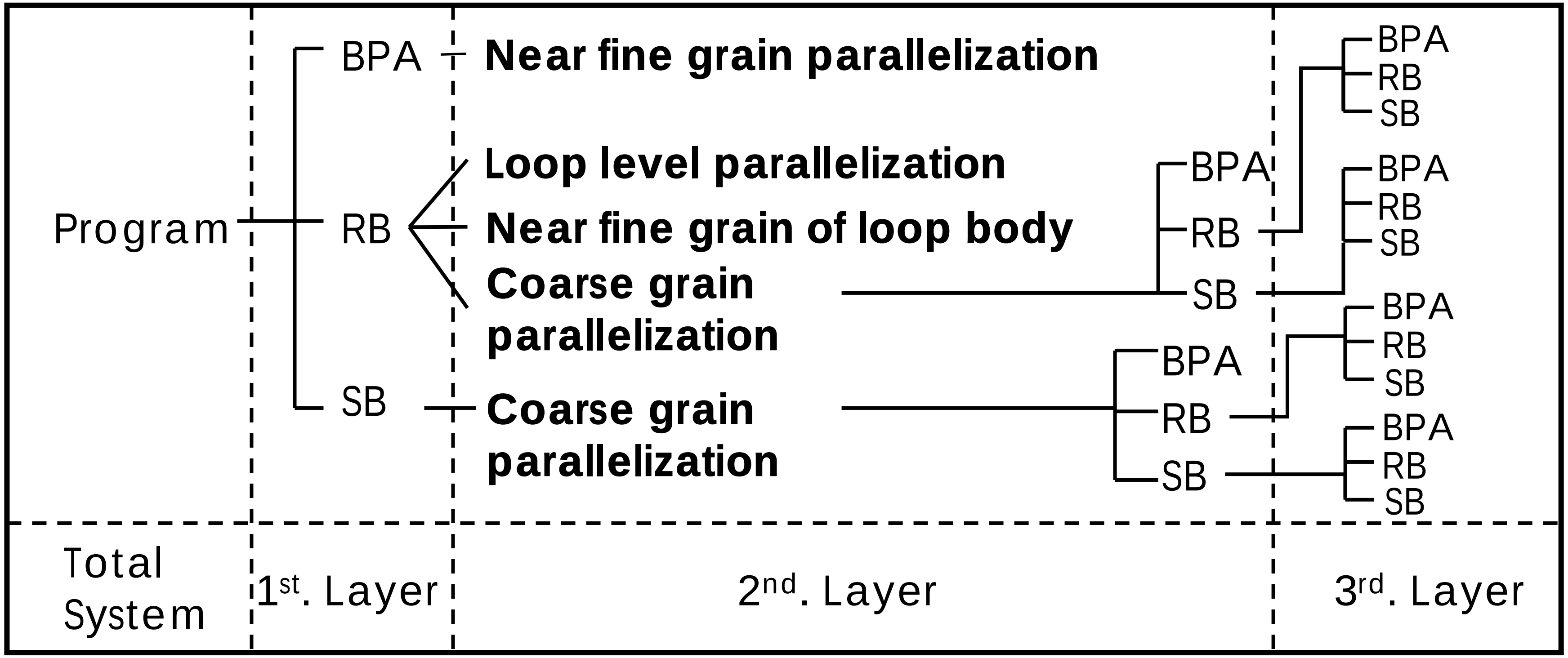
① Development of competitive next generation PC and HPC
② Putting the innovative automatic parallelizing compiler technology to practical use
③ Development and market acquisition of future single-chip multiprocessors
④ Boosting R&D in the following many fields:
IT, Bio-tech., Device, Earth environment, Next-generation VLSI design, Financial engineering, Weather forecast, New clean energy, Space development, Automobile, Electric Commerce, etc



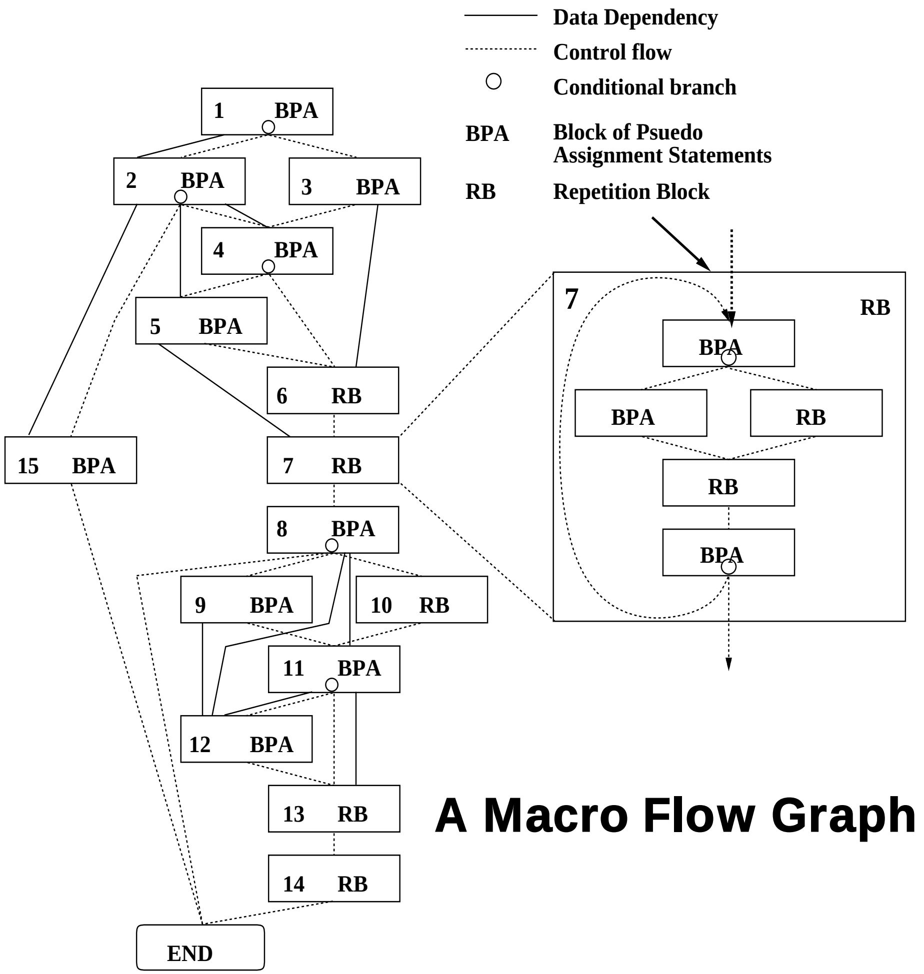
Generation of Coarse Grain Tasks

Macro-tasks (MTs)

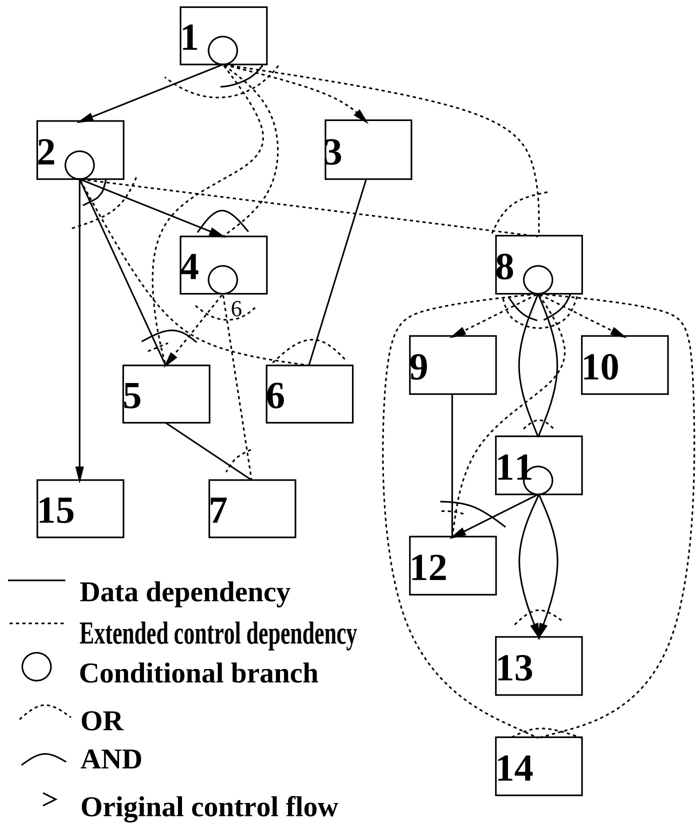
- Block of Pseudo Assignments (**BPA**): Basic Block (BB)
- Repetition Block (**RB**) : outermost natural loop
- Subroutine Block (**SB**): subroutine



Earliest Executable Condition Analysis for coarse grain tasks (Macro-tasks)



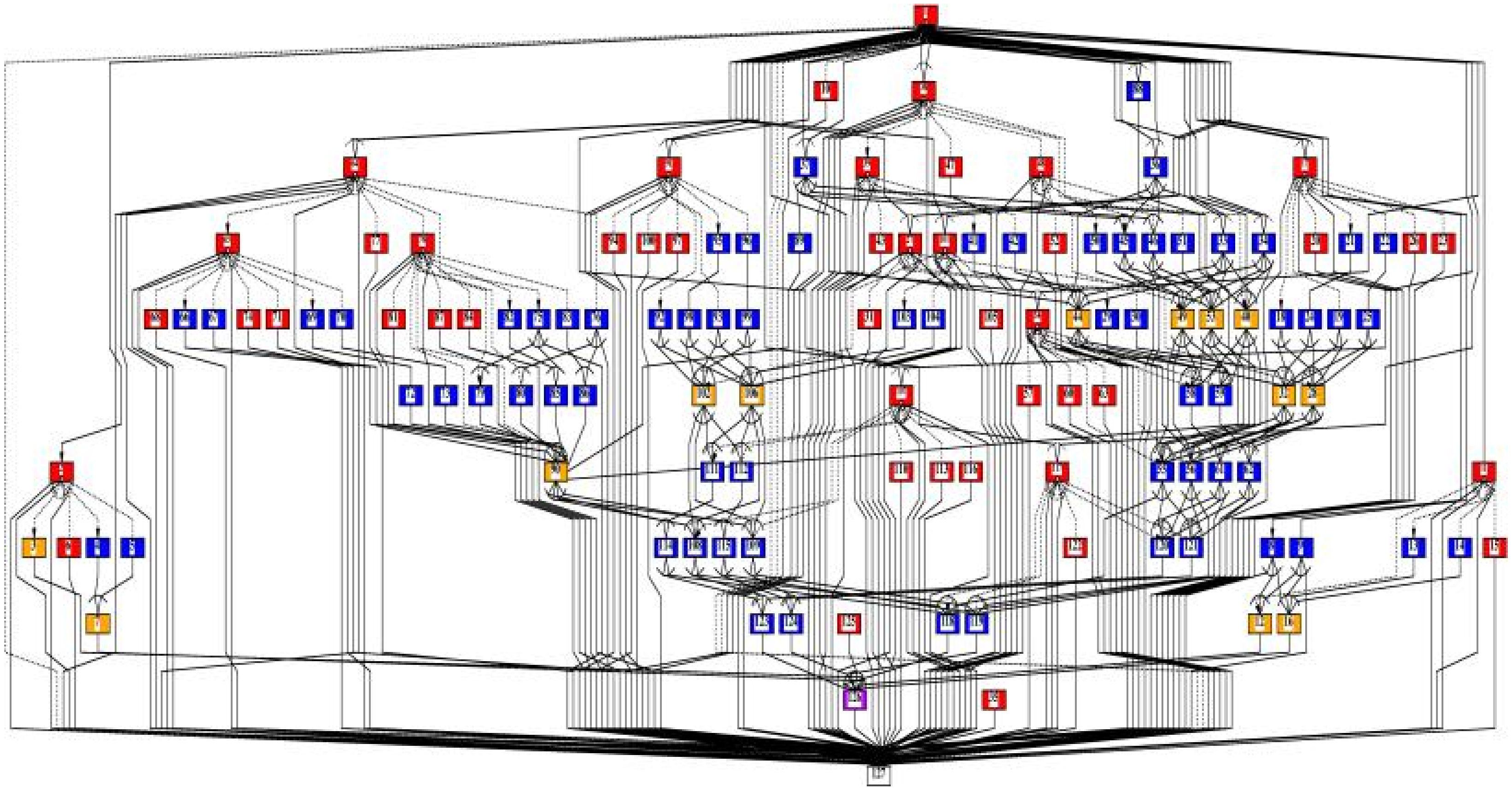
A Macro Flow Graph



A Macro Task Graph

MTG of Su2cor-LOOPS-DO400

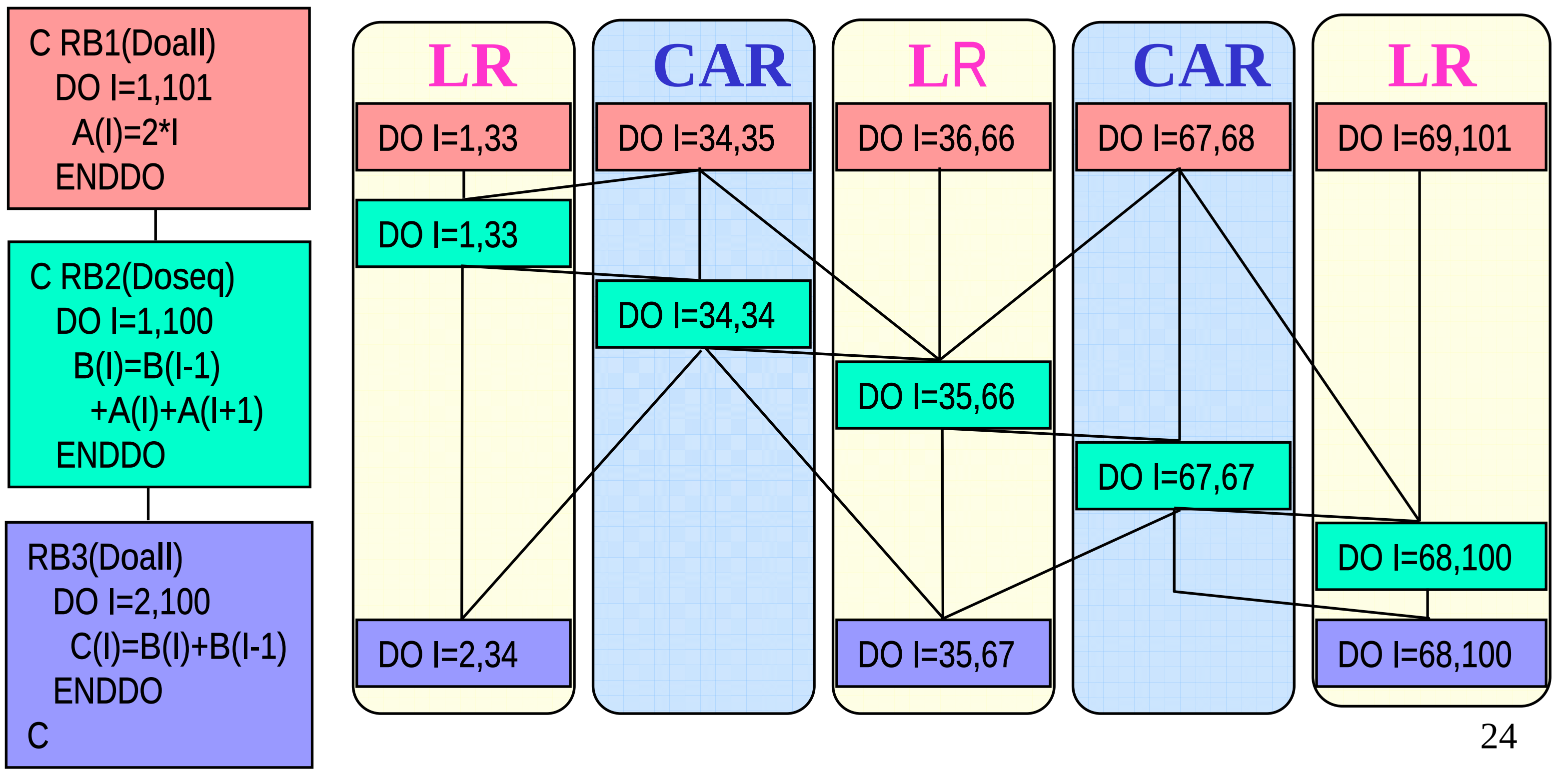
- Coarse grain parallelism PARA_ALD = 4.3



