PUMP–AND–TREAT GROUNDWATER REMEDIATION ANALYSIS USING THE INTEL PARAGON

Kumar G. Mahinthakumar
Center for Computational Sciences
Oak Ridge National Laboratory

OUTLINE

- Introduction
- The Computer Architectures
- The Groundwater flow problem
- The Solute transport problem
- Conclusions

Practical Example

Measured Plume for Chloride at the BORDEN Aquifer

3 Stages in Simulation

- Generate hydraulic conductivity field of aquifer using a random field generator
- Solve the flow problem using a direct method (finite dfference/finite element) –> velocity field
- Solve the contaminant transport problem using either a particle method or a direct method –> conatminant distribution, concentration moments (plume statistics), travel time distribution etc.

Motivation for supercomputers

- Flow problem: Large scale effects of small scale aquifer heterogenity (macro dispersion) –> large 3D fields, fine discretization –> large matrices

- Particle tracking transport problem: accurate statistics –> Large number of particles required (especially for 3D)

- Many simulations may be required due to uncertainties in the hydraulic conductivity fields (in a MonteCarlo sense)

Performance, Algorithm and Architecture: a simple example

- Jacobi: a simple iterative scheme which uses all previous iterate values of x

$$ {}^{(k+1)}_{i} = \left[b_i - \sum_{=1}^{i-1} a_i \;{}^{(k)} - \sum_{=i+1}^{n} a_i \;{}^{(k)} \right] \Big/ a_{ii} \qquad i = 1:n $$

- Gauss–Seidel: identical to Jacobi but uses most current updates of x (faster convergence, but data dependency in loop)

$$ {}^{(k+1)}_{i} = \left[b_i - \sum_{=1}^{i-1} a_i \;{}^{(k+1)} - \sum_{=i+1}^{n} a_i \;{}^{(k)} \right] \Big/ a_{ii} \qquad i = 1:n $$

- test matrix: a simple tridiagonal system

M	I	Y−MP PU		HP−U	
		PU S	M	PU S	M
G−S					

The computer architectures

P orstatio

- 32–64 MB Memory
- PA–RISC processor at 50 MHz
- Peak performance: 15 Mflops

Cra MP

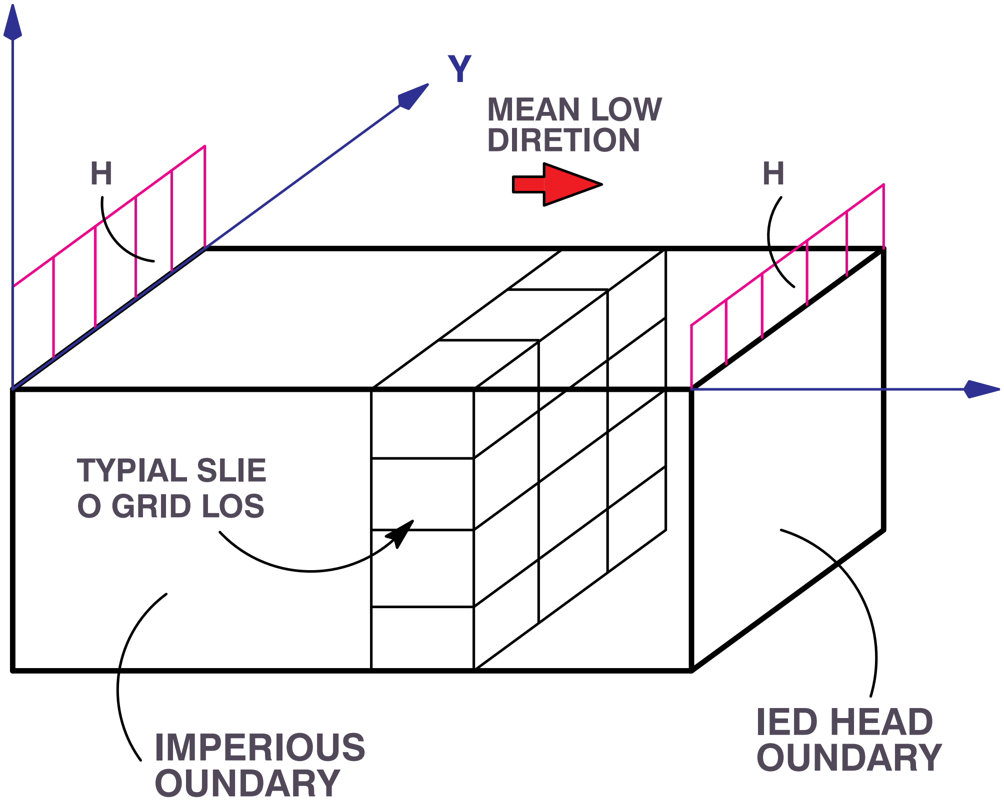
- 8 vector processors
- 64 Mw memory/processor
- 6.2 ns clock speed
- Peak performance: 333 Mflops/processor

