Pyrros+: Automatic parallelization and performance prediction tool

- Apostolos Gerasoulis
- Rutgers University, USA
  www.cs.rutgers.edu/~gerasoul

Colaborators

- **Tao Yang, UCSB**
- **Michel Cosnard, Michel Loi, France**
- **Jia Jiao, Bell Labs- Lucent**
- **David Rhodes, US Army CECOM/RDEC.**
- **Rutgers Students:**
  - Chengli Wang
  - Yingfang Lu
  - Baohua Wu

Supported

- DARPA(Hpcd, Prediction)
- NSF-INRIA (travel)
- SUN Microsystems(22 Ultraspark)
- HP(40 Kayak)
- Rutgers University(HPC Lab)
- Rutgers University(5 New system hires)

Research area

- **Static & Dynamic Scheduling**
  - Program information at compile and run time, e.g. computation and communication
  - Processor information at compile and run time, e.g. processor load
- **Scheduling Tools**
  - PYRROS and D-PYRROS : Static and dynamic scheduling systems
  - Pyrros+: Automatic parallelization and scheduling

Goals

- **Predict Program Performance:**
  Pyrros+
- **Use Prediction to improve the performance of parallel programs :**
  Examples:
  Harmonic Balance(Rhodes),
  Ship Design(SAIC),
  Linear Algebra(TaoYang),
  Nbody Appl. (Gerasoulis)

