Student: Vidar Tulinius

Email: vidarot@brandeis.edu

Distributed frameworks and parallel algorithms for processing large-scale geographic data

Kenneth A. Hawick

P. D. Coddington

H. A. James

Overview

- Increasing power
  - PC
  - Workstations
  - Mono-processors
- Geographic information systems (GIS) provide a resource-hungry application domain.
- Environmental and defense applications need to deal with really large data sets.

Outline

- Metacomputing and grid computing
- Metacomputing middleware
  - DISCWorld
- Transparent parallelism
- Image processing applications
- Parallel spatial data interpolation
  - Kriging interpolation
- Spatial data grids
- Conclusion

Parallel techniques

- Satellite imagery
  - Increased size
  - Improved resolution
  - More frequency channels
- Data stored as lists of objects
  - Locations
  - Vectors for roads
- Objects allocated for processing among processors in a load balanced fashion

Metacomputing and grid computing

- Number of resources distributed over a network.
  - High-performance computers
  - Clusters of workstations
  - Disk arrays
  - Tape silos
  - Data archives
  - Data visualization
  - Scientific instruments

Metacomputing and grid computing

- GIS applications adhere well to grid computing
  - Require access to multiple databases
  - Large data sets
  - Require significant computational resources.

Metacomputing and grid computing cont.

- ‘bleeding-edge’ technology
- Globus Grid Toolkit
  - Low-level
  - Hard to use
  - Requires substantial technical expertise
- OLDA
  - developed a prototype computational grids.
    - Distributed data archives
    - Distributed computational resources

