

# New features of ParFORM

SFB-TR9, Project A2:

M. Frank, H.M. Staudenmaier, M. Tentyukov

Fakultät für Physik, Universität Karlsruhe

## FORM

J.A.M. Vermaseren  
NIKHEF, Amsterdam

FORM is a program for symbolic manipulation of algebraic expressions specialized to handle very large expressions of **millions of terms** in an efficient and reliable way.

## ParFORM

Fakultät für Physik, Universität Karlsruhe

ParFORM is a parallel version of FORM developed in Karlsruhe.

There are some **internal mechanisms** of FORM that become important in its parallelization. In fact, ParFORM allows to parallelise the solution of algorithmically non-parallelizable problems. Example: reduction to master integrals via recurrence relations.

D. Fliegner, A. Rétey:

**Working prototype** 1998 – 2000.

A. Onischenko:

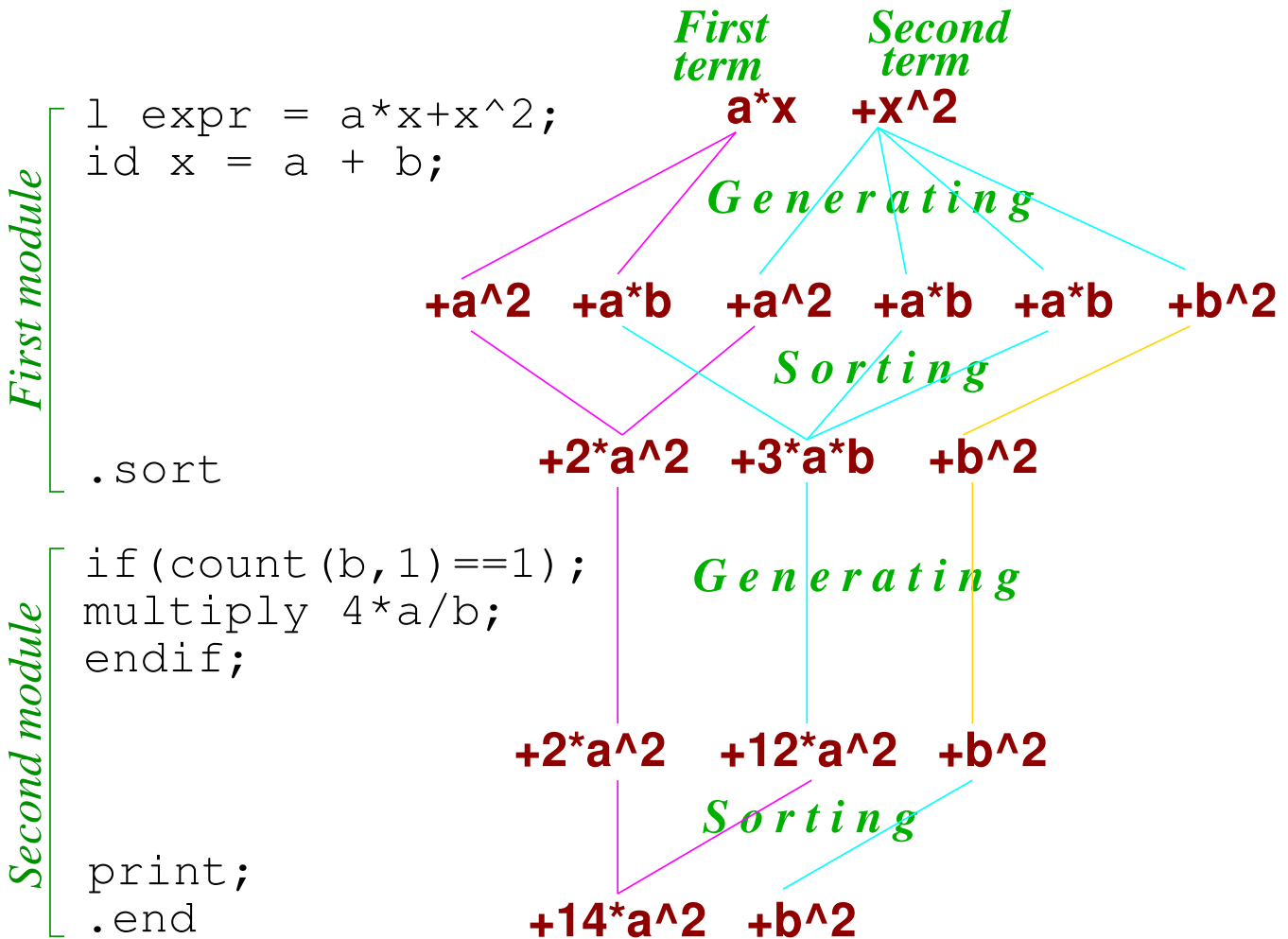
**FORM 3.1 features** support,

**Large file** (more than 2 Gbytes on 32-bit systems) support.

# The sequential version of FORM

Module by module. Each module:

1. Compilation.
2. Generating.
3. Sorting:



FORM specific feature: only **local** operations on single term:

`id x = a + b;`

Non-local operations are forbidden: ~~`id a + b = x;`~~



Expressions as “streams” of terms

⇒ { expressions bigger than the memory (RAM) available;  
allows **parallelism.**

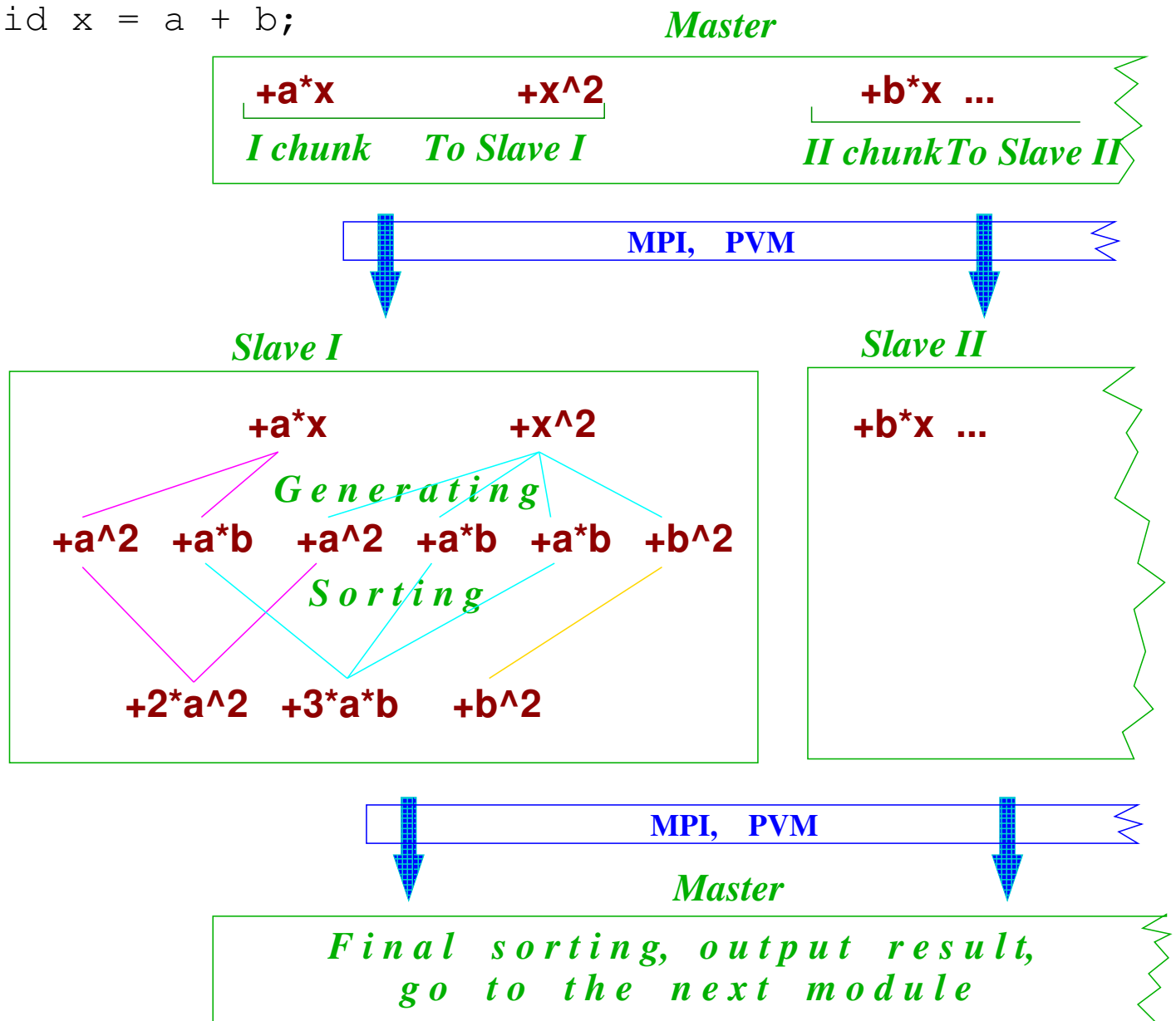
The concept of **parallelization** is straightforward:

hep-ph/9906426, hep-ph/0007221

a **master-slave** structure.

**Master** splits expressions into chunks. Each chunk is sent to one **slave**. Slaves generate terms, sort them and send sorted terms back to the master. Master performs final sorting:

```
l expr = a*x+x^2+b*x+...
id x = a + b;
```



## Working prototype of ParFORM (1999 – 2000)

is based on FORM version 3.-1 – a transitional version of FORM, syntax of FORM 3, but no FORM 3 features. **Many restrictions;** in particular, only one expression is allowed.

Two real physical applications,

**MINCER** and **BAICER**

A. Rètey, J.A.M. Vermaseren, P. Baikov and K. Chetyrkin.

Since 2000:

- **FORM version 3.1** is parallelized. In particular, \$-variable support.
- **Large file** (more than 2 Gbytes on 32-bit systems) support (as well as in the sequential version).
- **Restrictions** are reduced, now most of FORM 3 programs runs with ParFORM.

New **physical applications** available:

J.A.M. Vermaseren, **SUMMER**

The summer package contains a number of FORM procedures and tables for the manipulation of harmonic sums,  $\approx 5\,000$  statements.

S. Bekavac, application for summation of series in 3-loop nonplanar master integral.

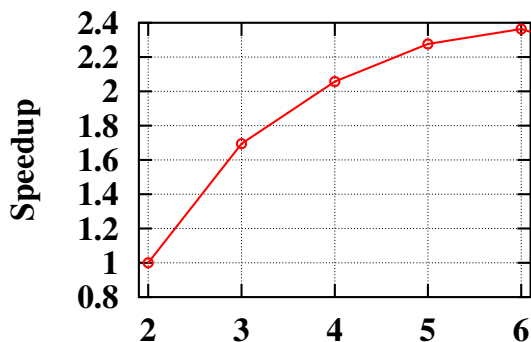
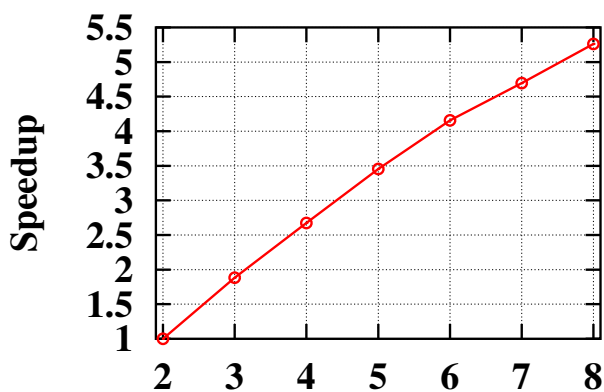
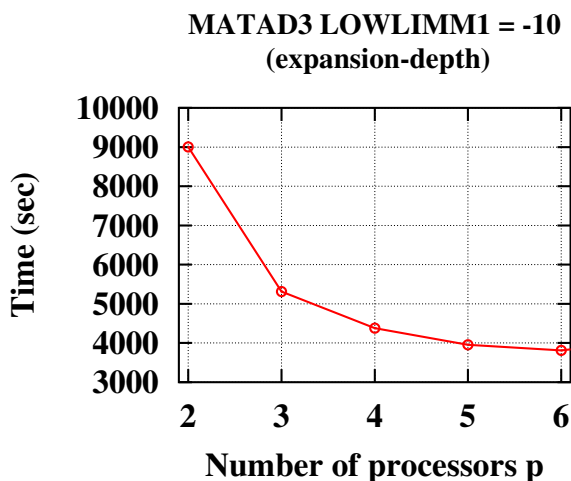
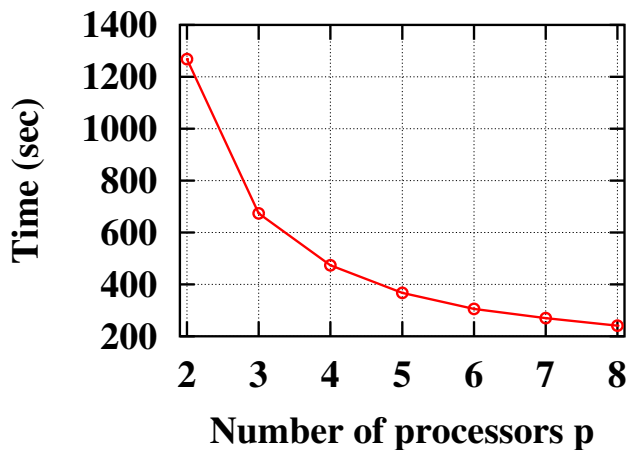
M. Steinhauser, **MATAD**