DOALL Sequential LOOP SB BB

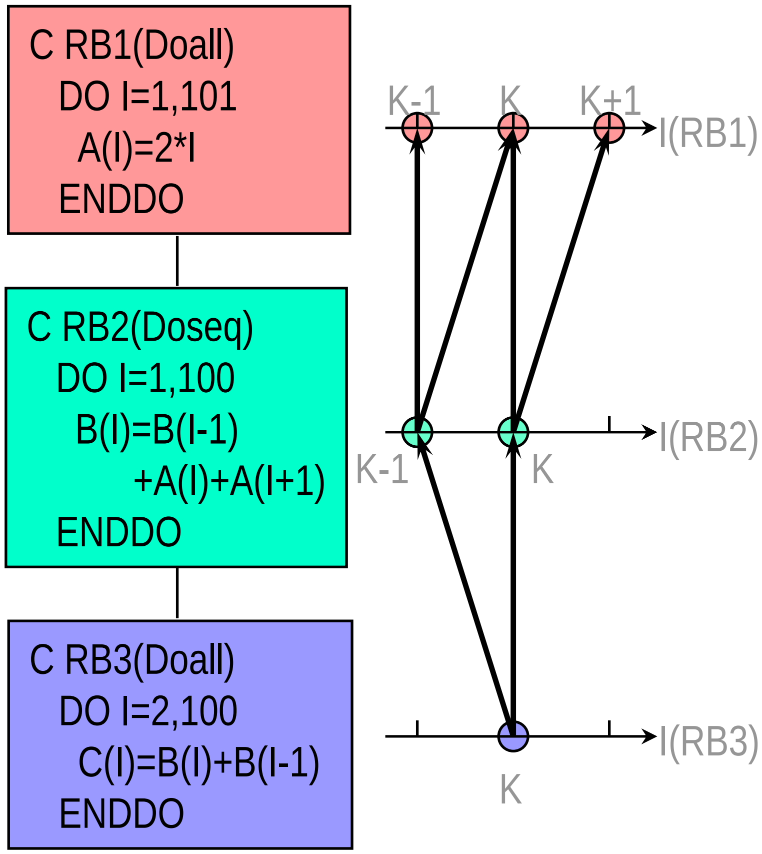
Data-Localization
Loop Aligned Decomposition

- **Decompose multiple loop (Doall and Seq) into CARs and LRs considering inter-loop data dependence.**
  - **Most data in LR can be passed through LM.**
  - **LR: Localizable Region, CAR: Commonly Accessed Region**



Inter-loop data dependence analysis in TLG

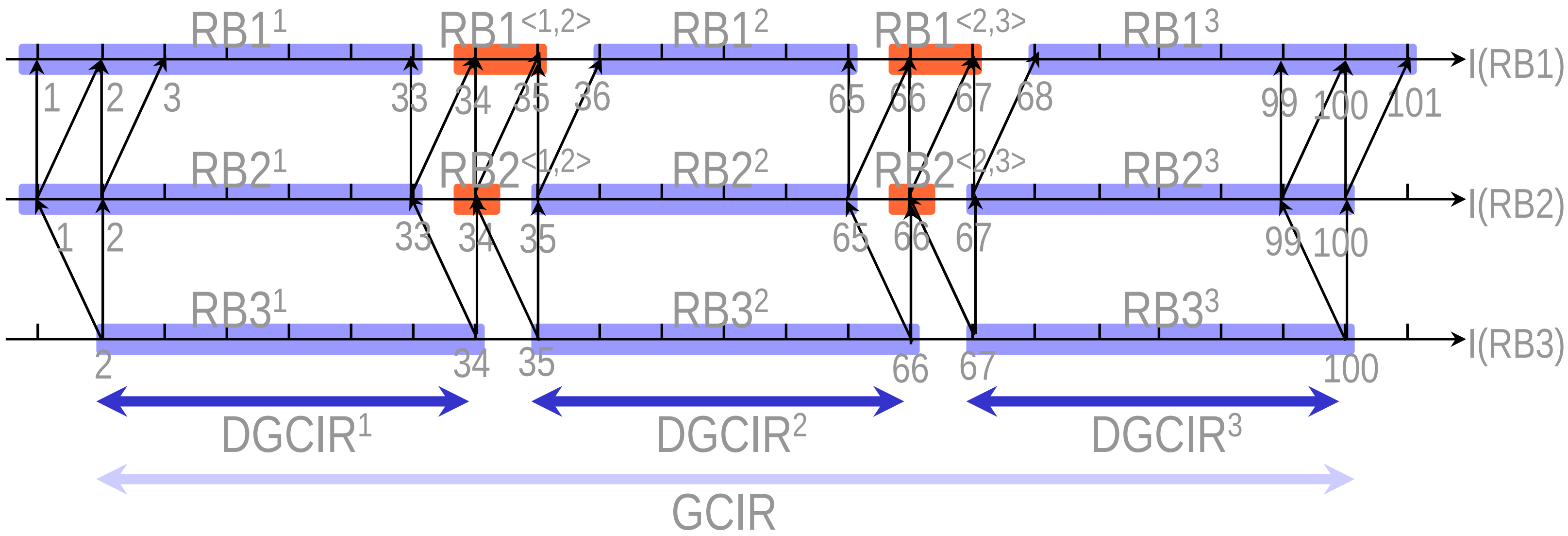
- Define exit-RB in TLG as Standard-Loop
- Find iterations on which a iteration of Standard-Loop is data dependent
  - e.g. K_{th} of RB3 is data-dep on $K\text{-}1_{th}, K_{th}$ of RB2, on $K\text{-}1_{th}, K_{th}, K+1_{th}$ of RB1



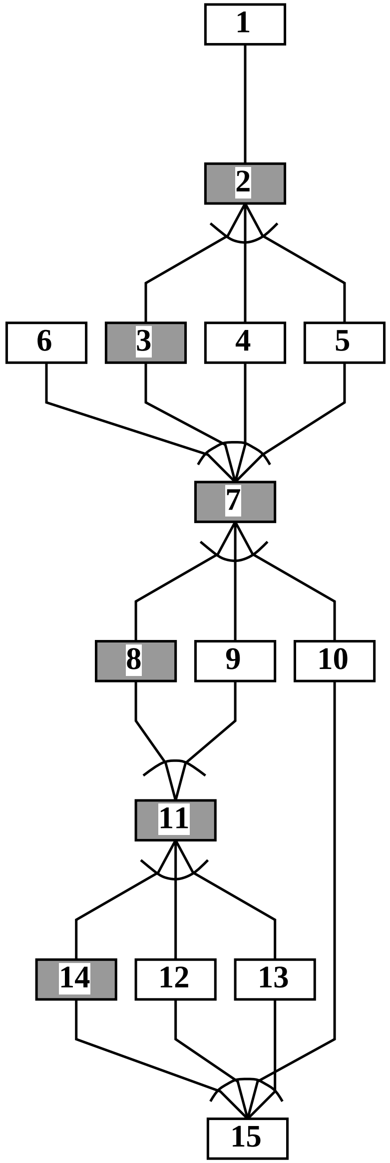
Example of TLG

Decomposition of RBs in TLG

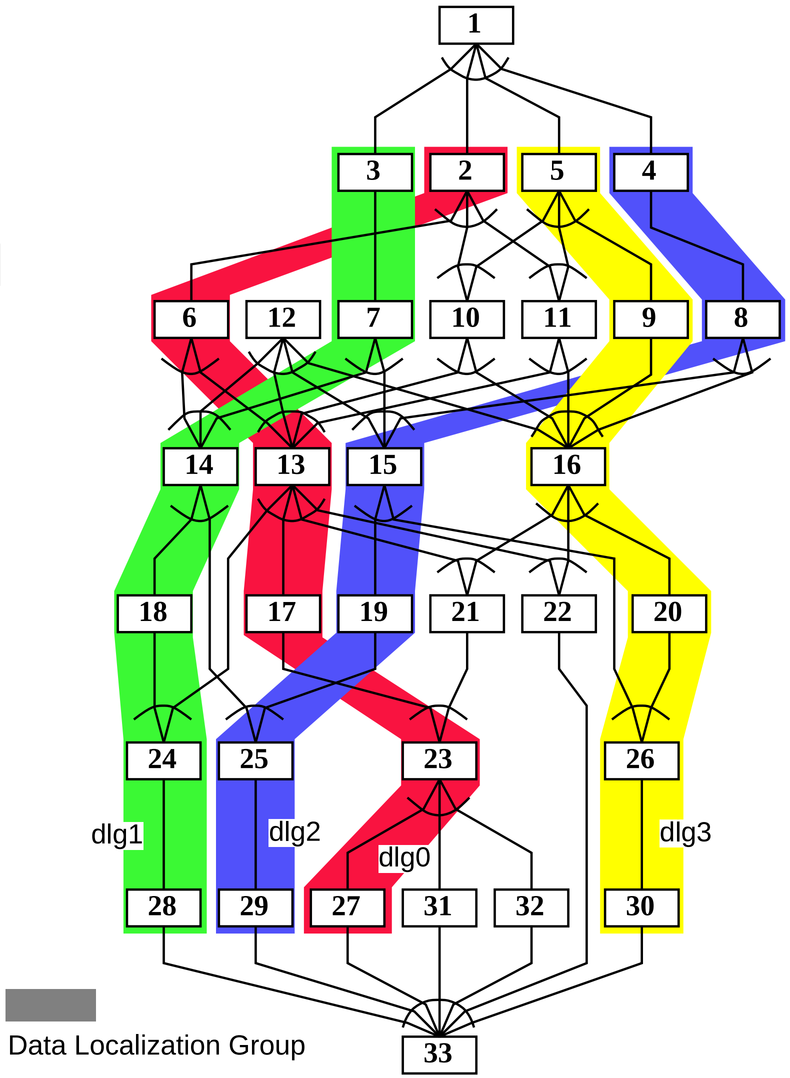
- Decompose GCIR into $DGCIR^p (1 \leqq p \leqq n)$
  - n: (multiple) num of PCs, DGCIR: Decomposed GCIR
- Generate CAR on which $DGCIR^p$&$DGCIR^{p+1}$ are data-dep.
- Generate LR on which $DGCIR^p$ is data-dep.



Data Localization



MTG



MTG after Division

PE0	PE1
12	1
2	3
6	7
4	14
8	18
15	5
19	9
25	11
29	10
13	16
17	20
22	26
21	30
23	24
27	28
	32
	31

A schedule for two processors

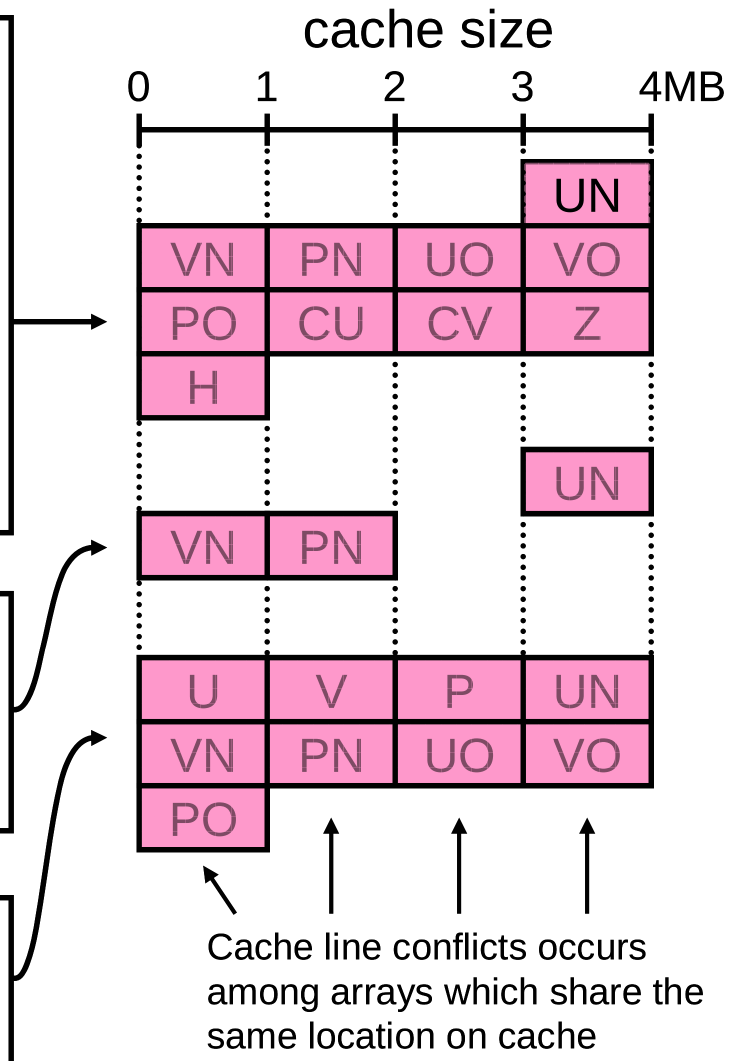
An Example of Data Localization for Spec95 Swim

```
      DO 200 J=1,N
      DO 200 I=1,M
         UNEW(I+1,J) = UOLD(I+1,J)+
     1   TDTS8*(Z(I+1,J+1)+Z(I+1,J))*(CV(I+1,J+1)+CV(I,J+1)+CV(I,J)
     2      +CV(I+1,J))-TDTSDX*(H(I+1,J)-H(I,J))
         VNEW(I,J+1) = VOLD(I,J+1)-TDTS8*(Z(I+1,J+1)+Z(I,J+1))
     1      *(CU(I+1,J+1)+CU(I,J+1)+CU(I,J)+CU(I+1,J))
     2      -TDTSDY*(H(I,J+1)-H(I,J))
         PNEW(I,J) = POLD(I,J)-TDTSDX*(CU(I+1,J)-CU(I,J))
     1      -TDTSDY*(CV(I,J+1)-CV(I,J))
  200 CONTINUE
```

```
      DO 210 J=1,N
         UNEW(1,J) = UNEW(M+1,J)
         VNEW(M+1,J+1) = VNEW(1,J+1)
         PNEW(M+1,J) = PNEW(1,J)
  210 CONTINUE
```

```
      DO 300 J=1,N
      DO 300 I=1,M
         UOLD(I,J) = U(I,J)+ALPHA*(UNEW(I,J)-2.*U(I,J)+UOLD(I,J))
         VOLD(I,J) = V(I,J)+ALPHA*(VNEW(I,J)-2.*V(I,J)+VOLD(I,J))
         POLD(I,J) = P(I,J)+ALPHA*(PNEW(I,J)-2.*P(I,J)+POLD(I,J))
  300 CONTINUE
```

(a) An example of target loop group for data localization



(b) Image of alignment of arrays on cache accessed by target loops

Data Layout for Removing Line Conflict Misses by Array Dimension Padding

Declaration part of arrays in spec95 swim

before padding

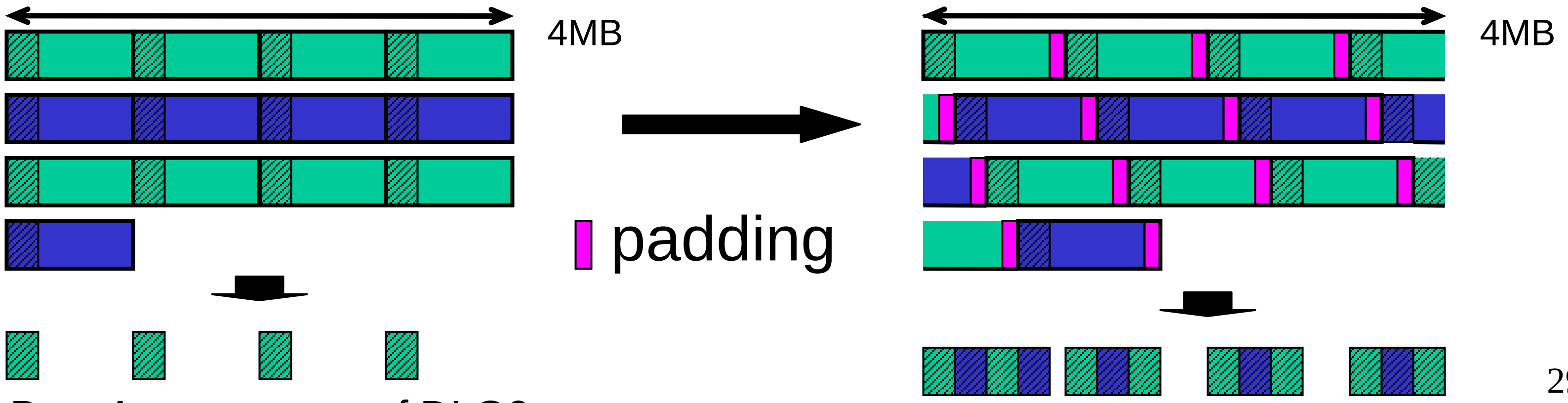
```
PARAMETER (N1=513, N2=513)

COMMON  U(N1,N2), V(N1,N2), P(N1,N2),
*       UNEW(N1,N2), VNEW(N1,N2),
1       PNEW(N1,N2), UOLD(N1,N2),
*       VOLD(N1,N2), POLD(N1,N2),
2       CU(N1,N2), CV(N1,N2),
*       Z(N1,N2), H(N1,N2)
```

after padding

```
PARAMETER (N1=513, N2=544)

COMMON  U(N1,N2), V(N1,N2), P(N1,N2),
*       UNEW(N1,N2), VNEW(N1,N2),
1       PNEW(N1,N2), UOLD(N1,N2),
*       VOLD(N1,N2), POLD(N1,N2),
2       CU(N1,N2), CV(N1,N2),
*       Z(N1,N2), H(N1,N2)
```