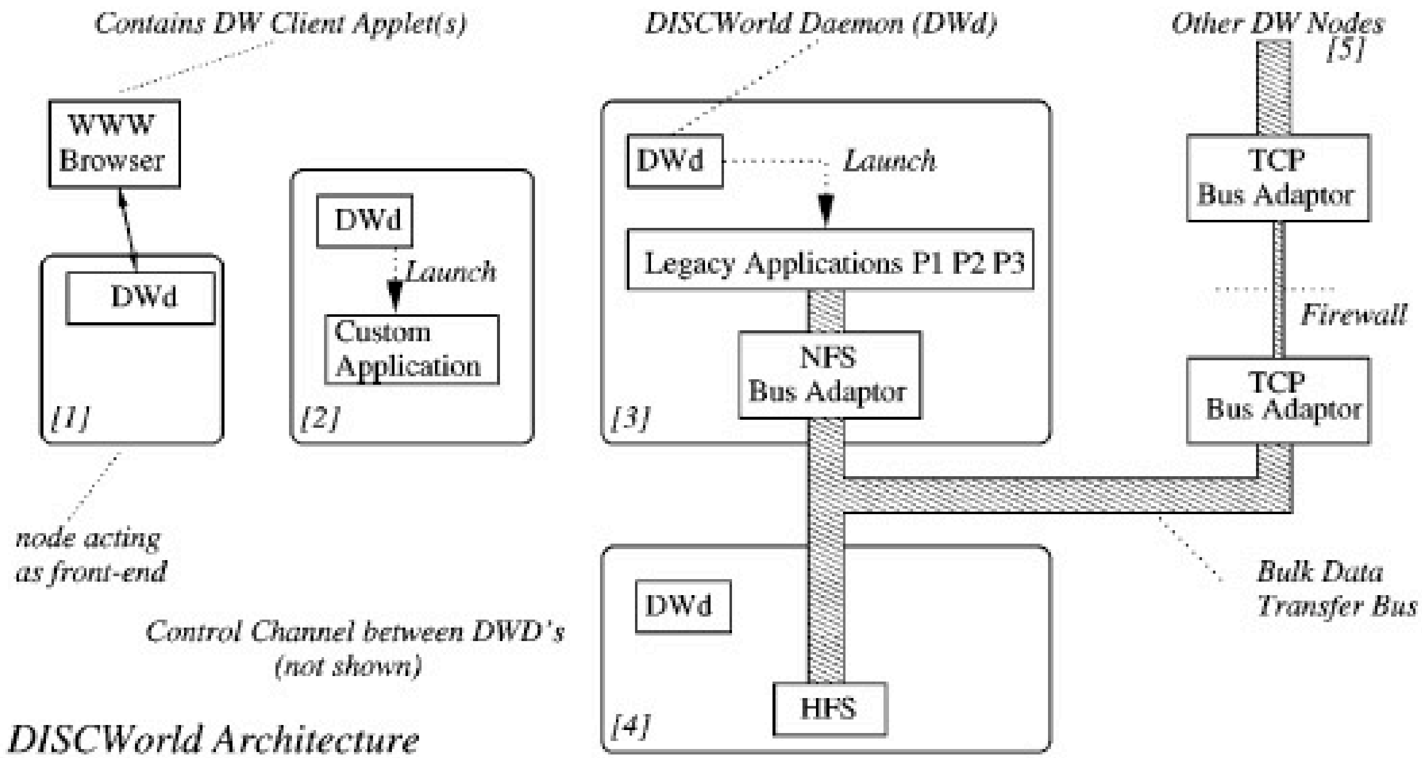
Metacomputing middleware

- Client applications should not require end user to be an expert in data access.
- Middleware should handle problems when searching for data
- Metacomputing systems are mainly targeted at scientific simulations.