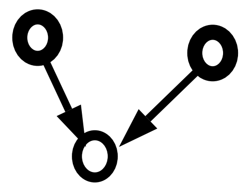
Program Prediction Range

ANALYTIC
STATISTICAL

Tp= n^3w/p + n^2 b

Isoeficiency
Hint

EMULATORS



PYRROS+

EXACT

LINPACK PERFORMANCE

MORE EXPENSIVE -----HIGHER ACCURACY →

← LESS EXPENSIVE -----LESS ACCURACY

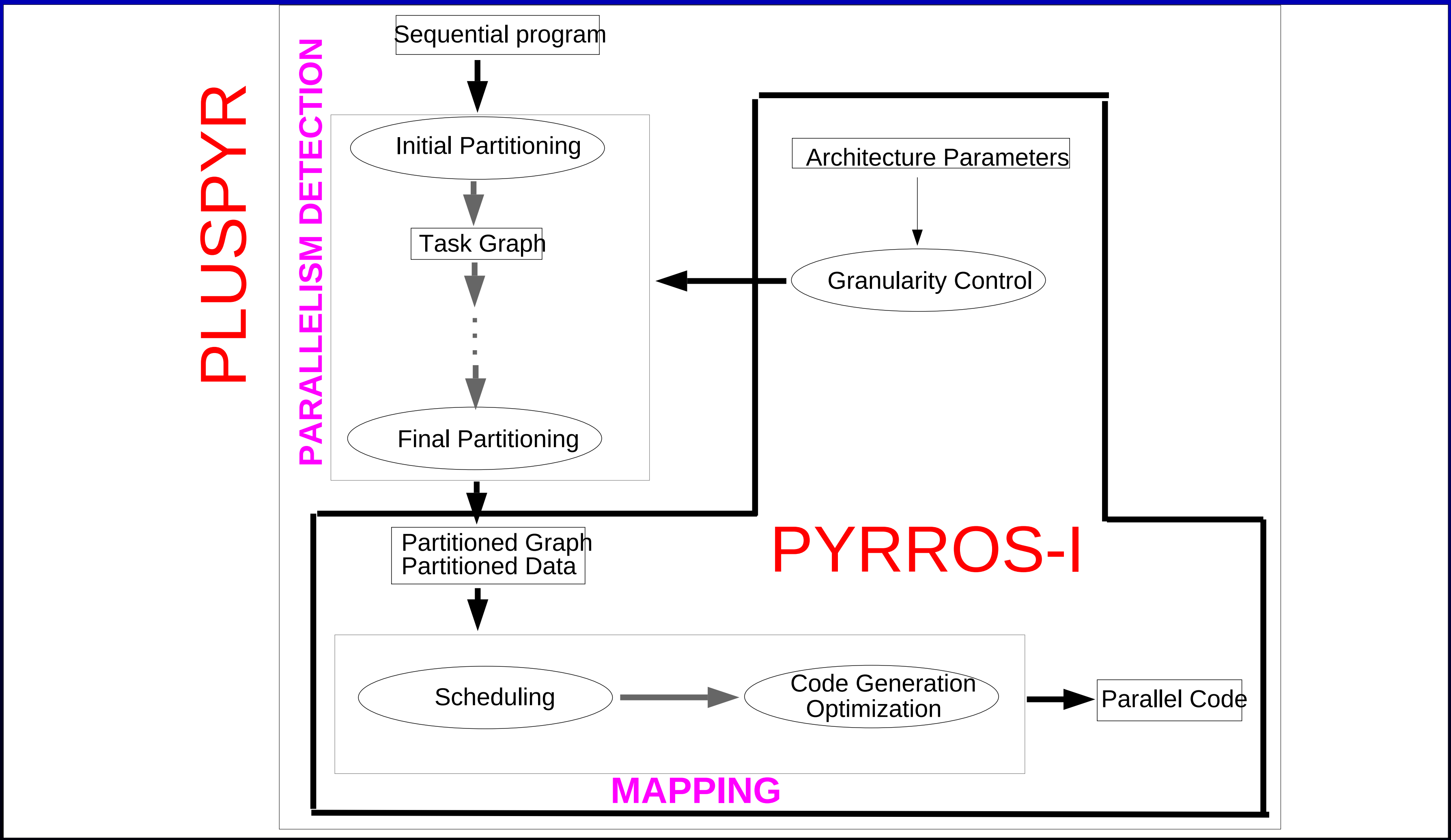
Pyrros+

- INPUT: Sequential program with task anotations
- OUTPUT: Parallel Program using PYRROS scheduler
- PREDICTS performance using Program information, processor and network speed estimation.
- INOVATION: First task based automatic parallelization and scheduling system

Pyrros+ Technology

- Omega Test for Fine grain analysis.
- Macrodataflow model for coarse grain analysis
- Pyrros static scheduling
- Pyrros code generation in MPI.
- Graphical user interface

Pyrros+ architecture



Pyrros+ Example

```
param n
    real a(n, n)
    for j = 1 to n do
     for i =  1 to n do
     task
      a(i,j)= a(i,j-1)+1
     endtask
      endfor
     endfor
```

Pyrros+ Interface

- A loop example at statement level partitioning
- Partitioning at interior loop
- Loop interchange to create parallelism at coarser grain

Pyrros+ estimation model

- **Computation**: number of operations times the cost per operation
- **Communication**: Linear model- a + size*b, where a is processor overhead and b is transmission rate
- **Scheduling**: Macrodatflow and Pyrros scheduling.

Pyrros Performance

- **Gauss Jordan algorithm**

Machine	Predicted time		Actual time	Seq. time
SP-2	P=2	19.37(s)	20.75(s)	23
	P=4	11.86	14.96	
MYRINET& ULTRA-1 167MHZ	P=2	24.74	23.50	53
	P=4	17.32	15.75	
NCUBE-2S	P=2	625(s) (*)	581	1152(*)
	P=4	328	336	
	P=8	184	200	(*) Estimated
PROBLEM SIZE	n=1000 , N=50, r=10			

Pyrros+ Performance

- **Gauss Elimination algorithm**

Machines	Actual time	Speedup	nproc
16 SUN ULTRA 167Mhz + 300Mhz. 1.2 Gbit Myrinet MPI MPICH TCP/IP MPE ON	31.7(sec) 15 8.7 6.7 8.6	2 (est) 4 7 9 7	2 4 8 12 16
PROBLEM SIZE	n=1200 , N=30, r=40		

Can Pyrros+ Predict Performance?

- **Gauss Elimination algorithm**

Machines	Predicted	Actual	nproc
22 SUN ULTRA	90(sec)	88.76 (est)	1
167Mhz .	50.5	50	2(**)
100 Mbit ethernet	38.5	37	3
MPI MPICH	32.5	30.6	4
TCP/IP	27.6	26.7	5
MPE OF	22.3	19.7	8
??????????	16	26.6	12
PROBLEM SIZE	n=1600 , N=40, r=40		

(**)Parameters where estimated from nproc=2
Comm. Load, OS overhead was set =0

David Rhodes Application

- Pyrros+ :Going Beyond the automatic parallelization restrictions!!!
- Problem: Parallelize the Harmonic Balance Simulation equation
- Goal: Achieve high speedups; e. g. 20 frequencies run in parallel gives 20 speedup; can we go beyond 20 speedup?