Box: Access range of DLG0

APC Compiler Organization

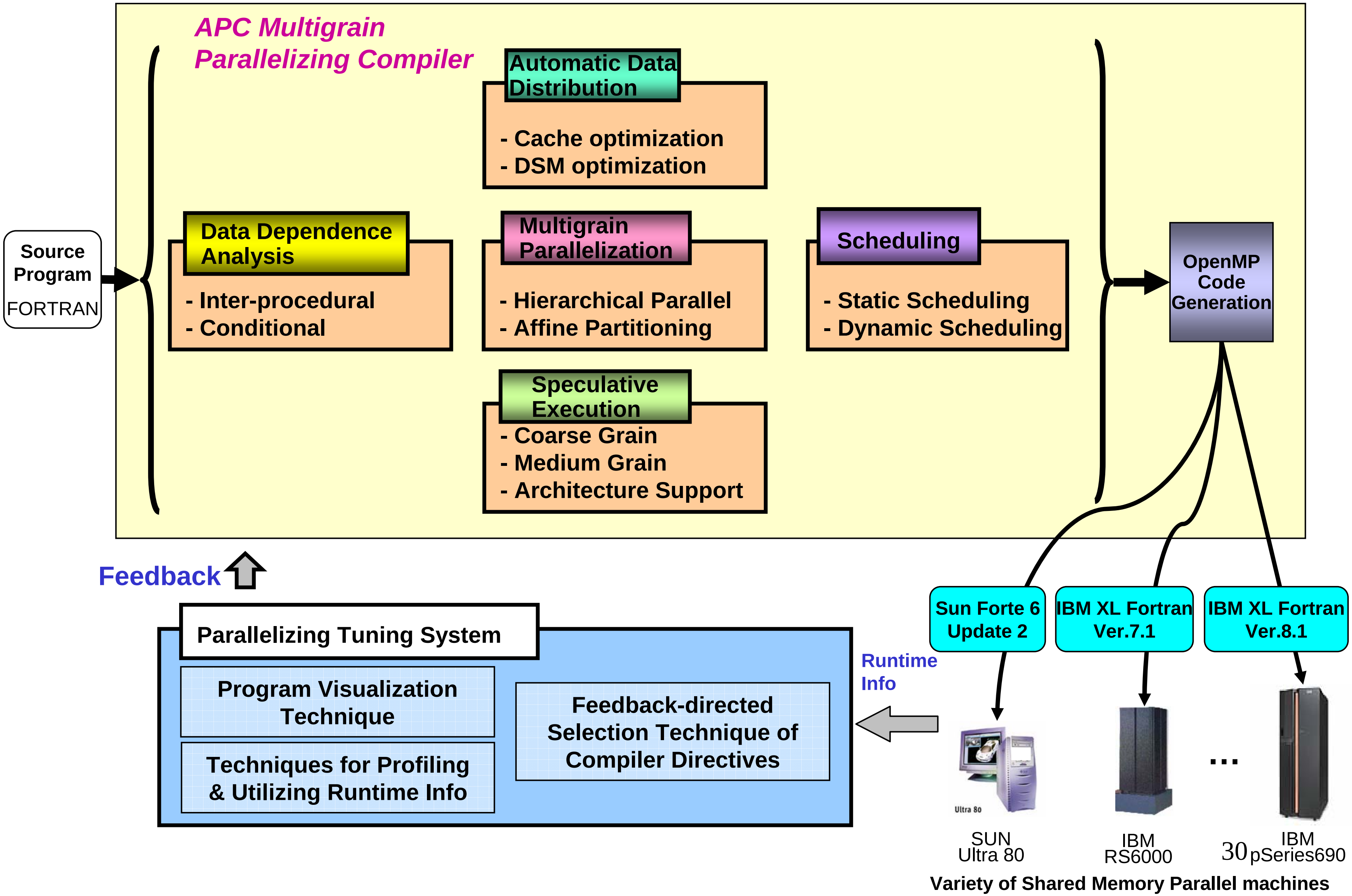
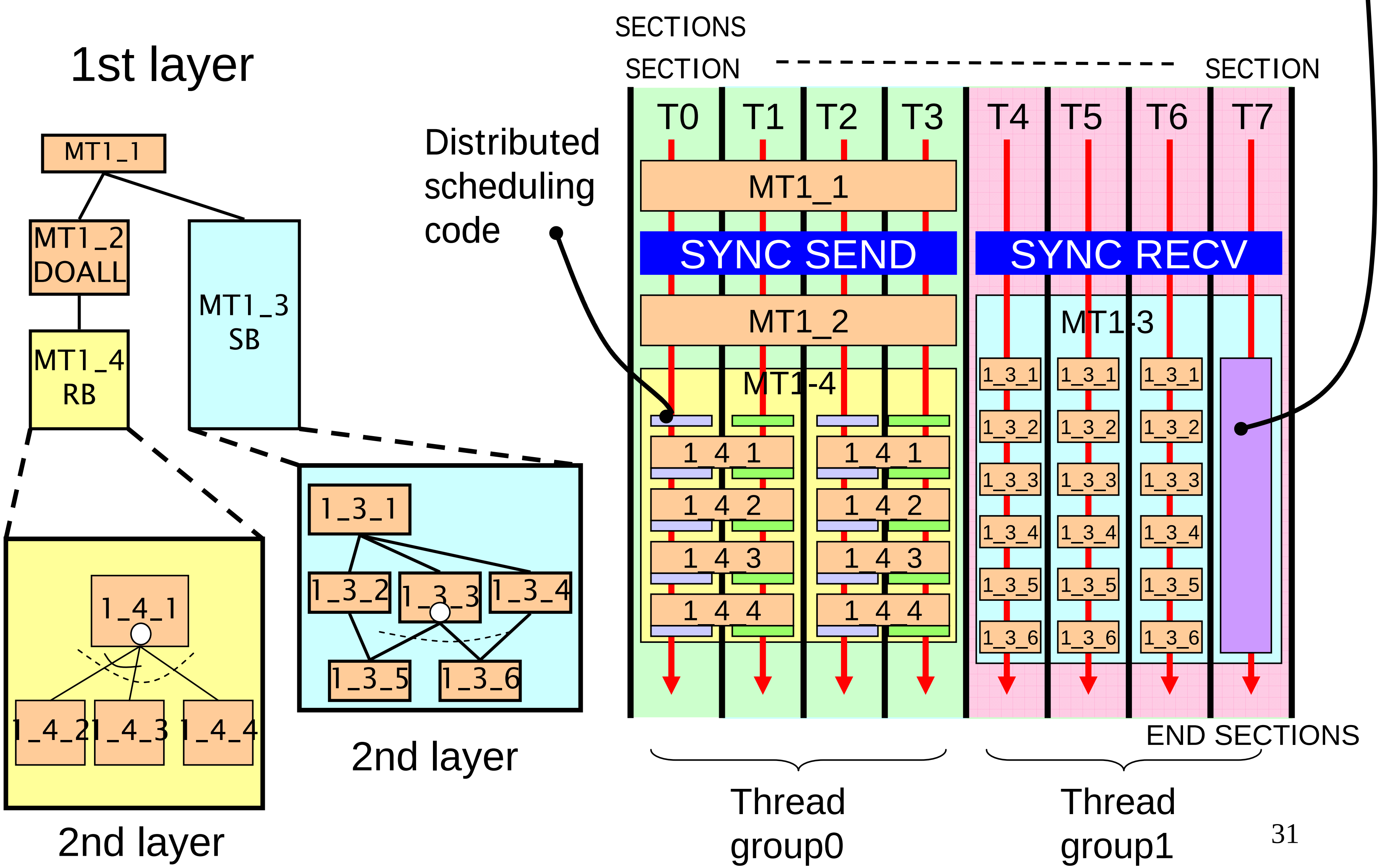
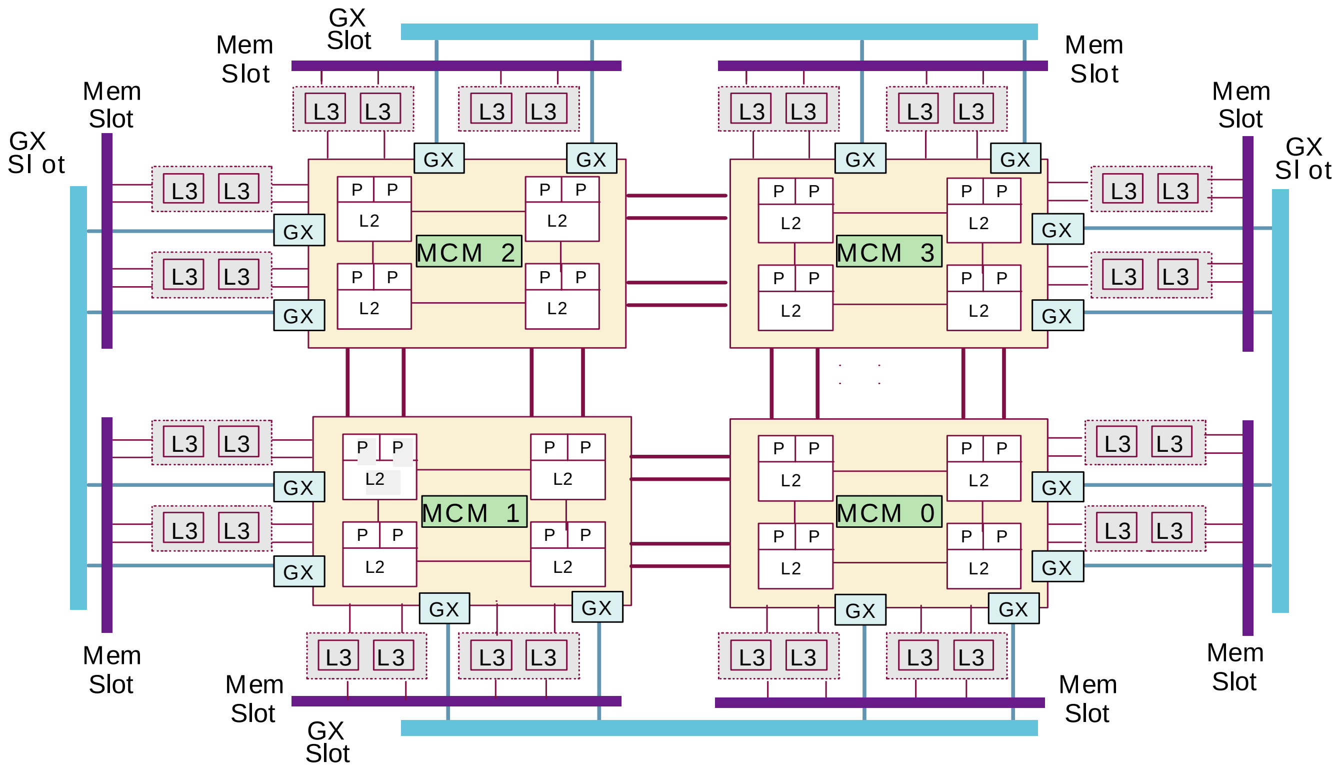


Image of Generated OpenMP Code for Hierarchical Multigrain Parallel Processing



IBM pSeries690 RegattaH

- Up to 16 Power4: 32 way SMP Server
  - L1(D) : 64 KB (32KB/processor, 2 way assoc.), L1(I): 128 KB (64KB/processor, Direct map)
  - L2 : 1.5 MB (shared cache with 2 procs., 4~8 way assoc.)
  - L3 : 32 MB (external, 8 way assoc.) [x 8 = 256 MB]