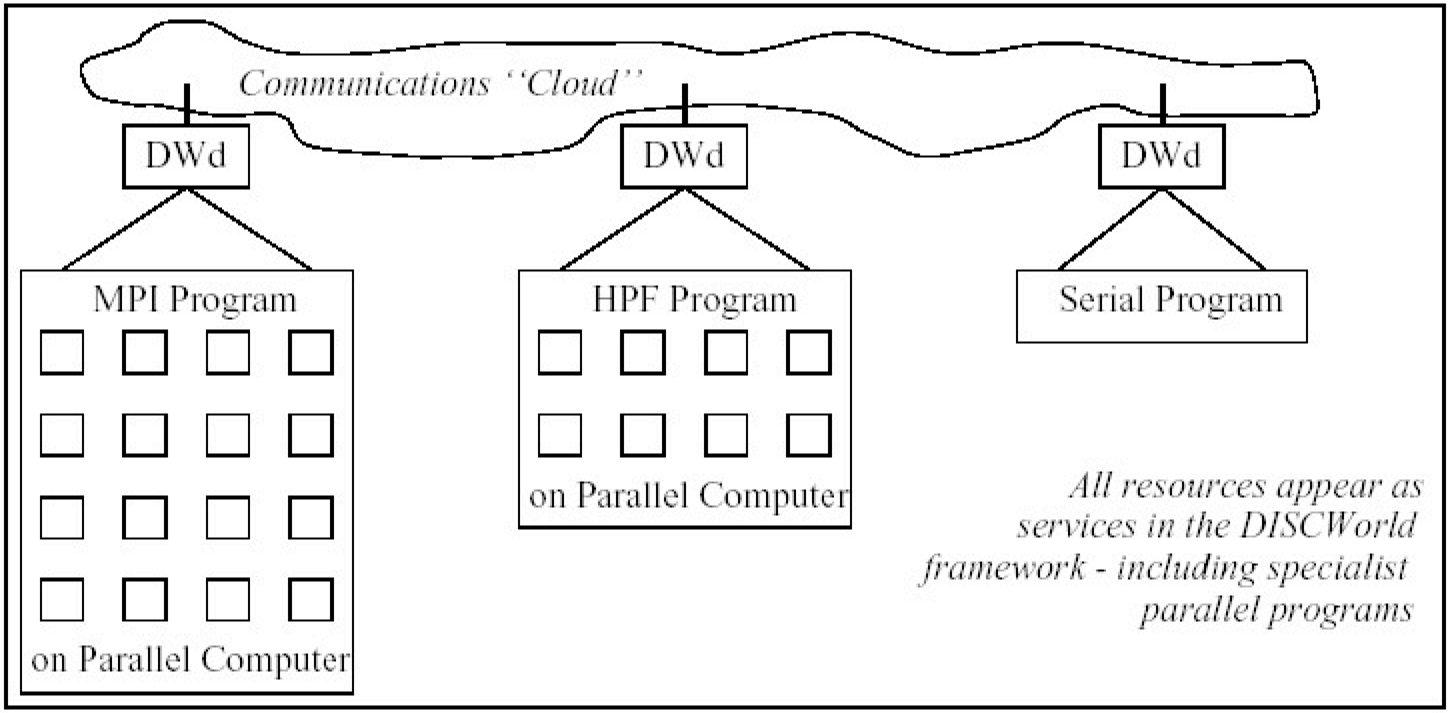
DISCWorld

- Prototype metacomputing middleware developed using Java
- Provides support for applications such as decision support systems.
- Services
  - Prewritten pieces of Java code
  - Parallelism is transparent
- Daemon
  - Interoperability mechanisms with legacy applications and data systems
  - Need for scalability to a large number of cooperating nodes

K.A. Hawick et al. / Parallel Computing 29 (2003) 1297–1333



The DISCWorld software architecture overview.



DISCWorld daemons acting as front-end network for linking parallel programs and other specialists services through communications cloud.

DISCWorld

- Summarization of events
  1. The user connects to the system.
  2. A domain appropriate interface is downloaded on-demand.
  3. The interface guides the user in submitting a valid query.
  4. The responsible node organizes analysis and execution options for the query.
  5. The node invokes and monitors execution of the service components.
  6. The system executes the query, allowing revocation and dynamic rescheduling.
  7. Results are delivered to the user.

Daemons

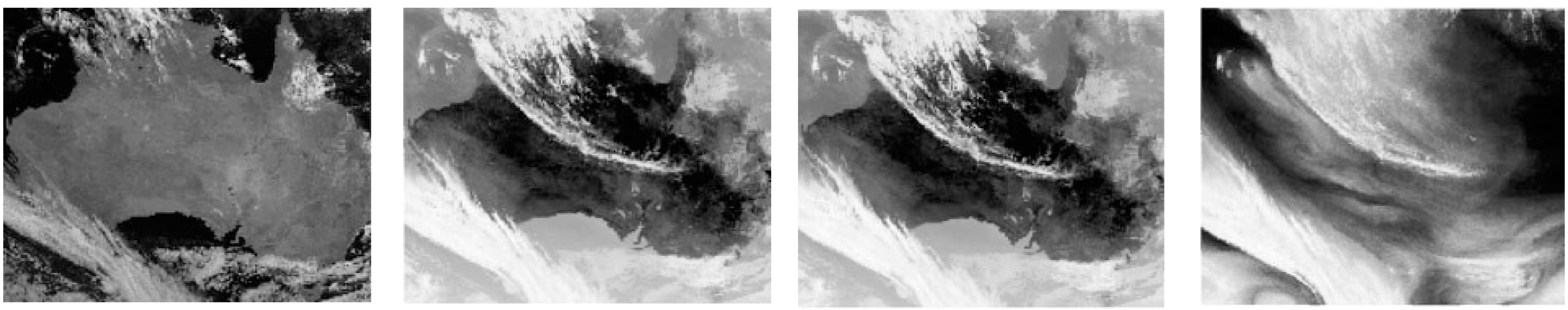
- Initiated by system admin or as a system daemon
  - Configures itself and listens for a user request
- Each daemon has a ‘gossip’ mechanism
- Each daemon stores its own internal database
- Daemons are highly configurable
  - Can be customized to reflect the local server or services it provides
  - Has a configurable policy where to look for help

Transparent parallelism

- Not ideal for an user to be aware of the parallel engine in their system
- How to embed the parallel engine in the system
- ERIC
  - Provides an infrastructure to embed parallel processing technologies
  - Web browser interface to a CGI program running on a web server
  - Uses remote execution in the form of distributed *rsh* invocations to spread the processing to a cluster of workstations

Image processing applic.