Coetio Mahie CM

- massively data parallel architecture
- 8K, 16K, 32K, and 64K processor configurations
- 64–256 Kb local memory/processor
- 8 MHz/processor
- Peak performance: 1.5 Gflops/ 64K processors

The Groundwater Flow Problem

- Schematic Representation:



- Uniform 3D finite difference grid
- Natural gradient conditions

Equations

- The flow equation:

$$(\; K \;\; h) \;\; =$$

- Velocity given by Darcy’s law:

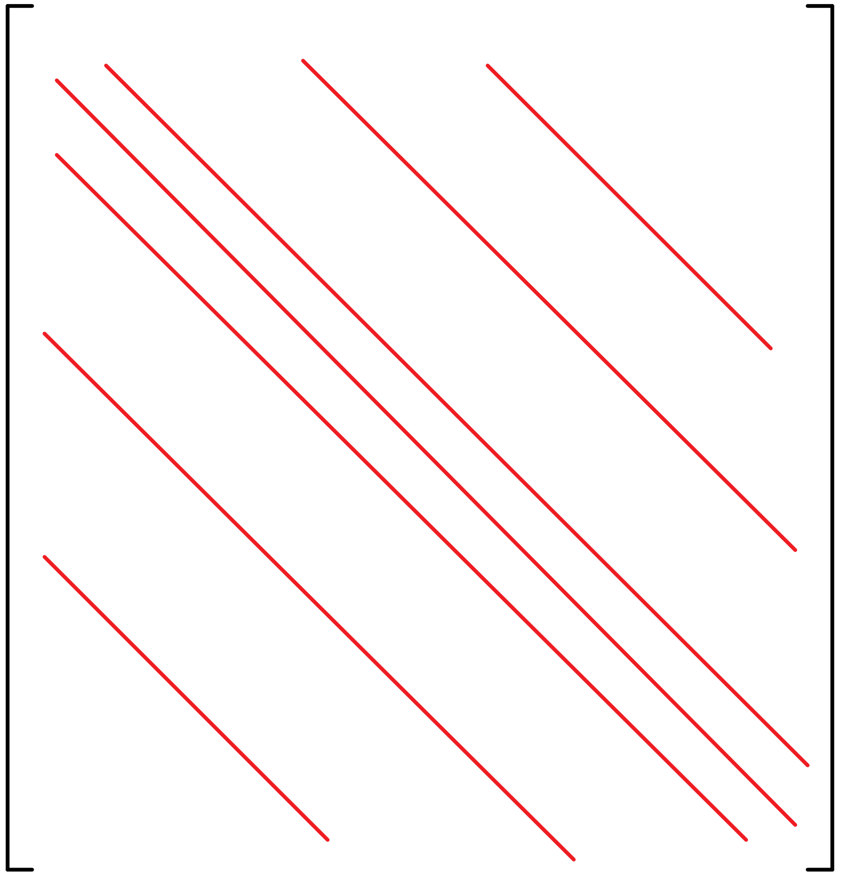
$$\boldsymbol{q} \; = \theta \, \boldsymbol{v} \; = \; - K \;\; h$$

- After finite difference approximation, the matrix equation:

A

Resulting Matrix A

- **7–diagonal symmetric matrix**



Matrix Solution Scheme

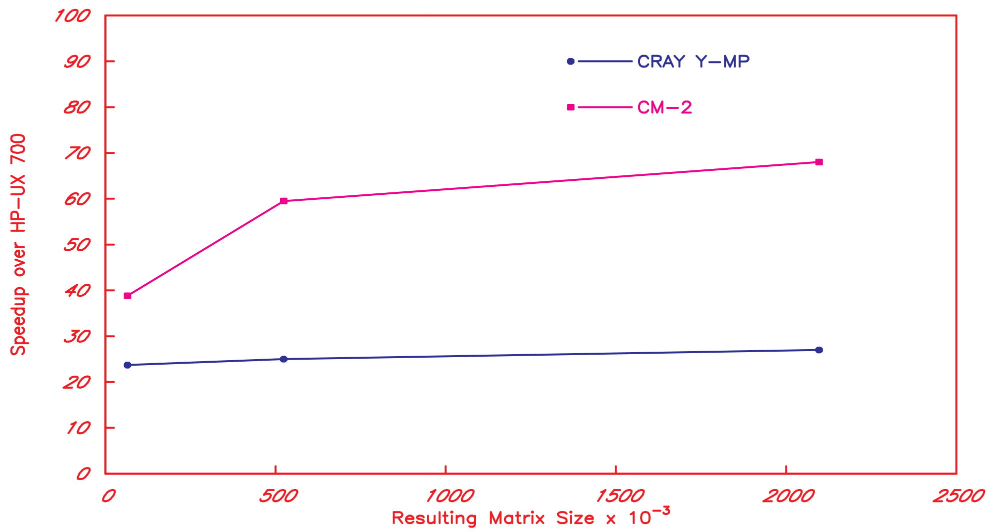
- The Diagonally Preconditioned Conjugate Gradient Algorithm (DPCG)
- Popular for large, sparse, symmetric and positive definite linear systems
- Simple scalar, vector, and matrix operations involving large arrays
- Easily vectorizable/parallelizable internal loops