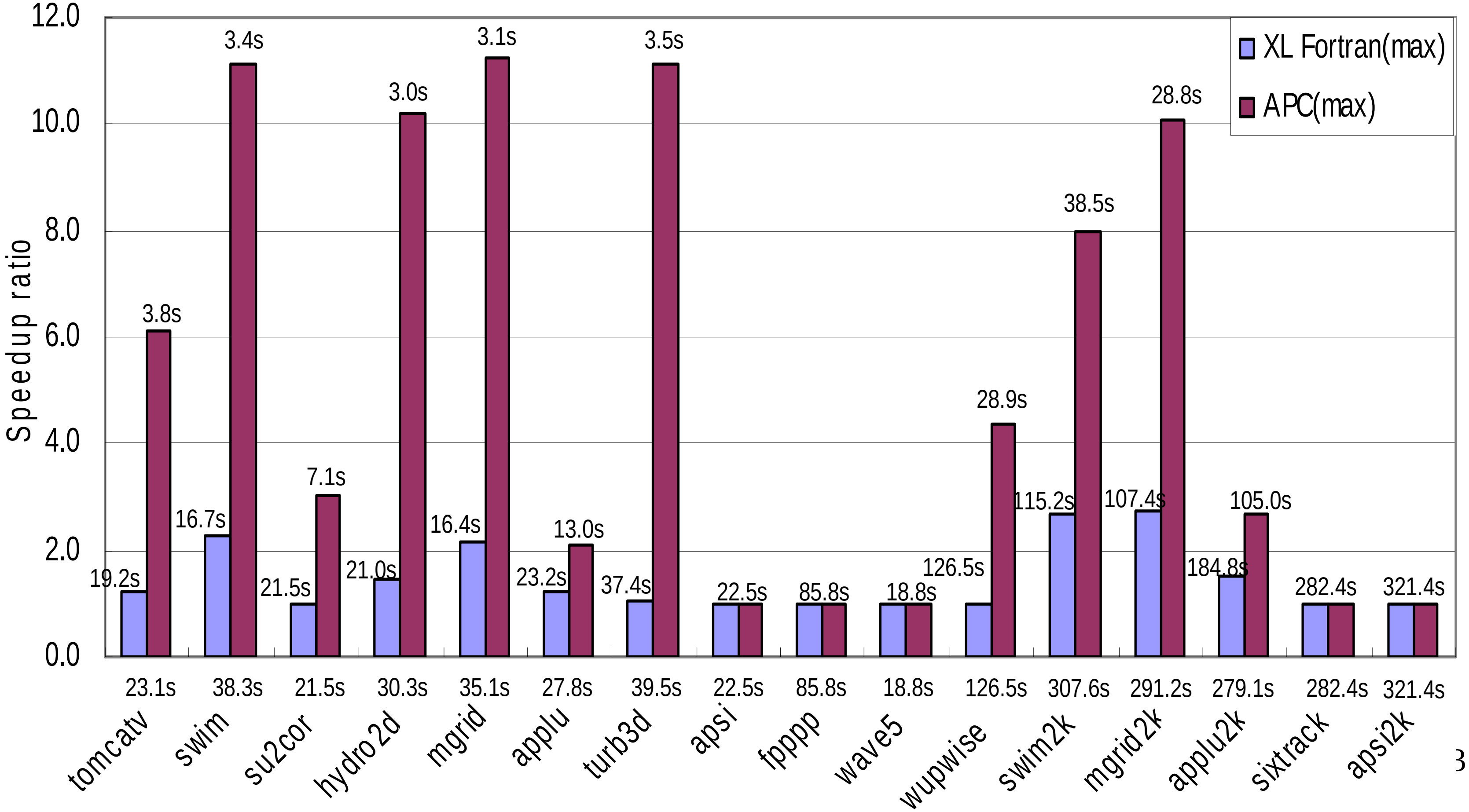


Four 8-way MCM Features Assembled into a 32-way pSeries 690

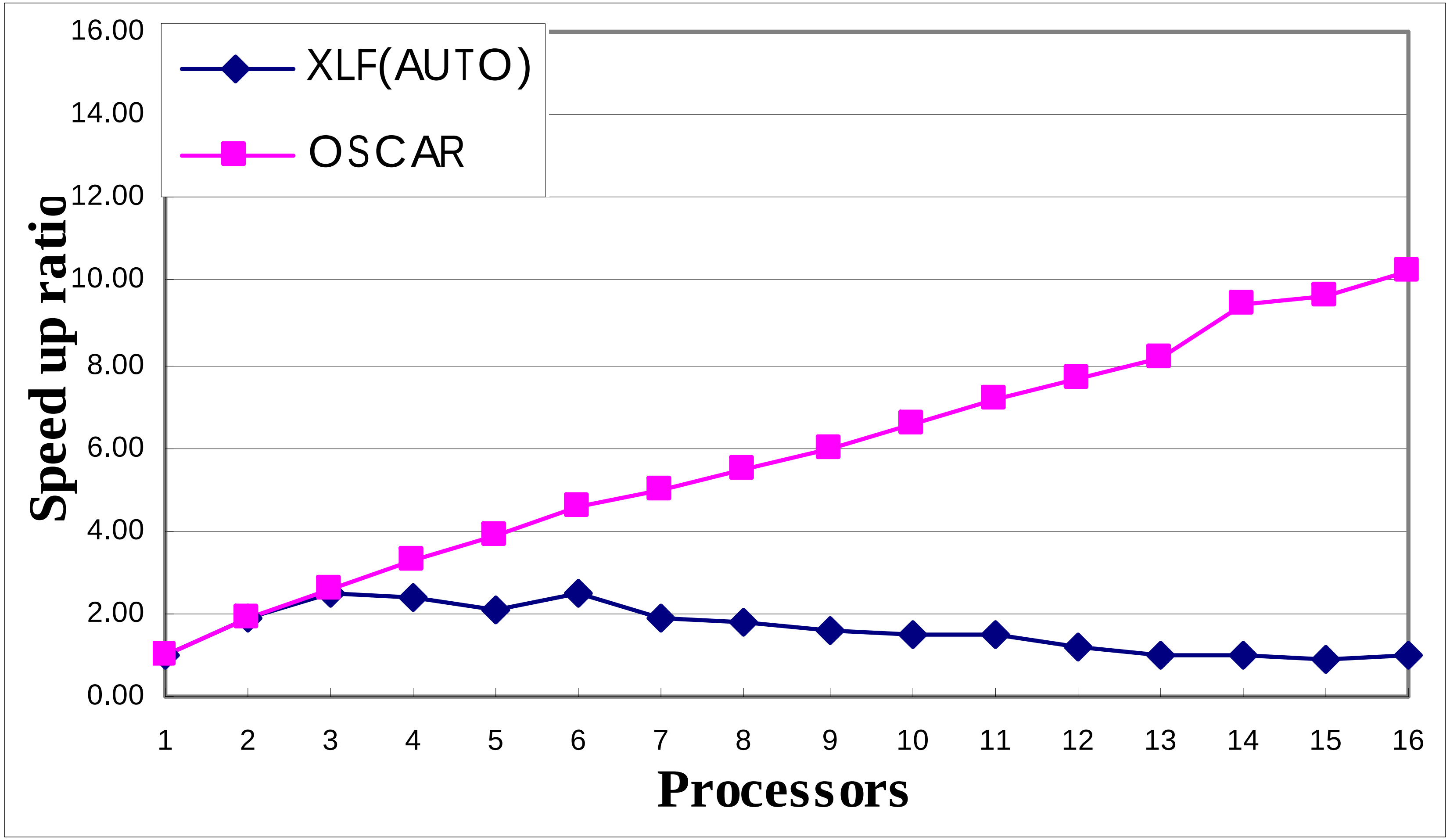
http://www-1.ibm.com/servers/eserver/pseries/hardware/whitepapers/p690_config.html

Performance of APC Compiler on IBM pSeries690 16 Processors High-end Server

- IBM XL Fortran for AIX Version 8.1
  - Sequential execution : -O5 -qarch=pwr4
  - Automatic loop parallelization : -O5 -qsmp=auto -qarch=pwr4
  - OSCAR compiler : -O5 -qsmp=noauto -qarch=pwr4 (su2cor: -O4 -qstrict)

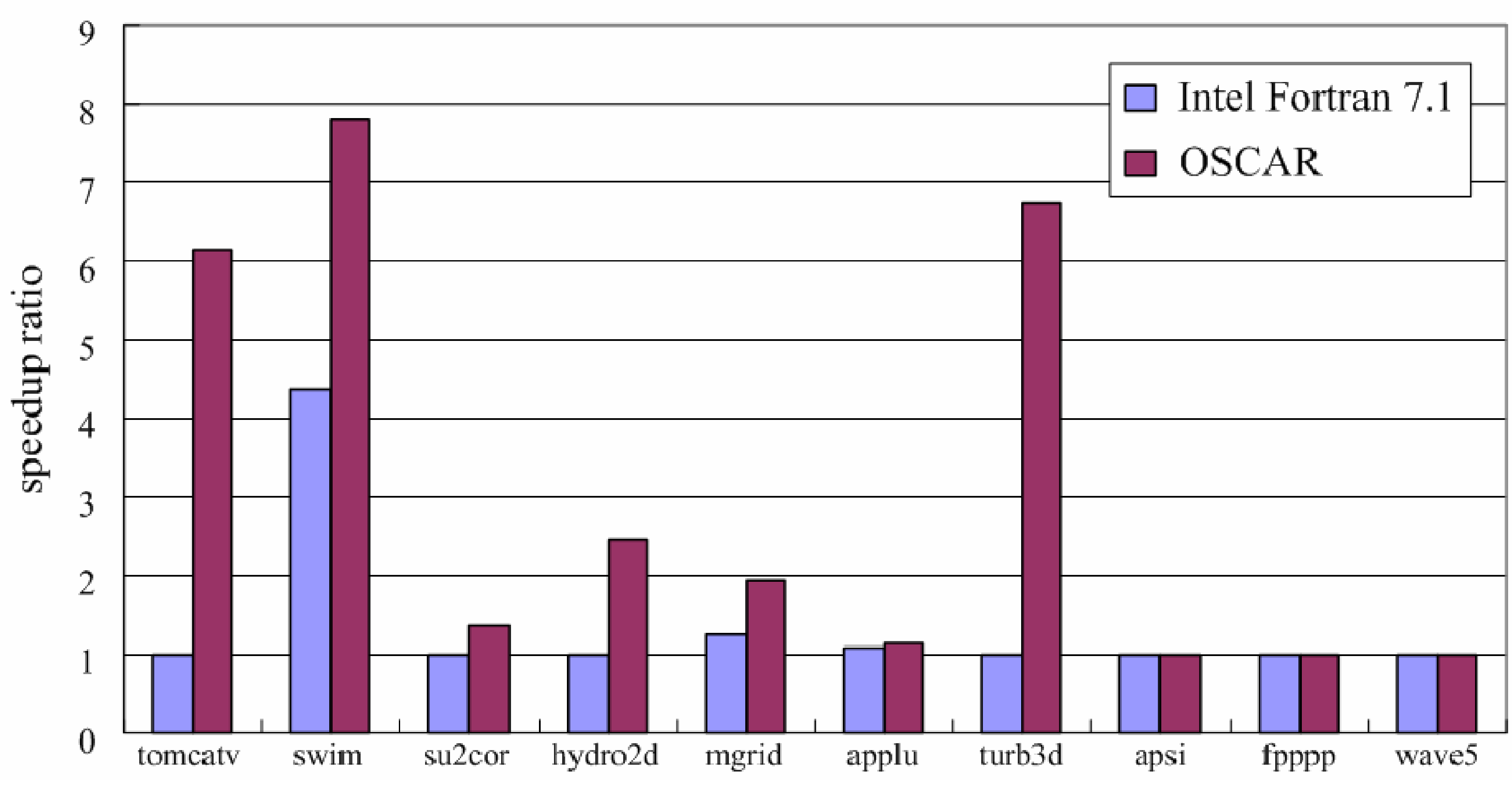


Performance of Multigrain Parallel Processing for 102.swim on IBM pSeries690



SGI Altix 3700

16 Itanium 2 (1.3GHz) SMP server
L1 16KB(4way), L2 256KB(8way), L3 3MB(12way)

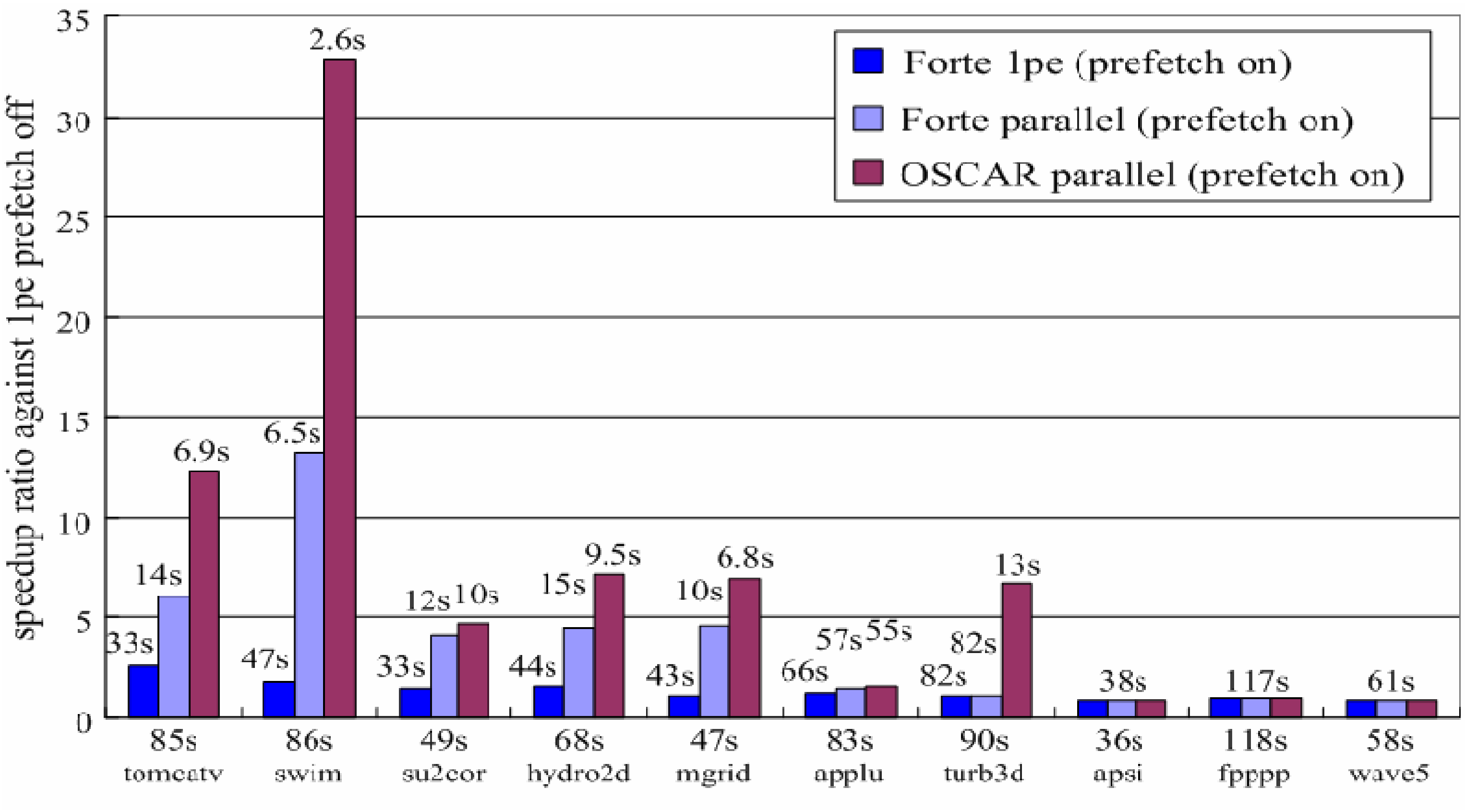


2.4 times speedup against Intel Fortran 7.1

Performance OSCAR multigrain Parallelizing Compiler on Sun V880

Data Localization and Cache Prefetching

Performance of OSCAR compiler on SUN V880(8 Ultra SPARC III Cu 1050MHz)

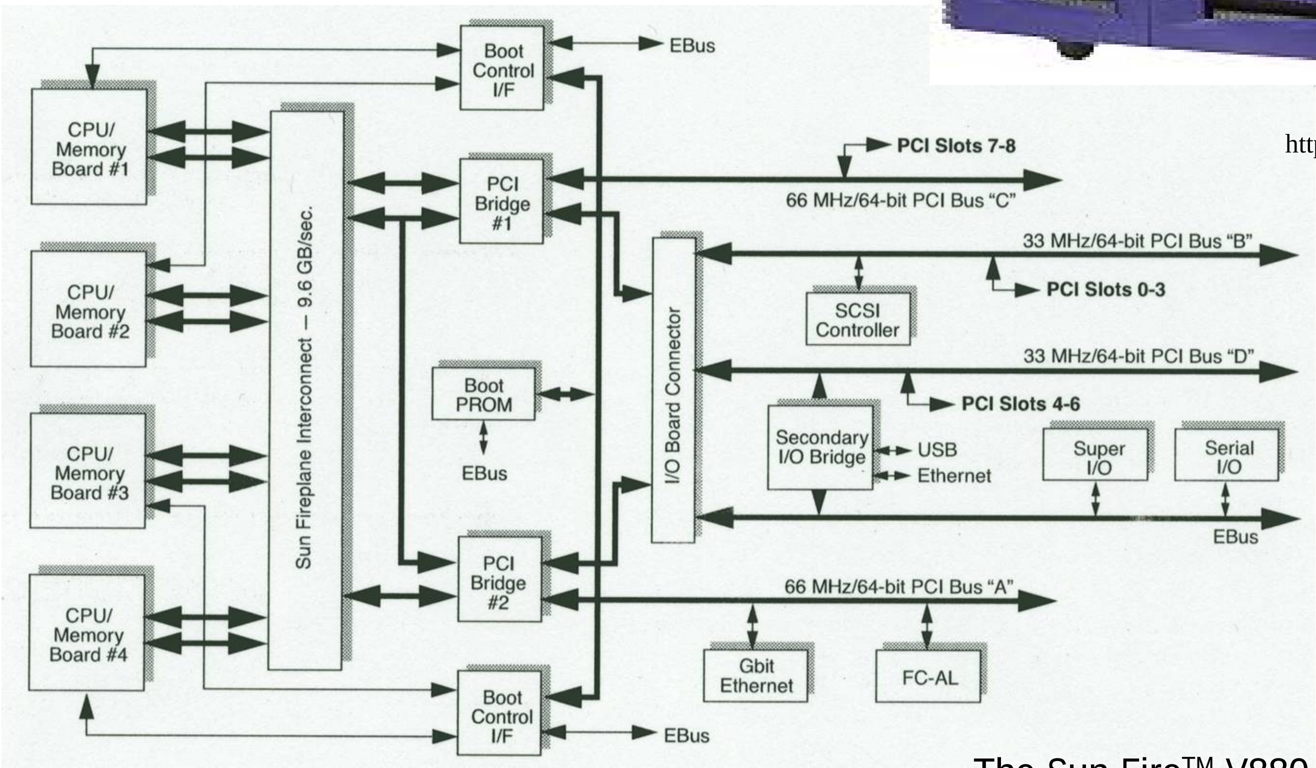


1.9 times speedup against Sun Forte compiler 7.1

Sun Fire V880

- UltraSPARCIII 8 processor SMP Sever
  - L1(Data) : 64 KB (4 way assoc.), L1(Inst.): 32 KB (4 way assoc.)
  - Prefetch : 2 KB (4 way assoc.)
  - L2 : 8 MB / processor (2 way assoc.)