- GMS-5 satellite data archive
  - Stored on a RAID disk array and tape silo
  - Produces images once an hour
  - 10,000 x 10,000 pixel image not uncommon
  - File are stored in Hierarchical Data Format



(a) (b) (c) (d)

Multi-spectral data cropped from GMS-5 worldview data: (a) visible spectra, (b) thermal IR1 spectra, (c) thermal IR2 spectra and (d) water vapour IR spectra.

Image processing applic.

- Image filtering and scanline corrections
  - Satellite imagery is generated as a set of scanlines
  - For a single dropout line it is possible to interpolate lines above and below to acceptable accuracy.
  - Fourier Filter can be used for multiple dropout line errors
  - Parallel algorithms are currently being developed and experimented.

Image processing applic.

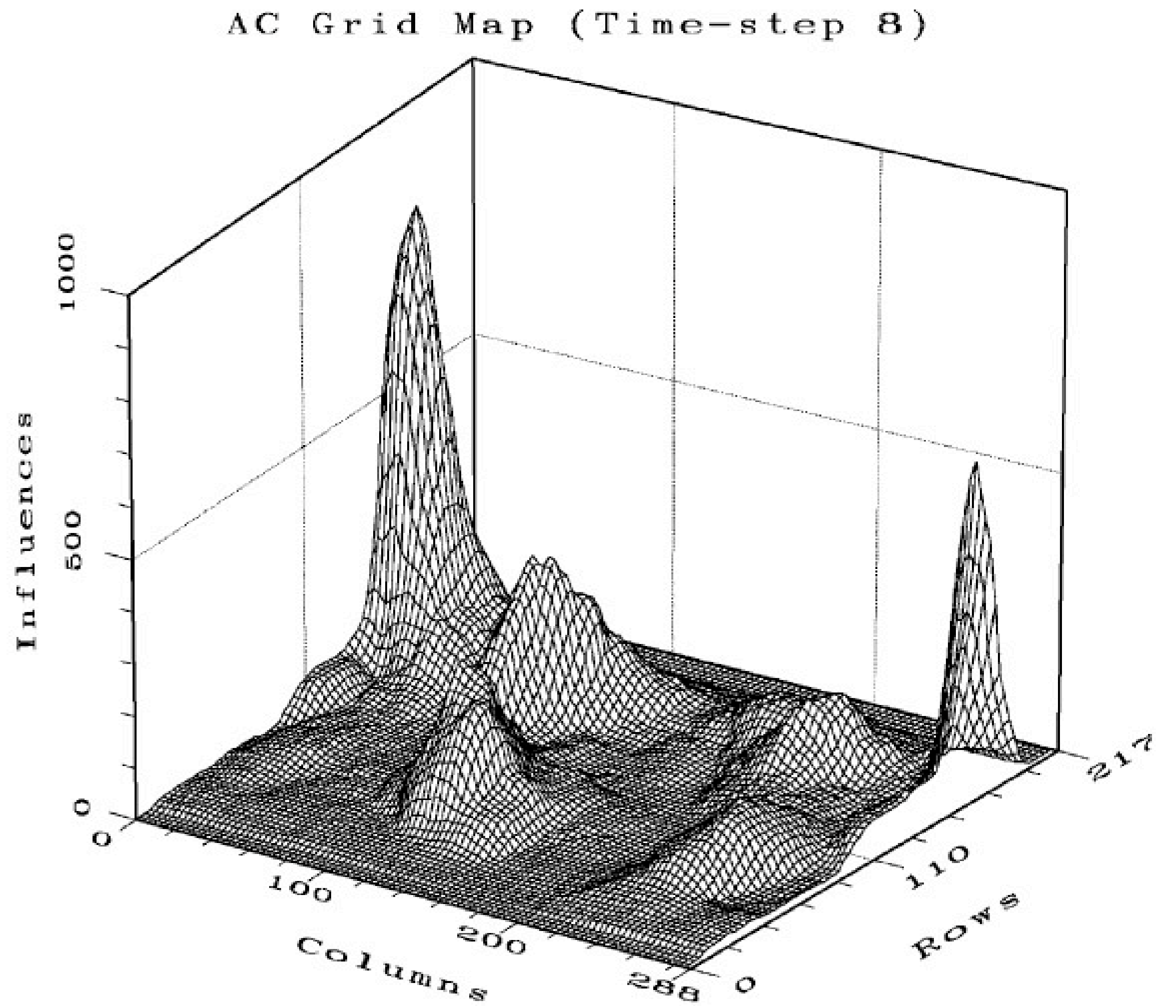
- Coordinate transformations and georectification
  - Images can be transformed from raw pixel co-ordinates to a mapping onto the Earth's surface.
  - Georectification is a computationally intensive task.
  - Two approaches
    - Image rectified pixel by pixel to map it onto longitude and latitude.
    - Warp the image to align with a desired map of earth.