A FORM package for the computation of MAssive TADpoles. The method of integration-by-parts is used in order to obtain recurrence relations which reduce one-, two- and three-loop vacuum integrals with one non-zero mass parameter to master integrals,  $\approx 20\,000$  statements.

# TTP-SMP

8 Alpha (EV67) Processors (700 MHz), 8 MB Cache 8 GB RAM.

Summer



S. Bekavac, summation of harmonic sums with SUMMER.

M. Steinhauser, Matad3, 3-loop diagrams contribution to the fermion propagator for  $q \rightarrow 0$ , diagram "number 2".

ParFORM was ported to MATAD!

Now time to do the other way round?

Examples of the ParFORM problems.

Many small expressions vs. one big with a lot of terms.

```
l F1=a;
```

```
. . .
```

```
l F512=a;
```

```
id a=(a+b)^10;
```

```
id b=(a+c)^5;
```

```
id c=a+d;
```

```
s x,a,b,c,d;
```

```
#do i=1,512
```

```
l F'i'=a;
```

```
#enddo
```

```
id a=(a+b)^10;
```

```
.sort
```

```
id b=(a+c)^5;
```

```
.sort
```

```
id c=a+d;
```

```
.end
```

sequential: 90 sec

parallel: 59 sec (NP=6)

```
l F = a+a*e+...+a*e^511;
```

```
id a=(a+b)^10;
```

```
id b=(a+c)^5;
```

```
id c=a+d;
```

```
s x,a,b,c,d,e;
```

```
l F = a
```

```
#do i=2,512
```

```
  +a*e^'i'
```

```
#enddo
```

```
;
```

```
id a=(a+b)^10;
```

```
.sort
```

```
id b=(a+c)^5;
```

```
.sort
```

```
id c=a+d;
```

```
.end
```

sequential: 102 sec

parallel: 23 sec (NP=6)

## Load balancing problem.

Sometimes some terms may “explode”.

```
s x,a,b;
cf f;
l FF=f(18);

repeat;
  if(match(f?(0)) == 0) ;
    id f(x?)=a*f(x-1)+f(x-1);
  endif;
endrepeat;
print+s;
```

Real time is 12.8 sec.

| proc | CPU     | in | gen    | out |
|------|---------|----|--------|-----|
| 0    | : 12.42 | 1  | 0      | 19  |
| 1    | : 12.31 | 1  | 262144 | 19  |
| 2    | : 0.00  | 0  | 0      | 0   |
| 3    | : 0.00  | 0  | 0      | 0   |
| 4    | : 0.00  | 0  | 0      | 0   |
| 5    | : 0.00  | 0  | 0      | 0   |

```
s x,a,b;
cf f;
l FF=f(18);

#do i=1, 1
  id f(x?)=a*f(x-1)+f(x-1);
  if(match(f?(0)) == 0) ;
    redefine i "0";
  endif;
.sort
#enddo
print+s;
```