http://www.sun.com/servers/entry/880/

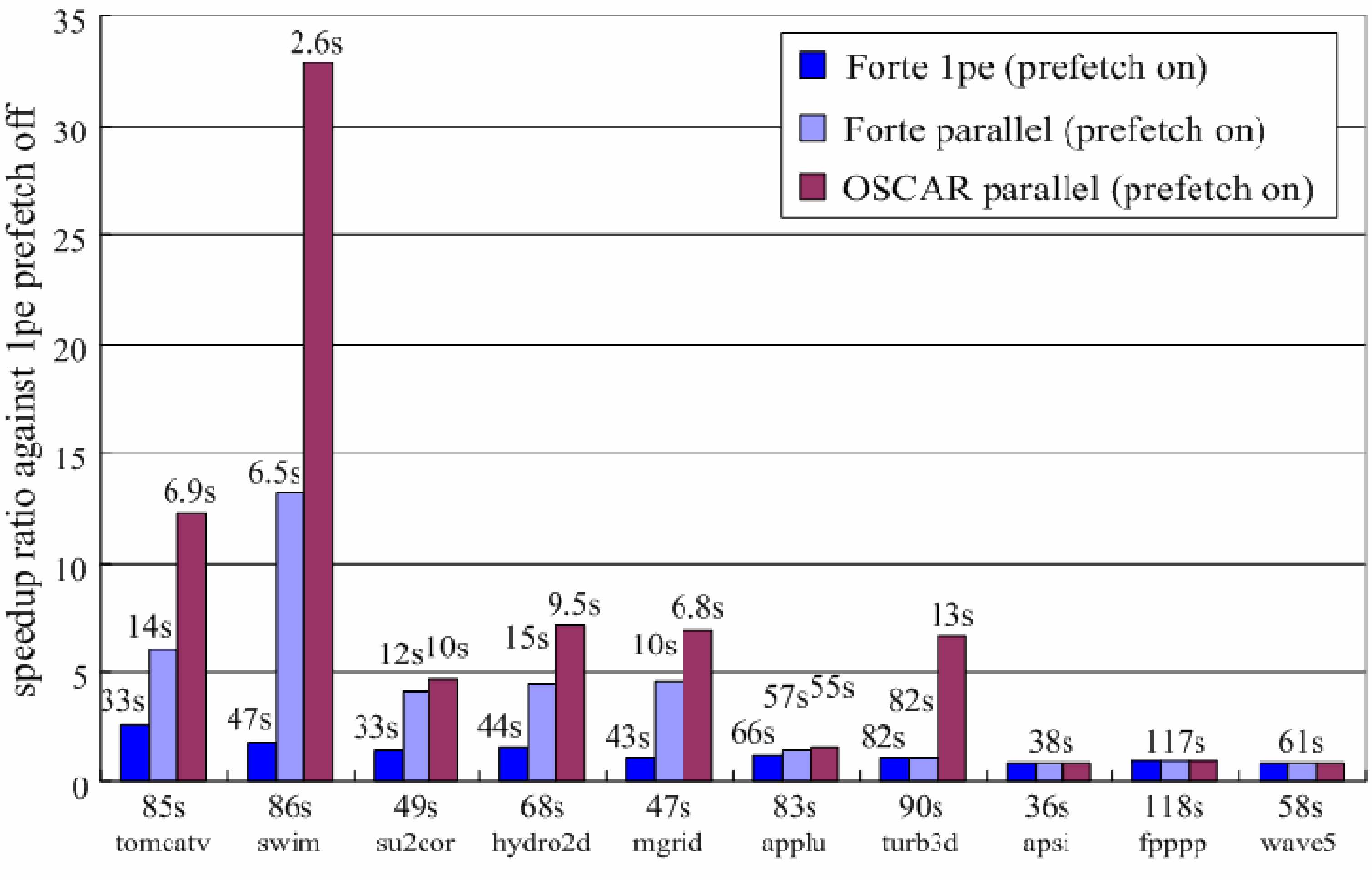


The Sun Fire™ V880 Server Architecture

Performance OSCAR multigrain Parallelizing Compiler on Sun V880

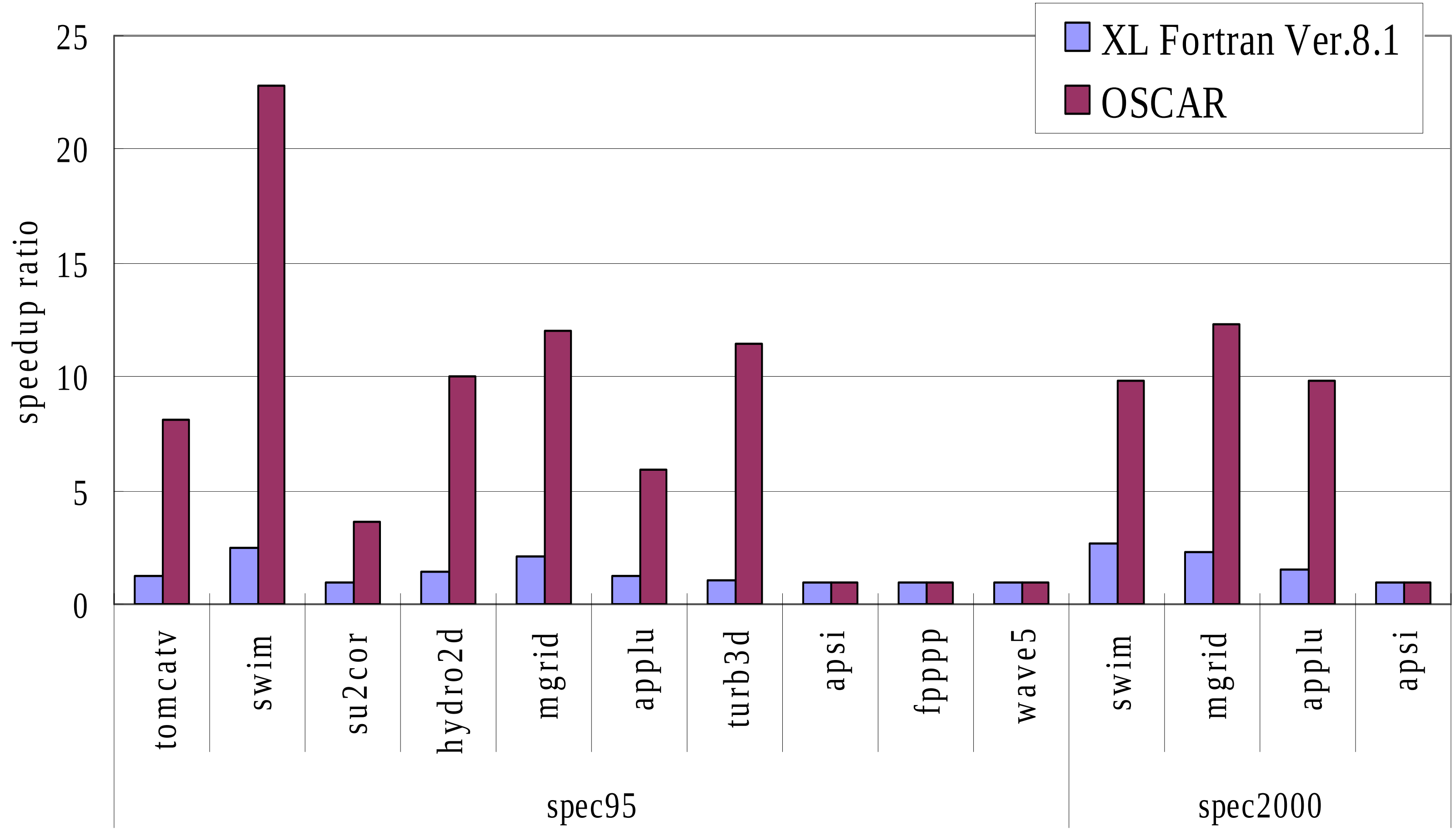
Data Localization and Cache Prefetching

Performance of OSCAR compiler on SUN V880(8 Ultra SPARC III Cu 1050MHz)



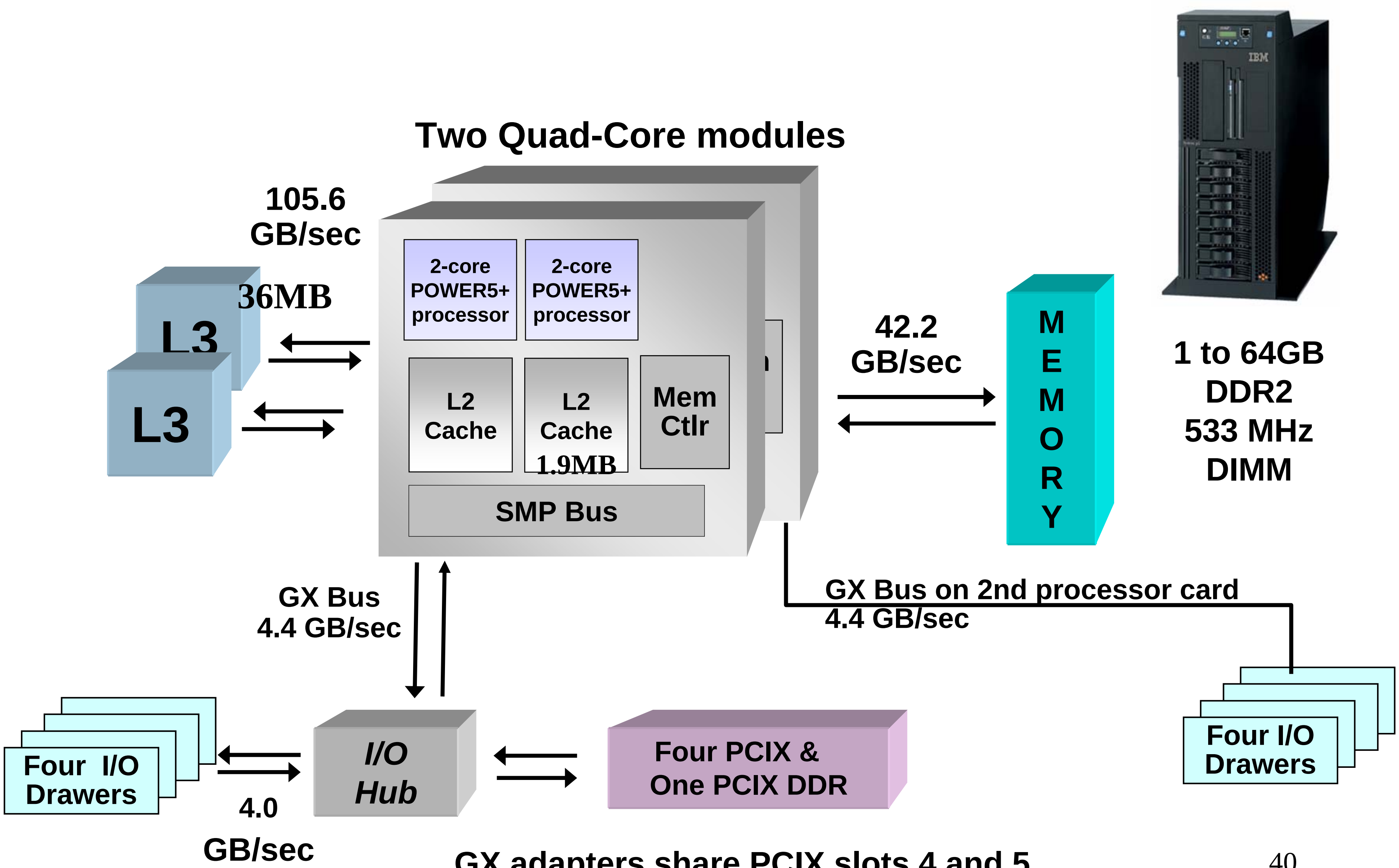
1.9 times speedup against Sun Forte compiler 7.1

Performance on IBM pSeries690 Power4 24 processors SMP server



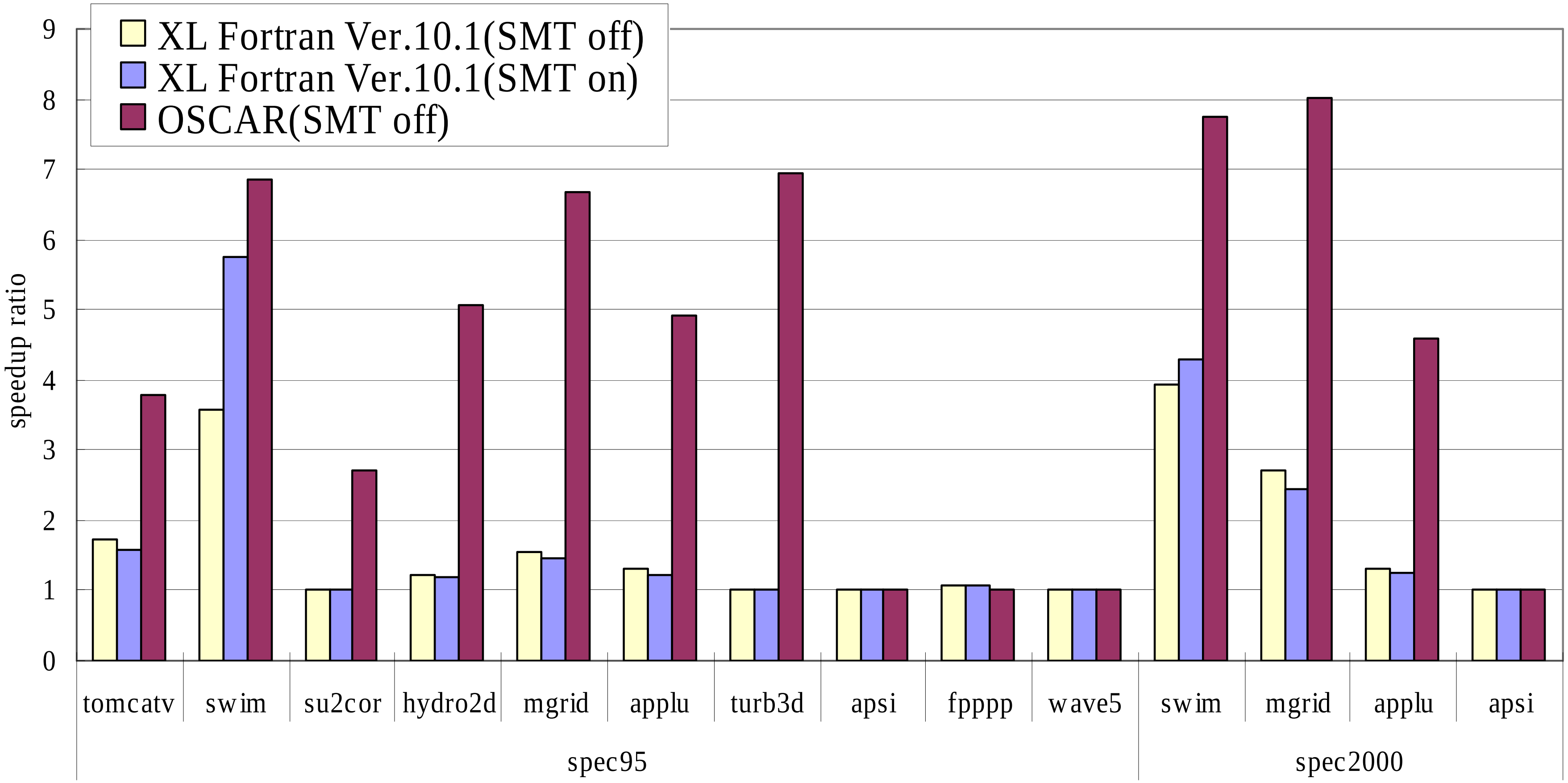
- OSCAR compiler gave us 4.82 times speedup against XL Fortran Ver.8.1

IBM p5-550Q server (1.5 or 1.65 GHz)



GX adapters share PCIX slots 4 and 5

Performance on IBM p5 550 POWER5+ 8 processors SMP server



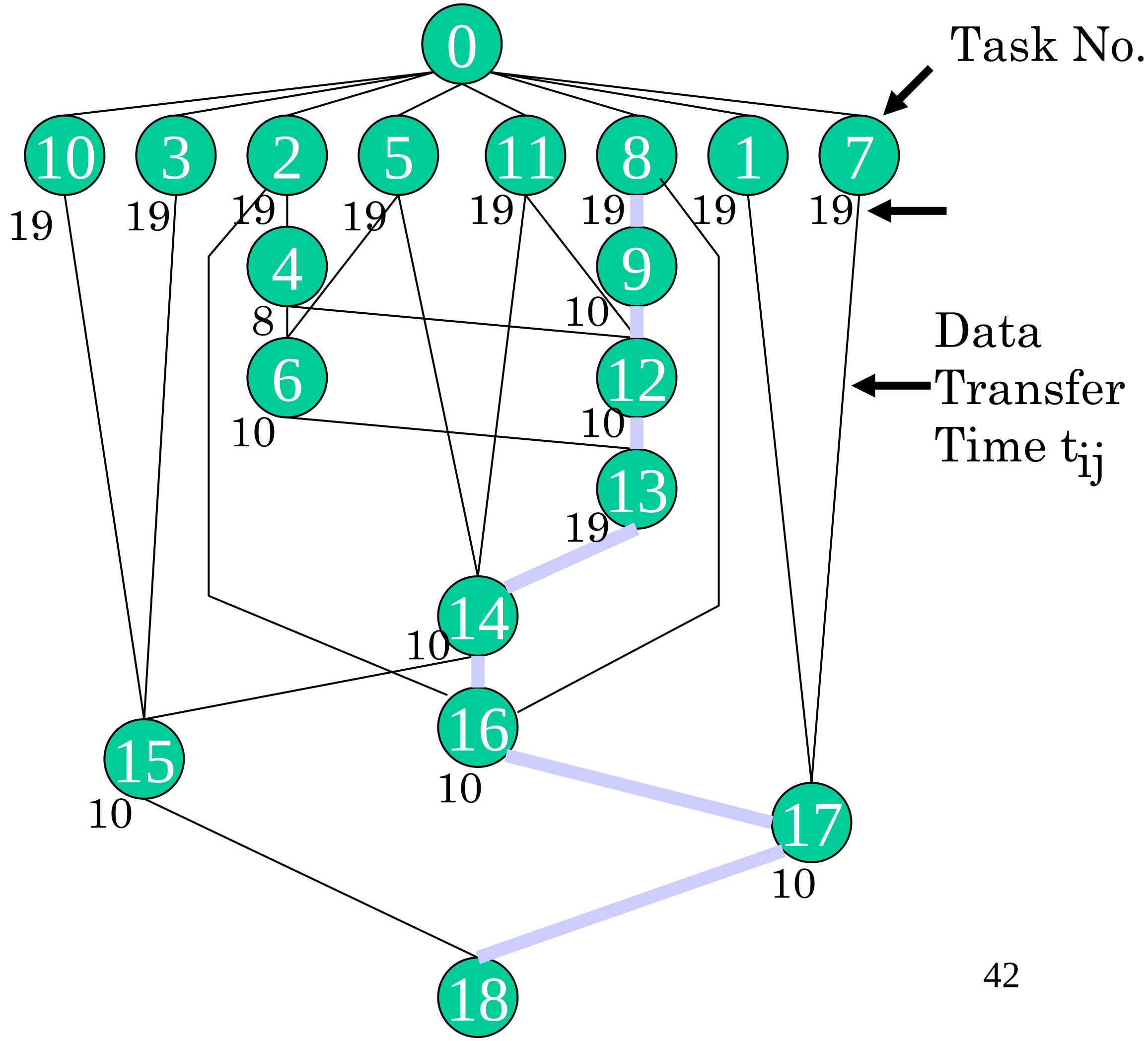
- OSCAR compiler
  - 2.74 times speedup against XL Fortran Ver.10.1 without SMT
  - 2.78 times speedup against XL Fortran Ver.10.1 with SMT