Parallel spatial data interpolation

- GIS application use regular 2D or 3D grid, such as pixels in a satellite image
- What's the problem?
  - Data needed as input for an application may not be co-located with grid coordinates used in the computation
  - Highly irregular spatial distribution
  - Interpolation is needed to align the data
- The Unified Model (UM)
  - Based around rectangular grid based model
  - Used for numerical weather/climate prediction
  - Three main components
    - Atmospheric dynamics
    - Physics calculations
    - Assimilation of observational data

Parallel spatial data interpolation

- The assimilation of observational data is by no means trivial.
- Analysis Correction scheme.
  - Interpolation of observed data.
    - Surface measurements
    - Aircraft reports
    - Satellite cloud vector data
    - Satellite sounding data



- Inhomogeneity of observation influences per grid point in space. The major peak corresponds to the location of the UK Meteorological Office in Bracknell, England

Kriging interpolation

- Fusion of large remotely sensed data sets with directly observed data for ground truthing is an important and computationally demanding problem.
- Satellite imagery data has a regular grid pattern completely different from the distribution of ground truth data.
- Data fusion requires interpolation of the irregularly spaced ground truth data with the regular grid of the satellite image pixels.

Kriging interpolation

Location points for Rainman data collection.

- "Rainman" is a data set for ground thruthing data.
- Measurements of rainfall for the whole of Australia.
- The rainfall data is clustered around cities and more populated regions.
- This problem is well suited to Kriging interpolation.

Kriging interpolation

- Kriging is formulated as a matrix problem connecting the data sets.
  - Weightings computed using standard matrix solver techniques such as LU factorization.
- Kriging produces the best results when the largest possible number of known points is used in estimating each unknown point
- Kriging large data set is to computationally demanding
  - Needs to be run on a separate “accelerator platform”

Kriging interpolation

- We have a set of *K* known points P
- Each point P_i is of the form (x_i, y_i, Z_i)
- We then estimate the value of an unknown point E_{ij} by calculating the weighted sum of the know points

$$E(x_i, y_j) = \sum_{k=K}^{k=1} w_k Z_k$$

- W_k is the weighting given to the K^th known point

Kriging interpolation

- A separate set of weights W_{ij} must be calculated for each estimation
- We first construct a variogram V of the known points where each element D_{ij} is the distance from the known point P_i to the known point P_j

$$V = \begin{bmatrix} D_{11} & D_{12} & \dots & D_{1K} \\ D_{21} & D_{22} & \dots & D_{2K} \\ \dots & \dots & \dots & \dots \\ D_{K1} & D_{K2} & \dots & D_{KK} \end{bmatrix}$$