Real time is 0.4 sec!

```
FF =
+ f(0)
+ 18*f(0)*a
+ 153*f(0)*a^2
+ 816*f(0)*a^3
+ 3060*f(0)*a^4
+ 8568*f(0)*a^5
+ 18564*f(0)*a^6
+ 31824*f(0)*a^7
+ 43758*f(0)*a^8
+ 48620*f(0)*a^9
+ 43758*f(0)*a^10
+ 31824*f(0)*a^11
+ 18564*f(0)*a^12
+ 8568*f(0)*a^13
+ 3060*f(0)*a^14
+ 816*f(0)*a^15
+ 153*f(0)*a^16
+ 18*f(0)*a^17
+ f(0)*a^18
;
```

## Hardware.

Working prototype of ParFORM was developed using [TTP-SMP](#): our Compaq-AlphaServer GS60e, 8 Alpha (EV67) processors (700 MHz), bought in 1999.

Last year we were allowed by DFG and Universität Karlsruhe to buy a new 16-processor computer.

### Testing of possible hardware

A “standard” testing application: [BAICER](#) by P. Baikov:

[Solving recurrence relations.](#)

[Requires:](#) calculation of the large dimension of integration space expansion coefficients.

[Implementation:](#) linear reparametrizations of some multi-variable polinoms.

Mathematical complexity transforms to necessity to make simple manipulations with very large polinomial expressions (billions of terms and more).

FORM program runs for several days.

A “complexity parameter” [N](#), even number 4-12: expansion up to  $D^{-N}$ . The consumed time is proportional to approx.  $N^6$ .

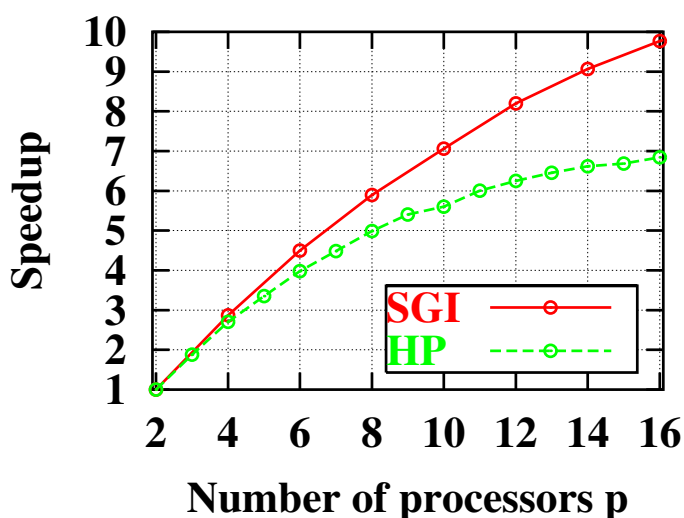
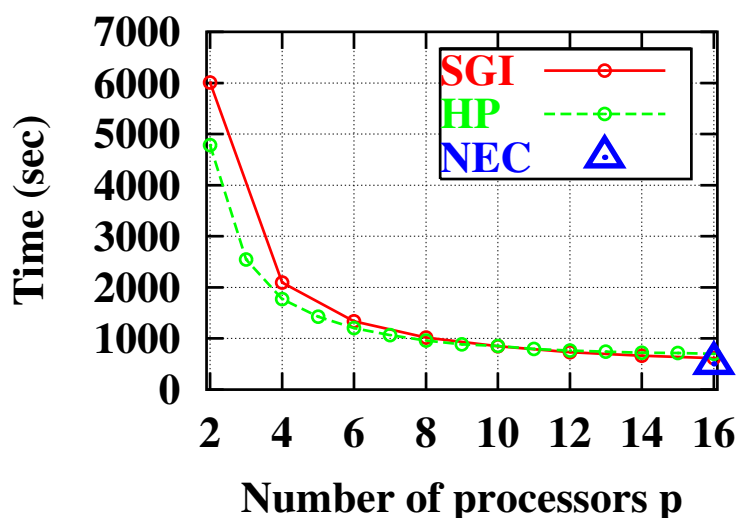