Results: Flow Problem

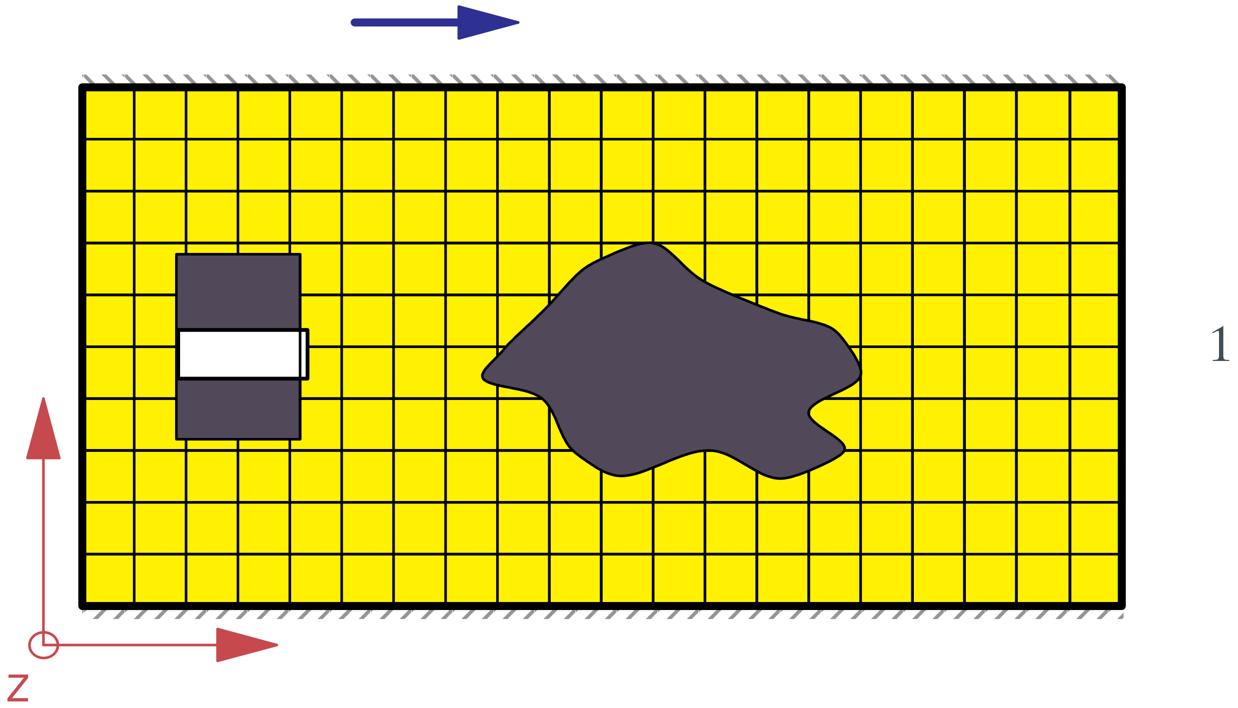
imi Performae for differet rolem sies

P S	M S	G I	HP−U		Y−MP		M−	
			PU S	M	PU S	M	PU S	M
			–	–				

eedus o CM ad MP oer P



The Transport Problem



- fixed cubic input zone
- velocity field obtained from flow solution for heterogeneous case

The Particle Tracking Method

- **Original PDE's**

$$\frac{c}{t} + \quad (c\,\mathbf{v} \quad - \quad (\qquad c) + \frac{}{\theta}\,\frac{S}{t} =$$

$$\frac{S}{t} = k_r(\ K_d\, c - S)$$

- **Particle Tracking Analog**

$$_p(t + \quad t) = \quad _p(t) + \mathbf{v}(\ _p) \quad t + \qquad \overline{\quad t}$$