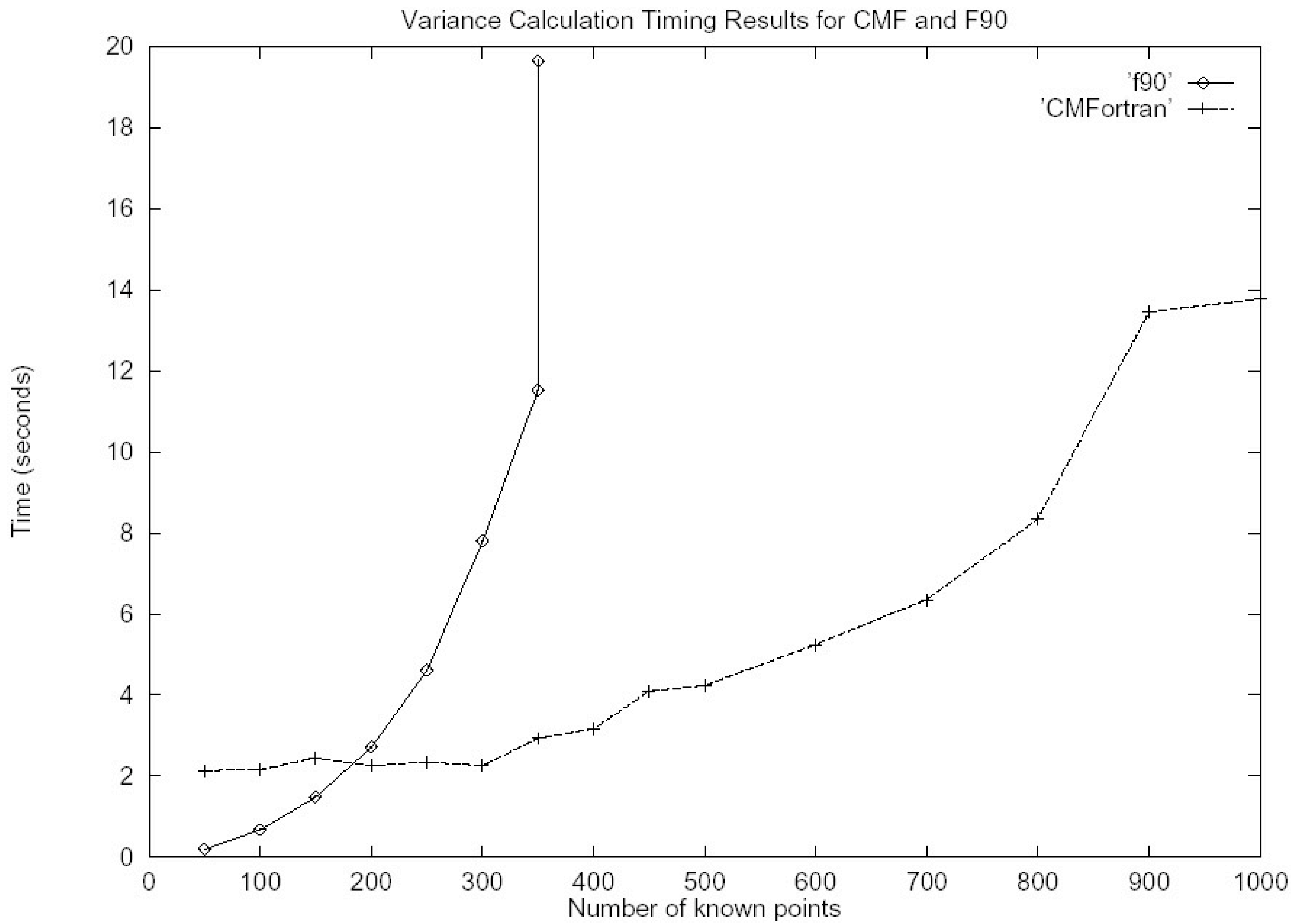
- By constructing a distance vector, d_{ij}, of the distances from the unknown point to each known point, we can calculate the weight vector by $E_{ij} = w_{ij}P$