Example of Near Fine Grain Tasks

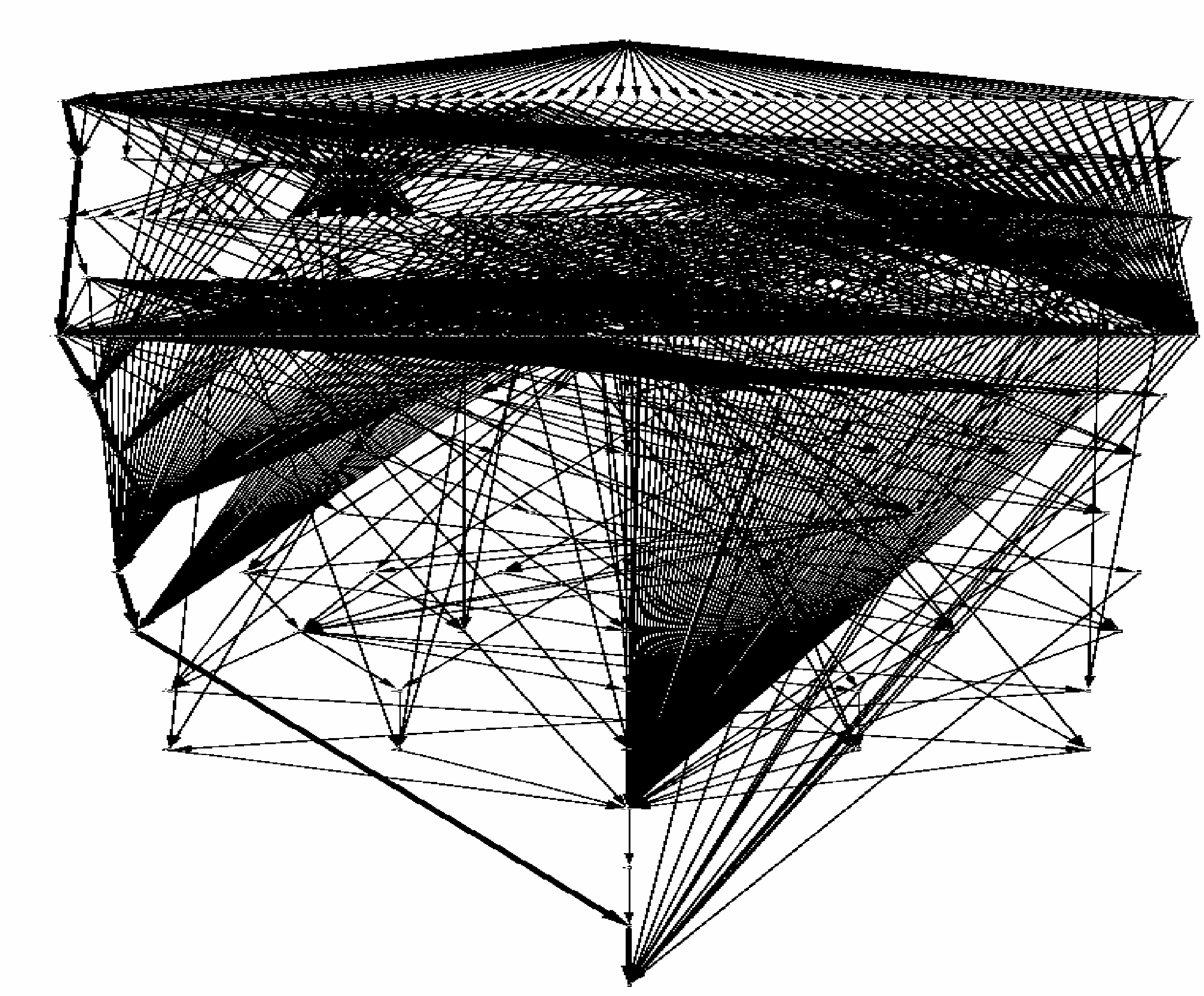
<<LU Decomposition>>
1) u12 = a12/l11
2) u24 = a24/l22
3) u34 = a34/l33
4) l54 = -l52 * u24
5) u45 = a45/l44
6) l55 = u55 - l54 * u45

<<Forward Substitution>>
7) y1 = b1 / l11
8) y2 = b2 / l22
9) b5 = b5 - l52 * y2
10) y3 = b3 / l33
11) y4 = b4 / l44
12) b5 = b5 - l54 * y4
13) y5 = b5 / l55

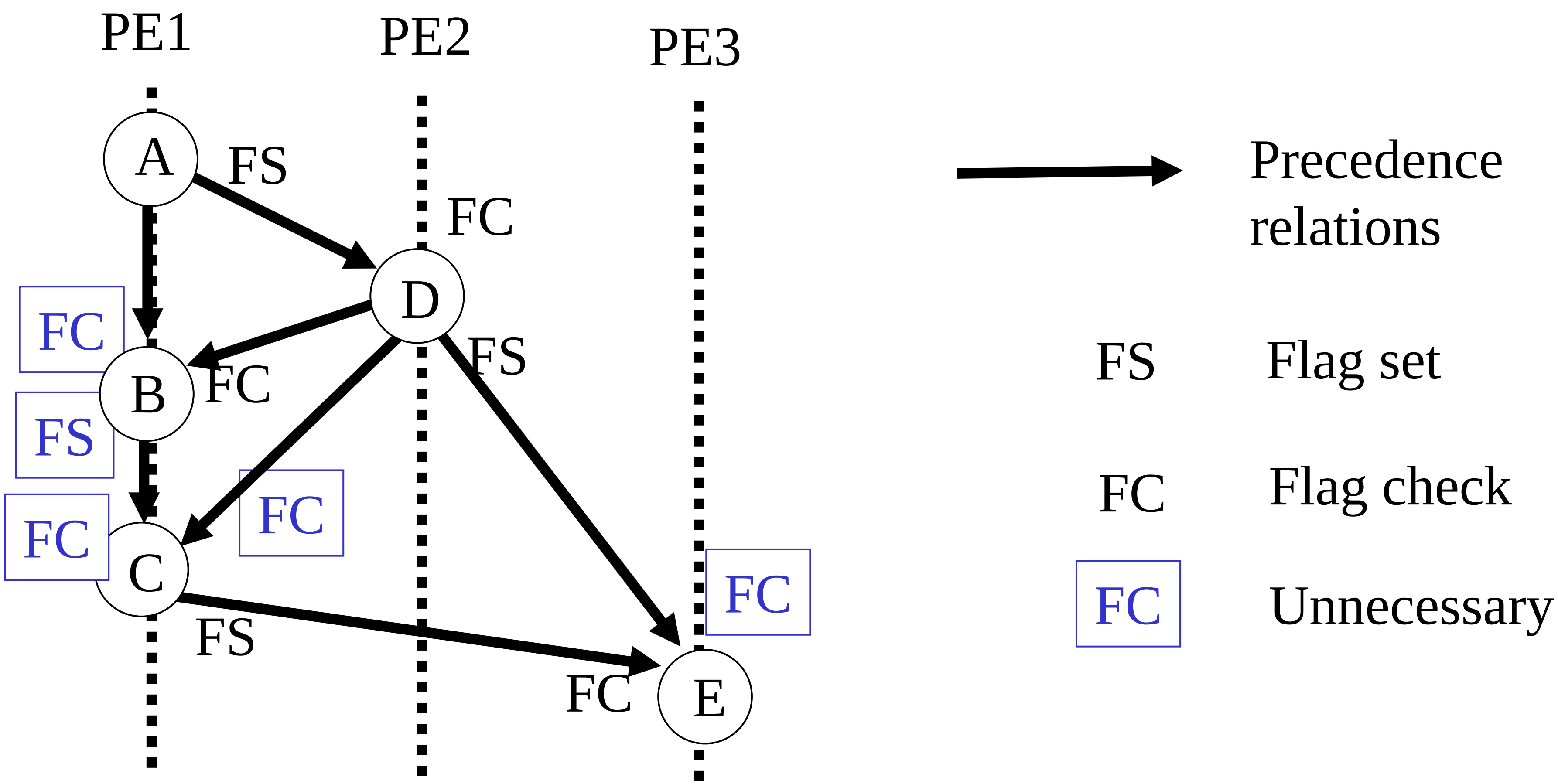
<<Backward Substitution>>
14) x4 = y4 - u45 * y5
15) x3 = y3 - u34 * x4
16) x2 = y2 - u24 * x4
17) x1 = y1 - u12 * x2



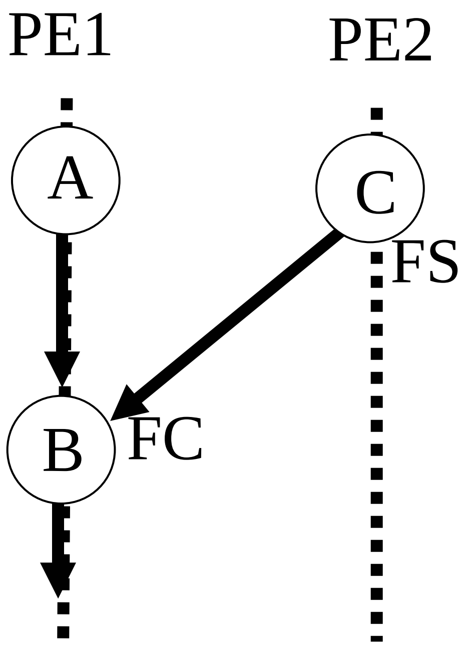
Task Graph for FPPPP



Redundant Synchronization Removal Using Static Scheduling Results on Shared Memory Multiprocessors



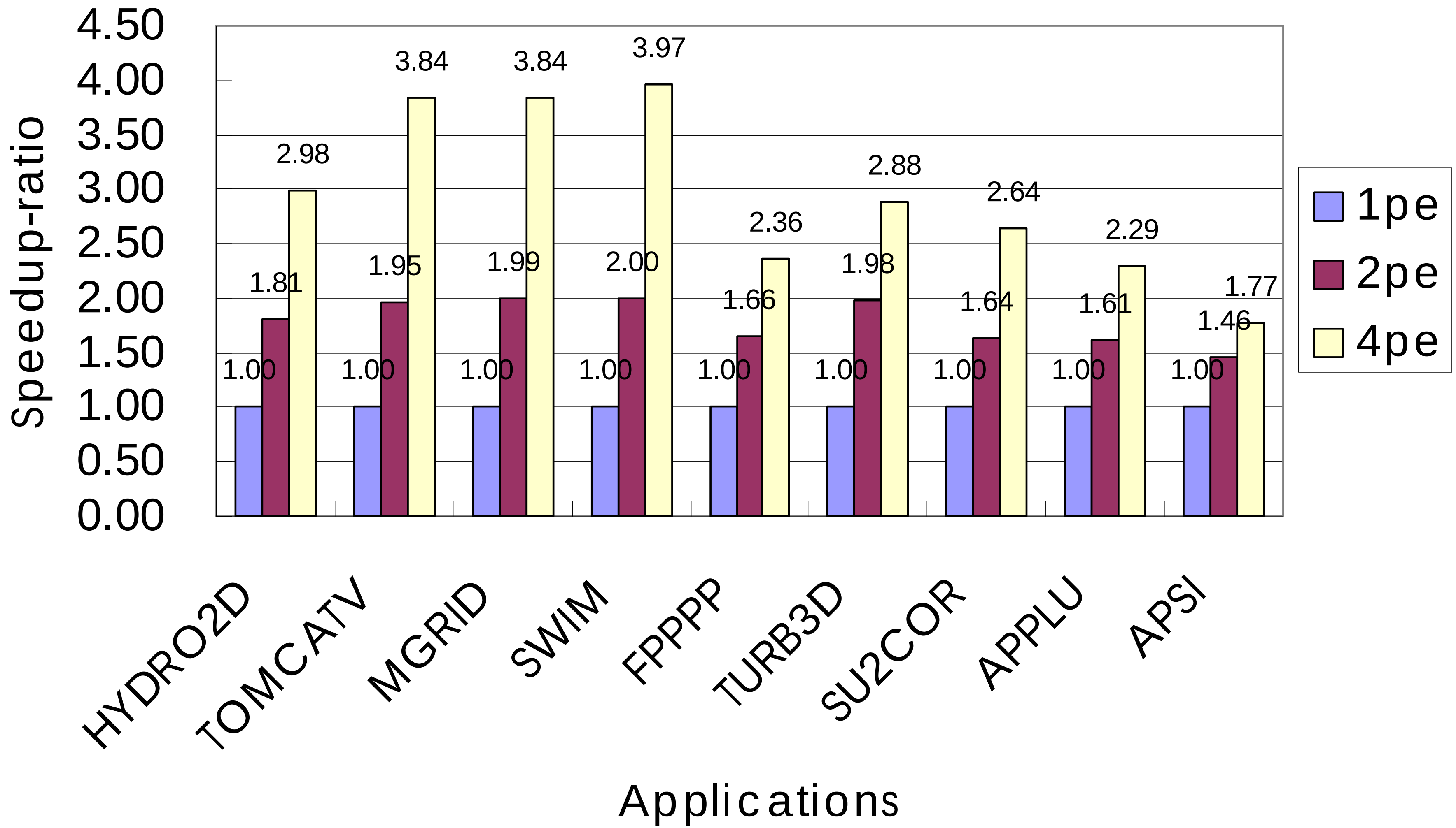
Generated parallel machine code for near fine grain parallel processing




```
      PE1
; ---- Task A ----
; Task Body
      ⋮
  FADD R23, R19, R21
; ---- Task B ----
; Flag Check : Task C
L18:
  LDR R28, [R14, 0]
  CMP R28, R0
  JNE  L18
; Data Receive
  LDR R29, [R14, 1]
; Task Body
  FMLT R24, R23, R29
      ⋮
```

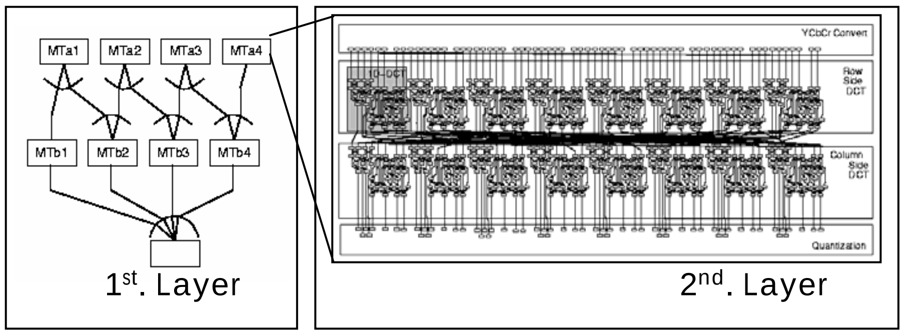
```
      PE2
; ---- Task C ----
; Task Body
      ⋮
  FMLT R27, R28, R29
  FSUB R29, R19, R27
; Data Transfer
  STR [R14, 1], R29
; Flag Set
  STR [R14, 0], R0
      ⋮
```