- **Simplified**

$$_p(t + \quad t) = \quad _p(t) + \mathbf{v}(\ _p) \quad t$$

Timing Results: Transport Problem

or artile trai

- velocity field taken from 128 x 64 x 64 flow solution
- 100 time steps used for all tests

NP	HP–U		Y–MP		M–	
	N–R	R	N–R	R	N–R	R

L

NP

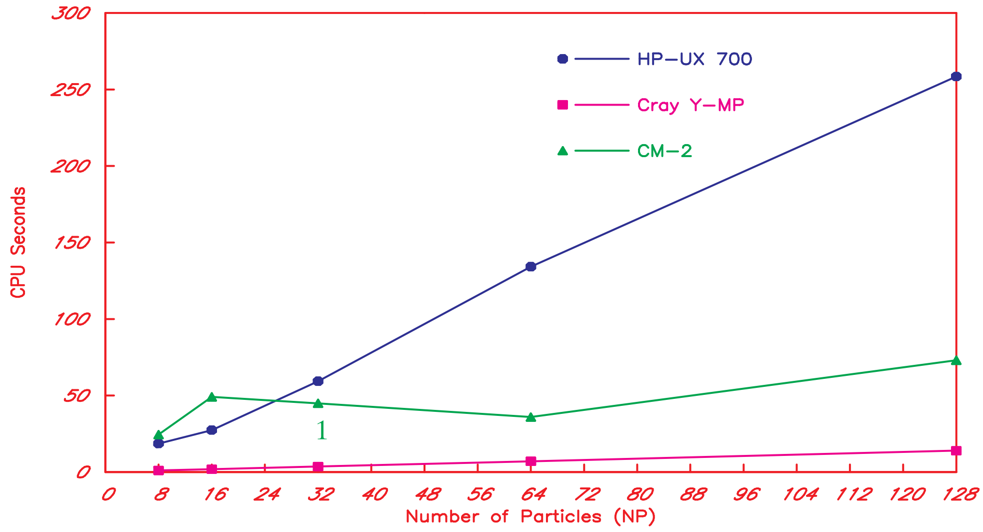
N–R –

R

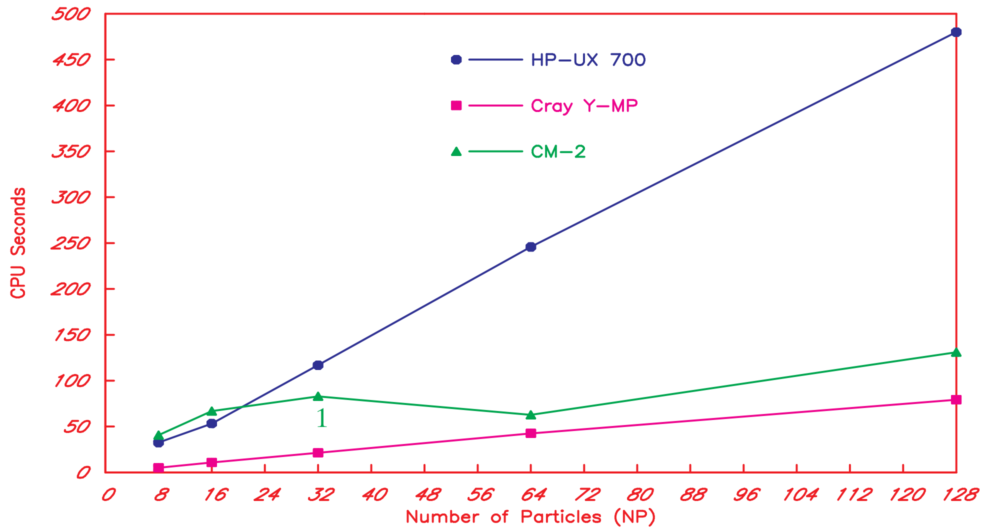
M

Timing Results for Particle Tracking

T N–R P T



T R P T



Conclusions

- The groundwater flow problem using finite–differences and a conjugate gradient solver performs well on both vector and massively parallel supercomputers. However, smaller problems could still be very efficiently solved using the lab workstations.
- The particle tracking problem does not perform well on these computers (particularly on the connection machine due to inter–processor communication).
- The particle tracking method is very well suited for a scalar computer like the HP–UX 700. Performance degradation due to the unconventional operations in this method is minimal for these types of computers.
- A direct method such a the finite element or finite difference method looks more promising for the transport problem on the supercomputers. A conjugate gradient like iterative technique (eg: GMRES) can be used for the non–symmetric matrix arising in the transport problem.