Kriging interpolation

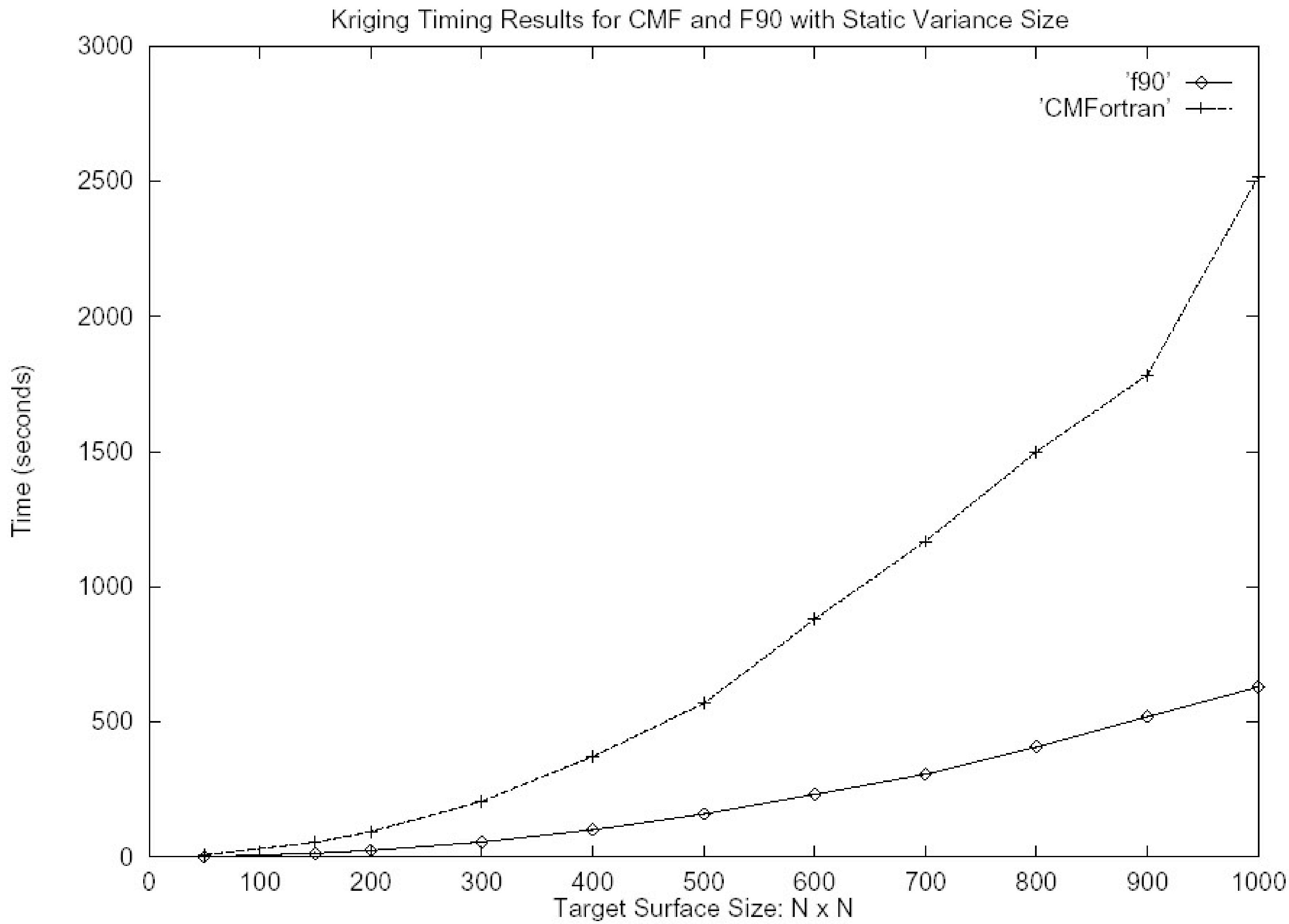
- The experiment.
  - 128 processor Connection Machine (CM5).
    - Used primarily to solve the matrix component of the Kriging algorithm.
  - 12 Alpha workstations connected by a high-speed ATM network.
    - They are investigating the performance of this system as an alternative to a dedicated parallel supercomputer.
- First parallel implementation of the Kriging algorithm.