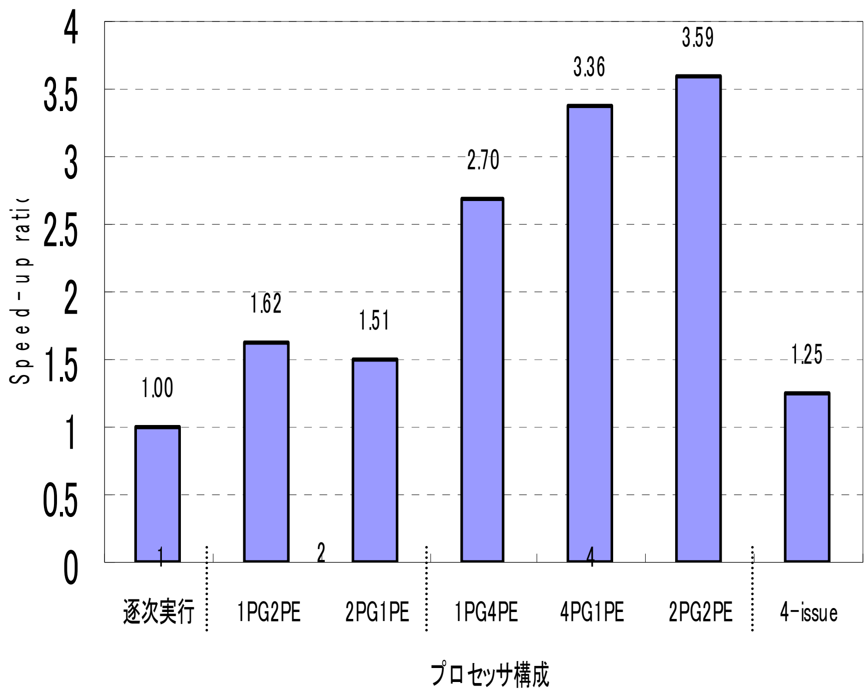
Performance of Multigrain Parallel Processing on OSCAR CMP



Multigrain Parallel Processing of JPEG Encoding on OSCAR CMP

- 処理単位：8×8ピクセルブロック
  - 8×8ピクセルブロック間での並列性
    ⇒　粗粒度並列処理
  - 8×8ピクセルブロック内の並列性
    ⇒　近細粒度並列処理
- 8×8ピクセルブロック間の粗粒度並列性と8×8ピクセルブロック内の近細粒度並列性を階層的に利用したマルチグレイン並列処理
  - <u>**4プロセッサ使用時，逐次処理に対して3.59倍の速度向上**</u>

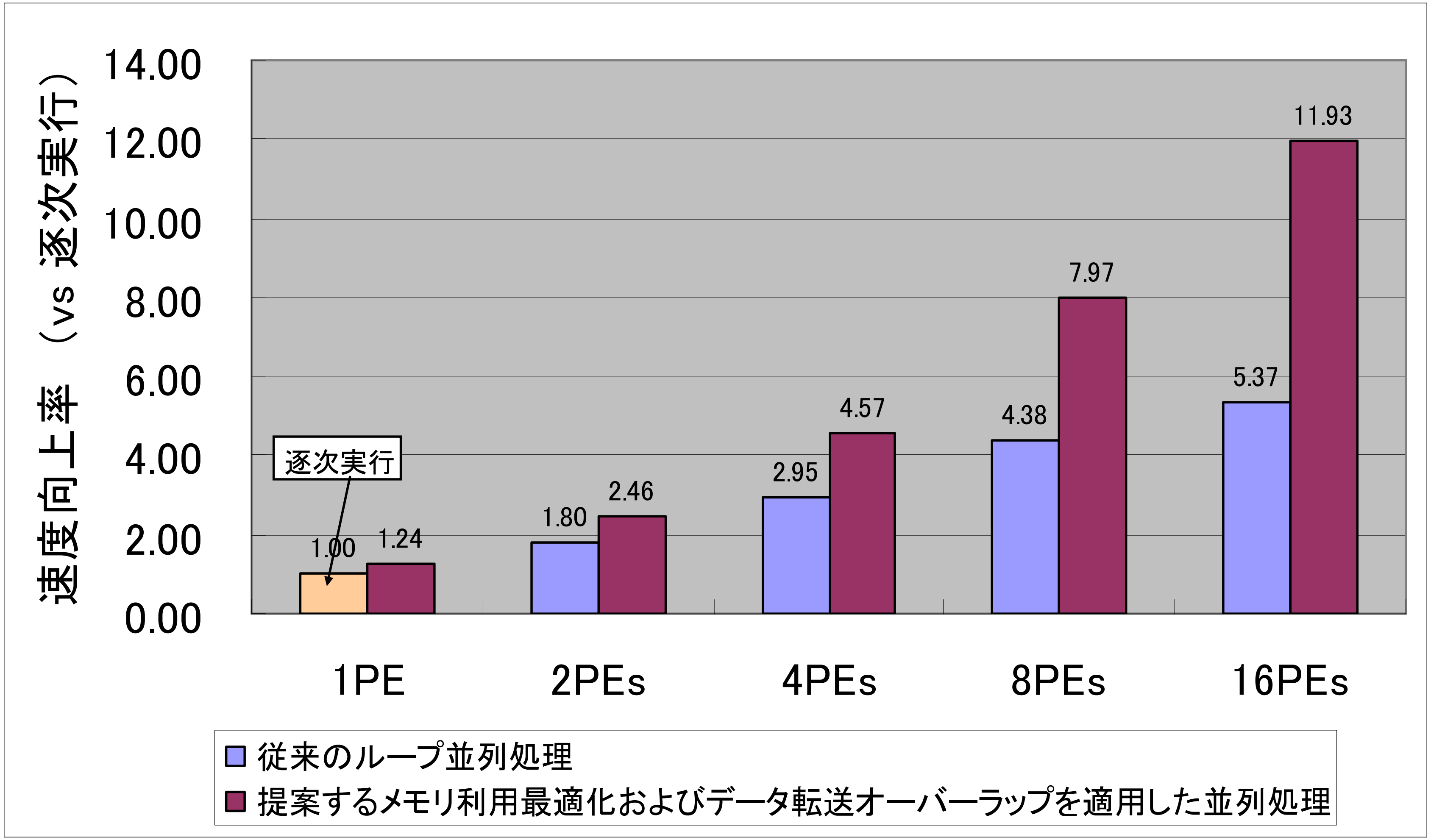




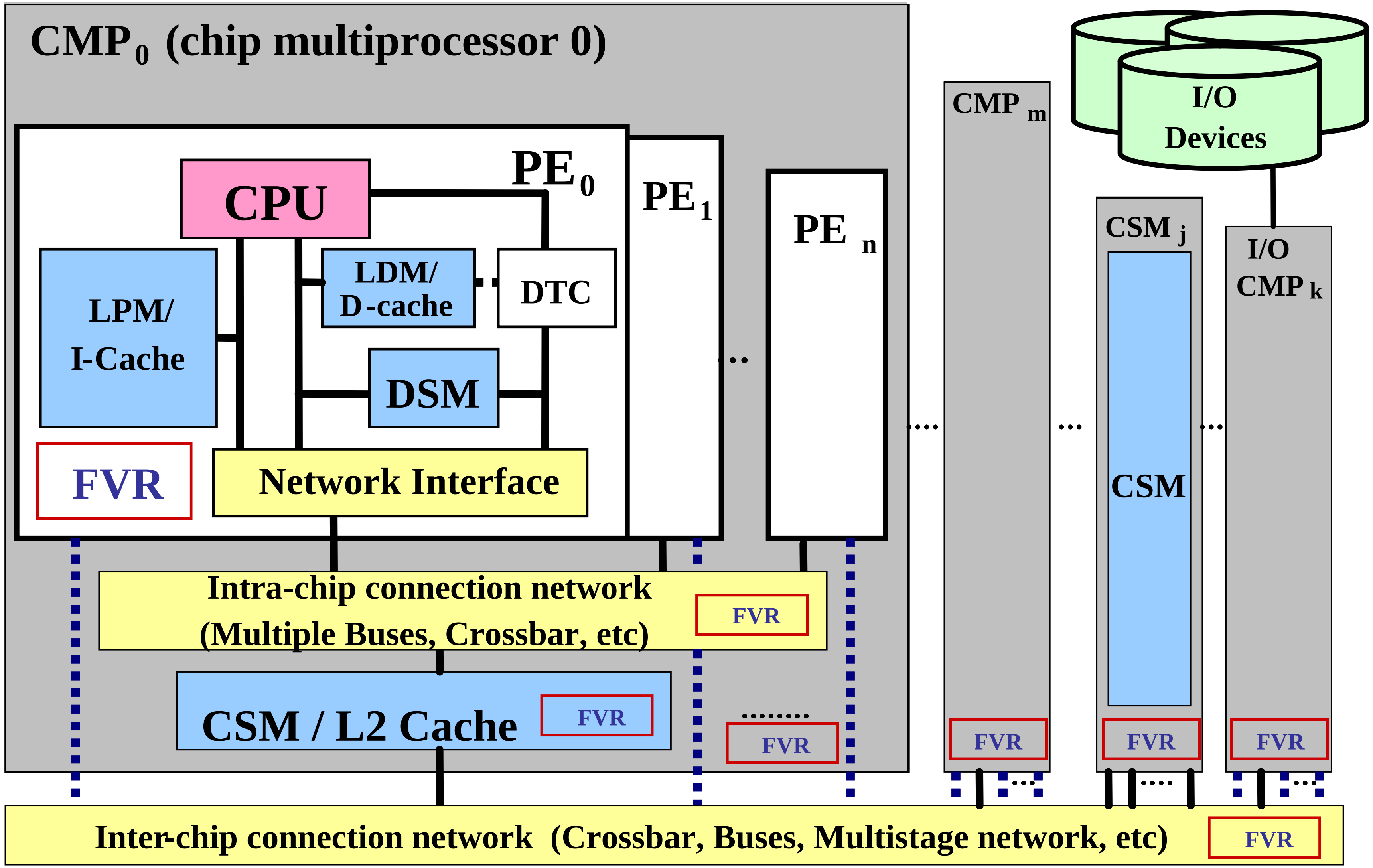
•2PG2PE　：　マルチグレイン並列処理

Performance of MPEG2 Encoding on OSCAR Chip Multiprocessor
(With data localization and data transfer overlapping)

○性能評価条件
•クロックレベルシミュレータにて計測，OSCARチップマルチプロセッサ(400MHz相当、90nmテクノロジ）
•4フレームエンコード（256x256ピクセルサイズ）, エンコードパラメータは、MediaBench相当　（但し、画像サイズ縮小）



OSCAR Multi-Core Architecture



CSM: central shared mem.

DSM: distributed shared mem.

DTC: Data Transfer Controller

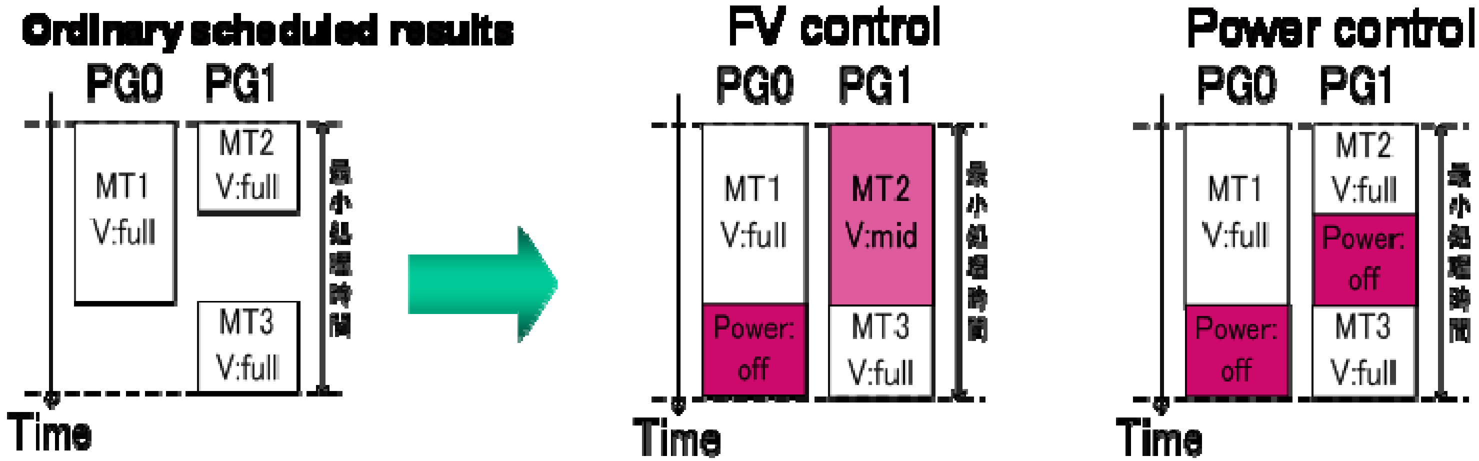
LDM : local data mem.

LPM : local program mem.

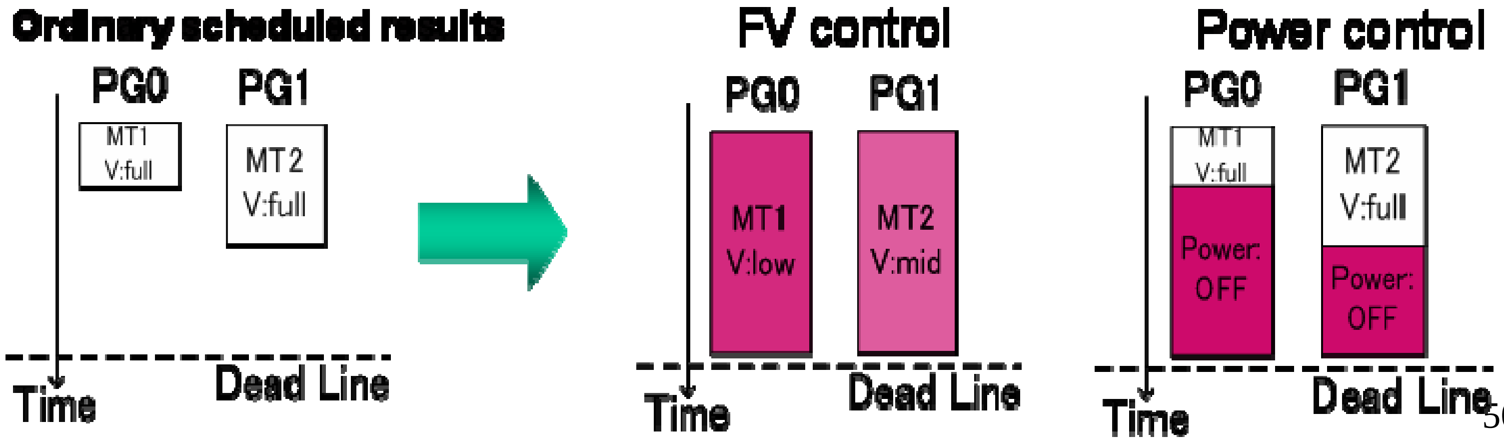
FVR: frequency / voltage control register

Power Reduction by Power Supply, Clock Frequency and Voltage Control by OSCAR Compiler

- Shortest execution time mode



- Realtime processing mode with dead line constraints



An Example of Machine Parameters for the Power Saving Scheme

- **Functions of the multiprocessor**
  - **Frequency of each proc. is changed to several levels**
  - **Voltage is changed together with frequency**
  - **Each proc. can be powered on/off**

state	FULL	MID	LOW	OFF
frequency	1	1 / 2	1 / 4	0
voltage	1	0.87	0.71	0
dynamic energy	1	3 / 4	1 / 2	0
static power	1	1	1	0