David Rhodes Application

- Each frequency is a Tree task graph!
- David Estimated the weighted task graph(169 nodes) for the ARMY CRAY T3E
- Pyrros+ Predicted a maximum spedup of 69 using 120 processors! It also proposed a scheduling to achieve this.
- WORK IN PROGRESS

Why Access Patern is important?

- **Sequential Gauss Jordan**

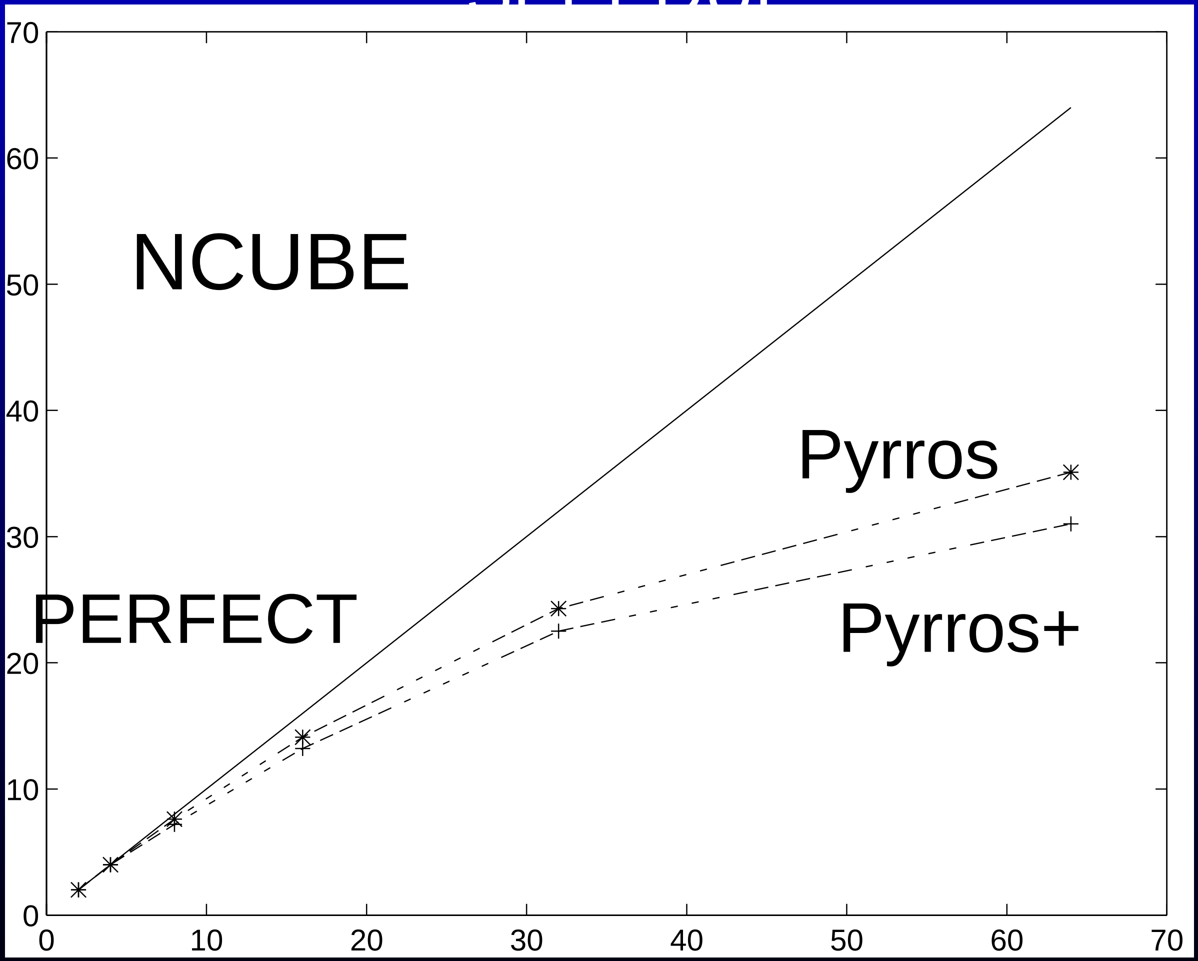
Machines	row major	column major
RS/6000(SP2)	514(sec)	23.3(s)
Ultraspark 140	175	53.9
Ncube2S	924	913
PROBLEM SIZE	n=1000	

What improvements are needed?