Kriging interpolation

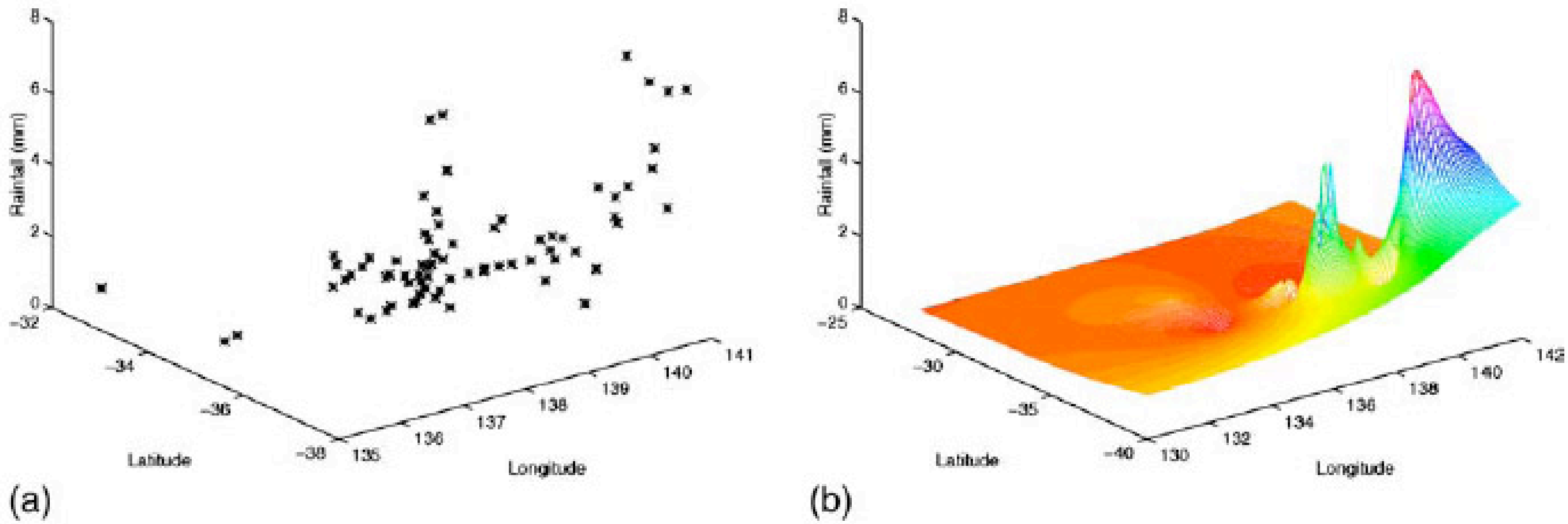
- Performance
  - Depends on the number of known data points and the size of the target surface.
  - LU matrix solver: grows as N^3 for an N x N target surface grid.
  - 10 min. to interpolate 100 points onto a 500 x 500 target surface.
  - Rainman contains 4000 points and the image of Australia is 1000 x 1000 pixels.
- The LU solver can be efficiently parallelized on most parallel architectures.



Timing Results for inversion of Variance matrix $N \times N$ with N = Number of known points.



Timing Results for Kriging algorithm, with target surface area, $E \times E$, with E = Target surface size and Number of known points = 100.



Interpolated rainfall surface for a selection of known rainfall data: (a) known data points; (b) interpolated surface using Kriging.

Spatial data grids

- Many computational grid projects require remote access to large amounts of data
- ‘Data grid’
  - To much data to handle efficiently
  - Metadata associated with a data set
  - The system must solve problems of where to locate data, and transmit it to where it is needed.

“Active” Data Grid

- A need for on-line data archives that can provide on-demand processing of data before delivery.
- Virtual data grid
  - On-demand processing of data
  - Data on remote servers can be accessed by the application as it were a local file.
  - Improved efficiency
  - Use pre-processing to crop out a region of interest in a large satellite image.
  - Must have a well-defined interfaces
  - Real-time results require high-end parallel servers.

Conclusion

- Improvements in the last decade
  - Power of individual processors
  - Memory and disk capacity
  - Bandwidth cost, capacity and availability
- Need to improve latency
  - Data replication
  - ‘chunking together’ of job requests into queries
- Movement to massive parallelism
  - Message passing
  - Language parallelism

Questions?