- **State transition overhead**

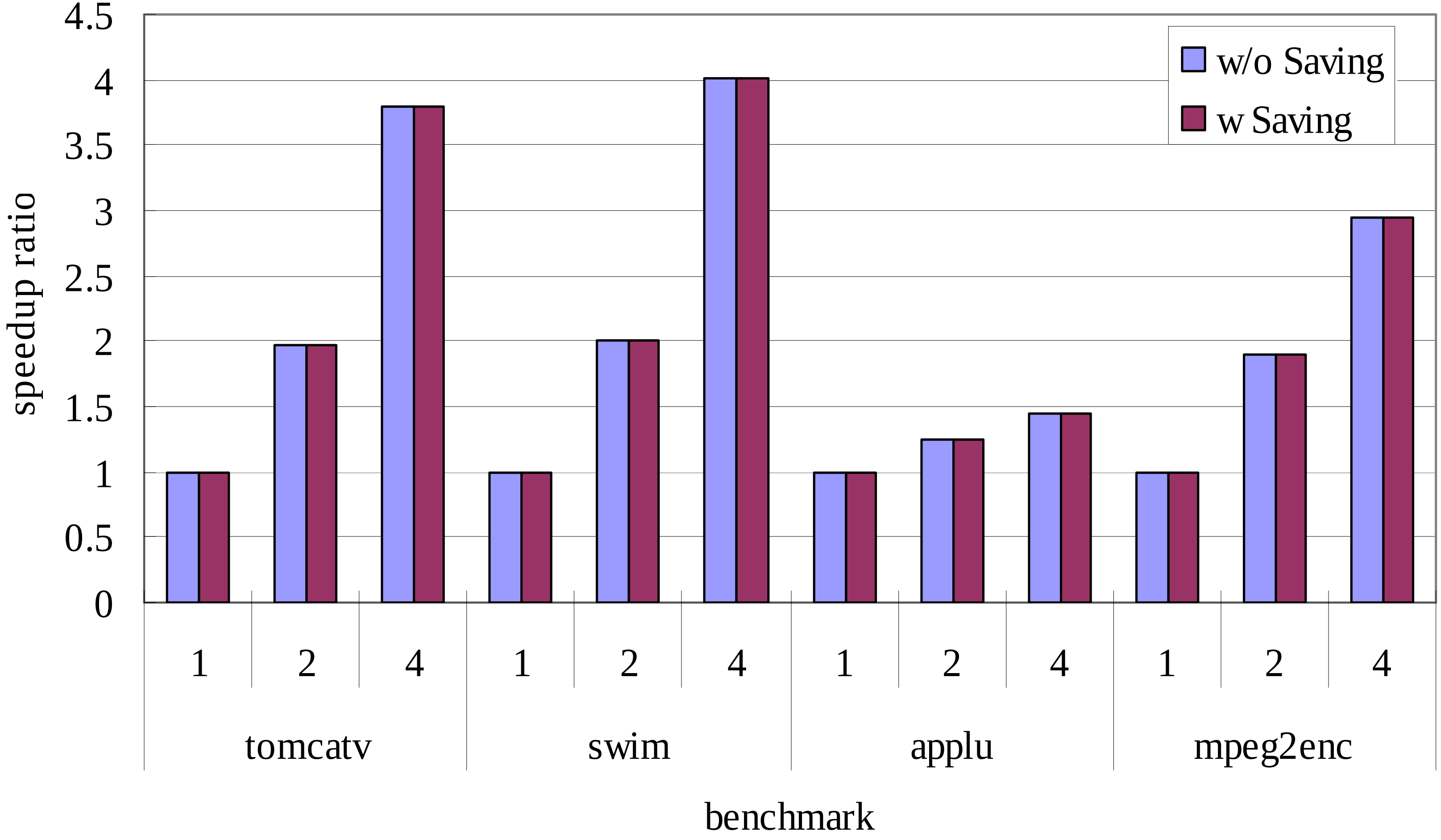
state	FULL	MID	LOW	OFF
FULL	0	40k	40k	80k
MID	40k	0	40k	80k
LOW	40k	40k	0	80k
OFF	80k	80k	80k	0

delay time [u.t.]

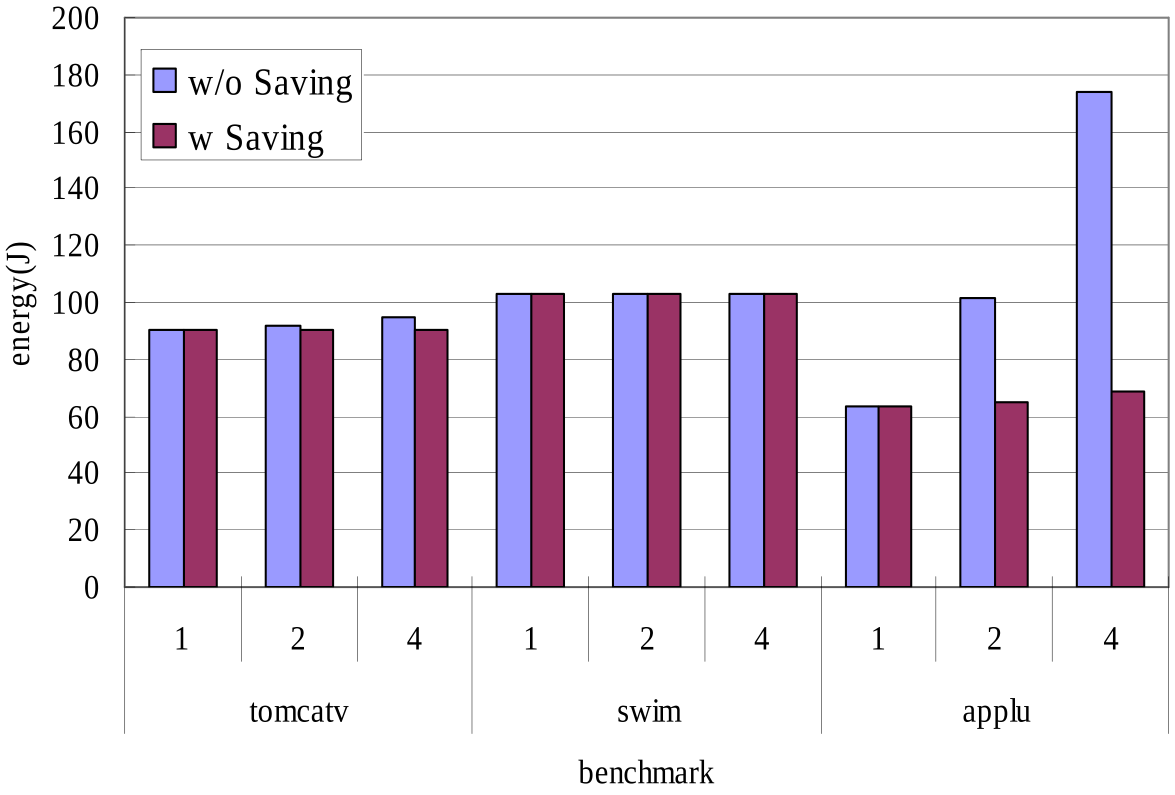
state	FULL	MID	LOW	OFF
FULL	0	20	20	40
MID	20	0	20	40
LOW	20	20	0	40
OFF	40	40	40	0

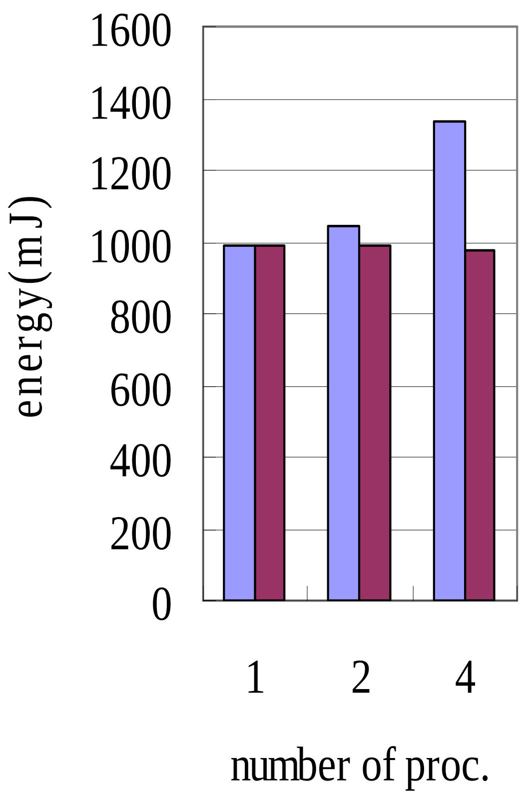
energy overhead [μ J]

Speed-up in Fastest Execution Mode



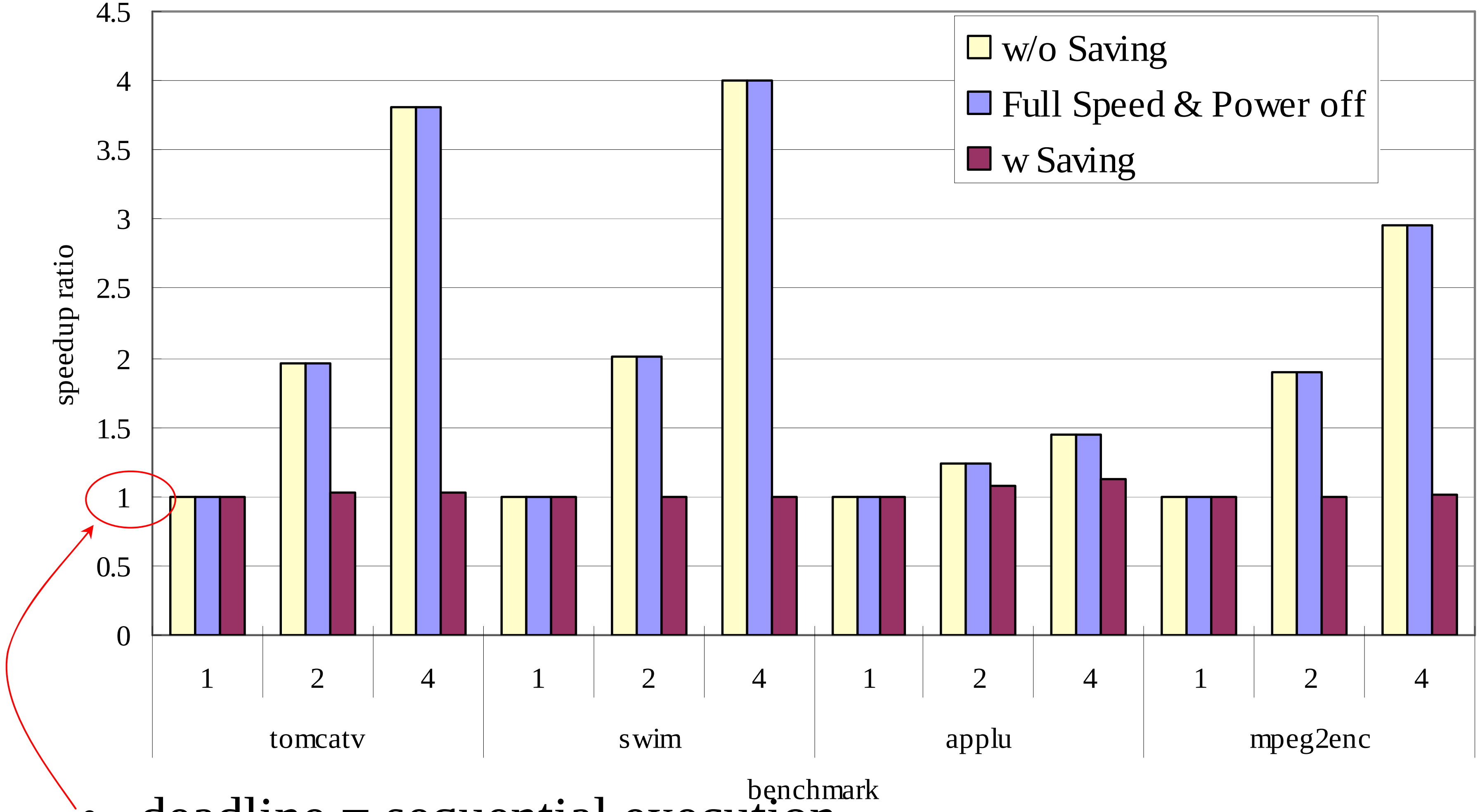
Consumed Energy in Fastest Execution Mode





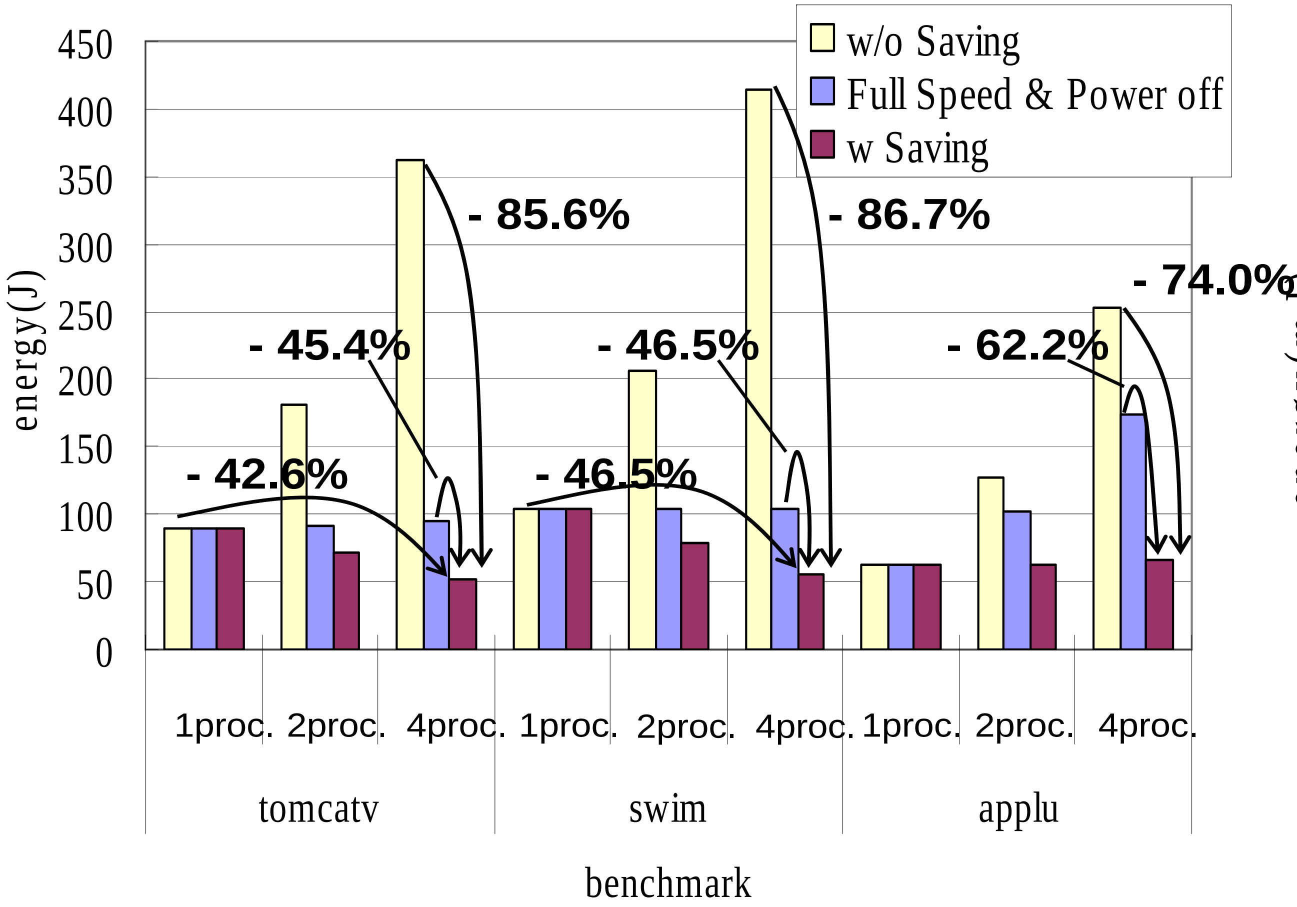
mpeg2_encode

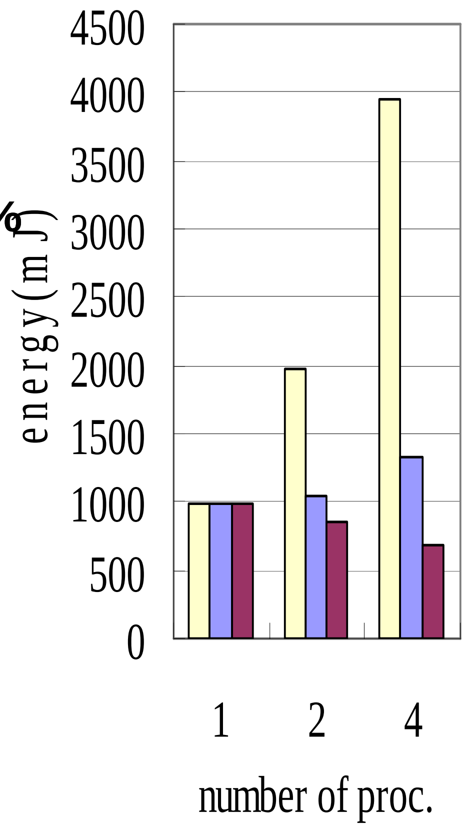
Speed-up in Real-time Execution Mode



- deadline = sequential execution time

Consumed Energy in Real-time Processing mode





MPEG2 Encode

- deadline = sequential execution time

Conclusions

- **Compiler cooperative low power high effective performance multi-core processors will be more important in wide range of information systems from games, mobile phones, automobiles to peta-scale supercomputers.**
- **Parallelizing compilers are essential for realization of**
  - **Good cost performance**
  - **Short hardware and software development periods**
  - **Low power consumption**
  - **High software productivity**
  - **Scalable performance improvement with advancement in semiconductor integration technology**
- **Key technologies in multi-core compiler**
  - **Multigrain parallelization, Data localization, Data transfer overlapping using DMA, Low power control technologies**