- Parameter estimation needs to be expanded to include processor and communication load statistics
- Data access paterns need to be added to the operation count statistics
- Alternative Scheduling algorithms need to be tested

Pyrros vs. Pyrros+

SPEEDUP

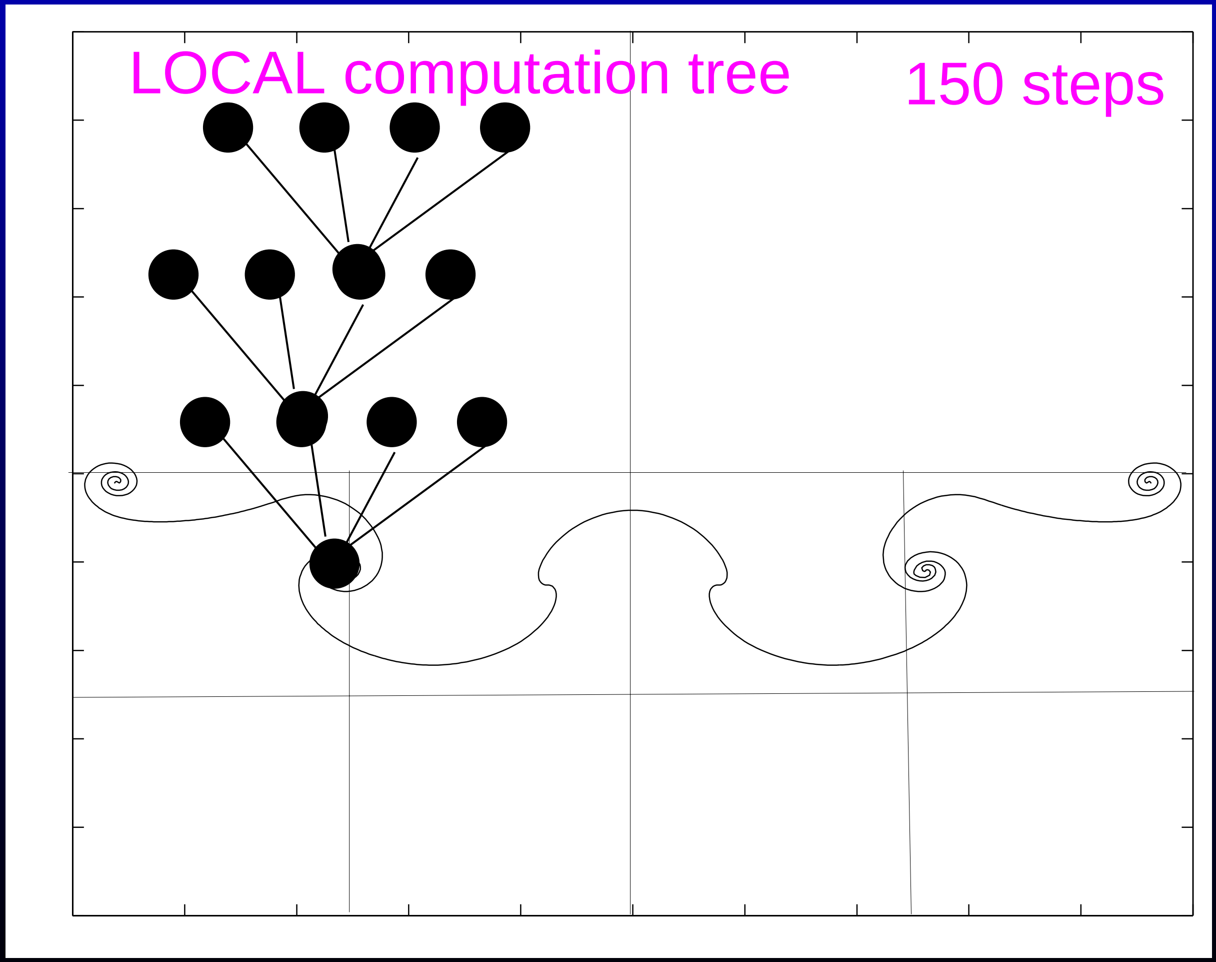


n=1000, N=1000/10=100, r=10

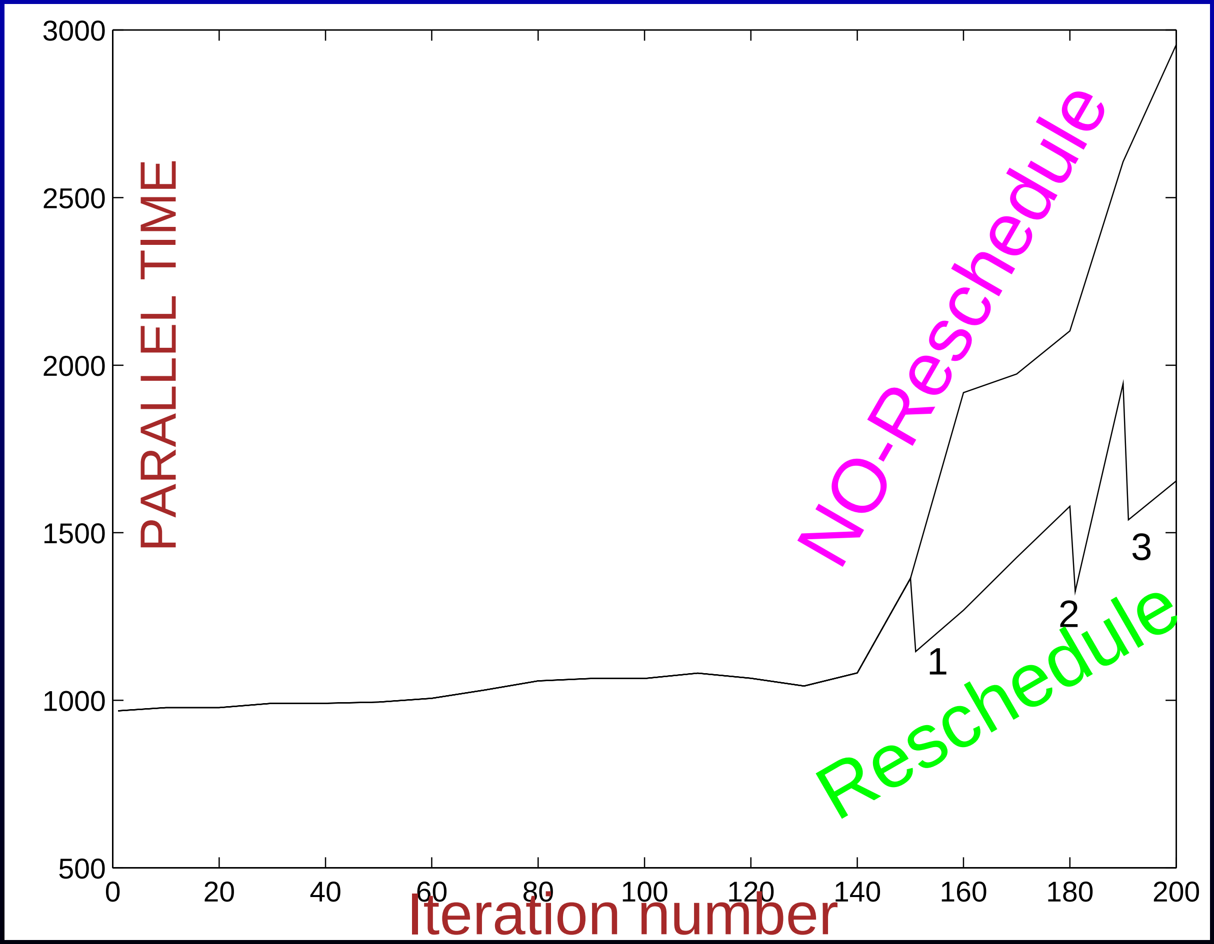
D-PYRROS

- **Innovations**
  - Incremental scheduling
  - Run time program computation and communication estimation for scheduling.
  - Local scheduling to lower scheduling overhead.
  - Local or global clustering(DSC) at run time to reduce high communication cost.
  - Re-scheduling(re-mapping) of computation based on run time multiprocessor performance deterioration.
  - Excellent performance for "slowly changing dynamic problems".
- **New in 98: MPI parallel code.**

2D vortex dynamics



D-PYRROS Performance



Plans 1997-2000

97

- **1997-1998: Pyrros+ prototype.**
  - Integrated GUI and Automatic Parallelization, Integrated Pyrros and MPI . First experimental results.
- **1998-1999: D-Pyrros and Pyrros+**
  - Incorporate dynamic scheduling.Incorporate Access Paterns in computation estimation, Incorporate Computation and communication load. OUT of CORE application emulators
- **1999-2000 Irregular Pyrros+**
  - Integrate RAPID with Pyrros+, Symbolic scheduling, Large scale irregular applications