## Tested hardware

- Various IA32 Linux (2-way SMP and clusters),
- IBM RS/6000: 112 2-way SMP-nodes (POWER3-II, 375 MHz) with 2 GB RAM and 4 8-way SMP-nodes (POWER3-II, 375 MHz) with 8 GB RAM
- Alpha True64: TTP-SMP and Göttingen Compaq-AlphaServer GS60e 16-way Alpha 731 MHz, installed in 2000
- IA64 Linux ([SGI](#)) Altix 3000 16x1.3 GHz 32 GB RAM,
- IA64 HP-UX ([HP](#)) Orca 16x1.5 GHz 32 GB RAM,
- IA64 Linux ([NEC](#)) TX7 16x1.5 GHz 16 GB RAM.

Negotiations: final stage with 3 companies (SGI, HP and NEC):

## [BAICER](#) N=12



During negotiation: [32-way](#) computer for the same money!  
(From two companies).

Decision: To orged [SGI](#), installed in Feb. 04, First ParFORM-BAICER results: 15 Feb.

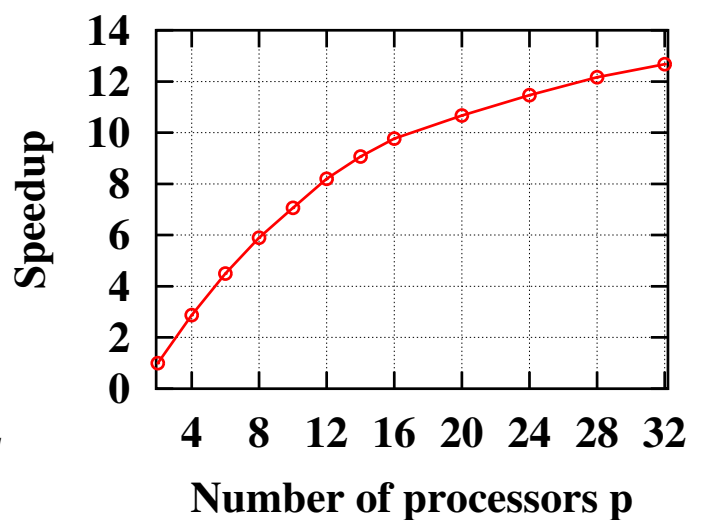
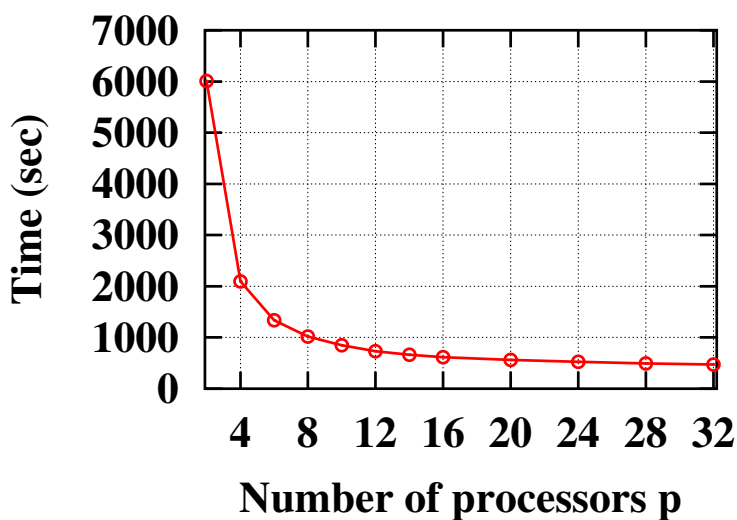
## SGI Altix 3700 Server

32x1.3 GHz/3Mb-SC IA64

CPUs, 64 GB DDR/166 MHz mem

2x18GB + 16x146GB SCSI disks ( $\approx$  2.1 TByte).

**BAICER** N=12



|               | Time (sec) | Speedup |
|---------------|------------|---------|
| 2 processors  | 6009       | —       |
| 16 processors | 615        | 9.8     |
| 32 processors | 474        | 12.6    |