Presented by Jeremy Kepner at the:

High Performance Embedded Computing
(HPEC98) Workshop
September 23-24, 1998
MIT Lincoln Laboratory
Lexington, MA

Interfacing Interpreted and Compiled Languages for Computing on a Massively Parallel Network of Workstations (MP-NOW)

Jeremy Kepner[1,3], Maya Gokhale[2], Ron Minnich[2],
Aaron Marks[2], John DeGood[2]

[1]Princeton University, Princeton, NJ
[2]Sarnoff Corporation, Princeton, NJ
[3]Current address: MIT Lincoln Laboratory, Lexington, MA

Two Trends in Computing

- MP-NOW systems:
  - Readily available.
  - Flexible (commodity hardware *and* software).
  - Deliver superior price/performance (~$10/MFlop/s).
  - Represent future of high performance computing.

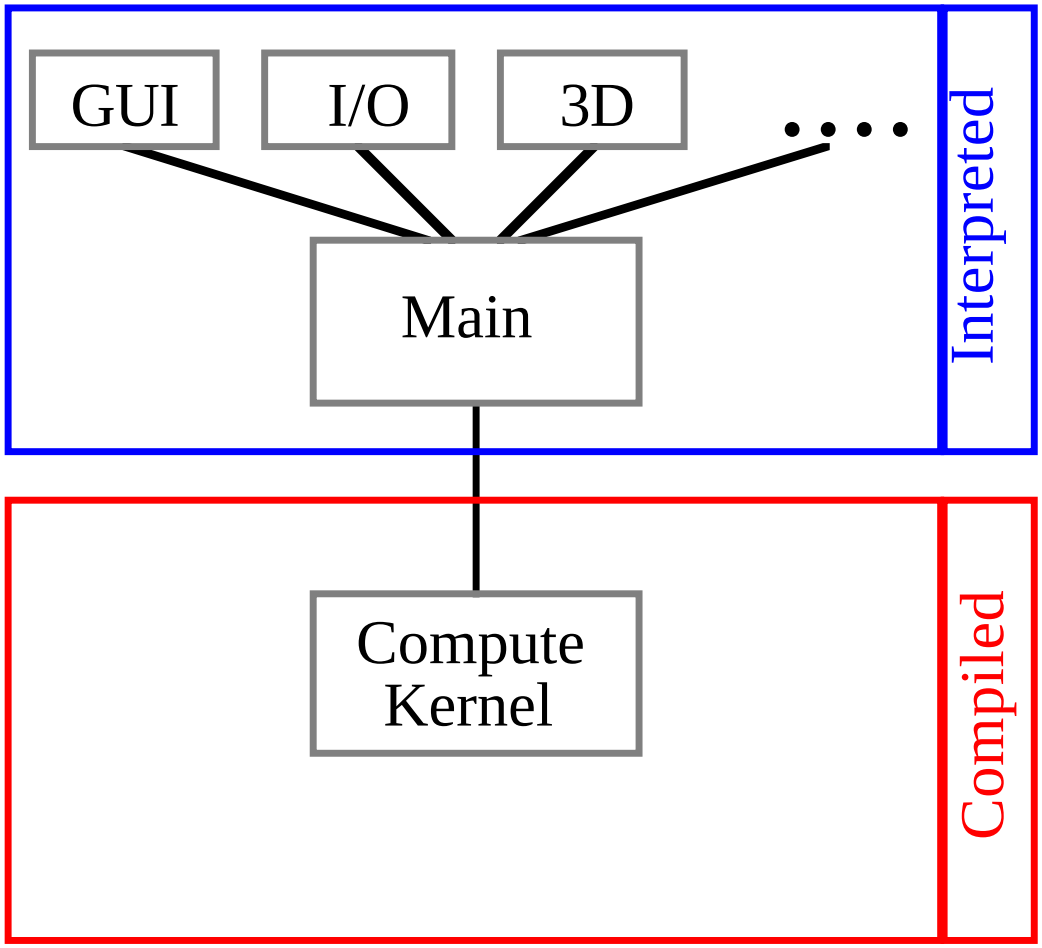
- Interpreted languages (e.g. MatLab, IDL & Mathematica) :
  - Widely used for algorithm design and prototyping.
  - Provide simplified I/O, GUIs and graphics.
  - Limited performance in certain CPU intensive operations.

Speedup of Interpreted Languages on a MP-NOW

Sarnoff Cyclone MP-NOW

- 128 nodes running Unix connected by a 100Mb/s switched Ethernet.
- Dual Pentium processor nodes, 64MB RAM and 3GB disk per node.
- Peak performance of 24,000 MFlop/s at a cost $10/MFlop/s.

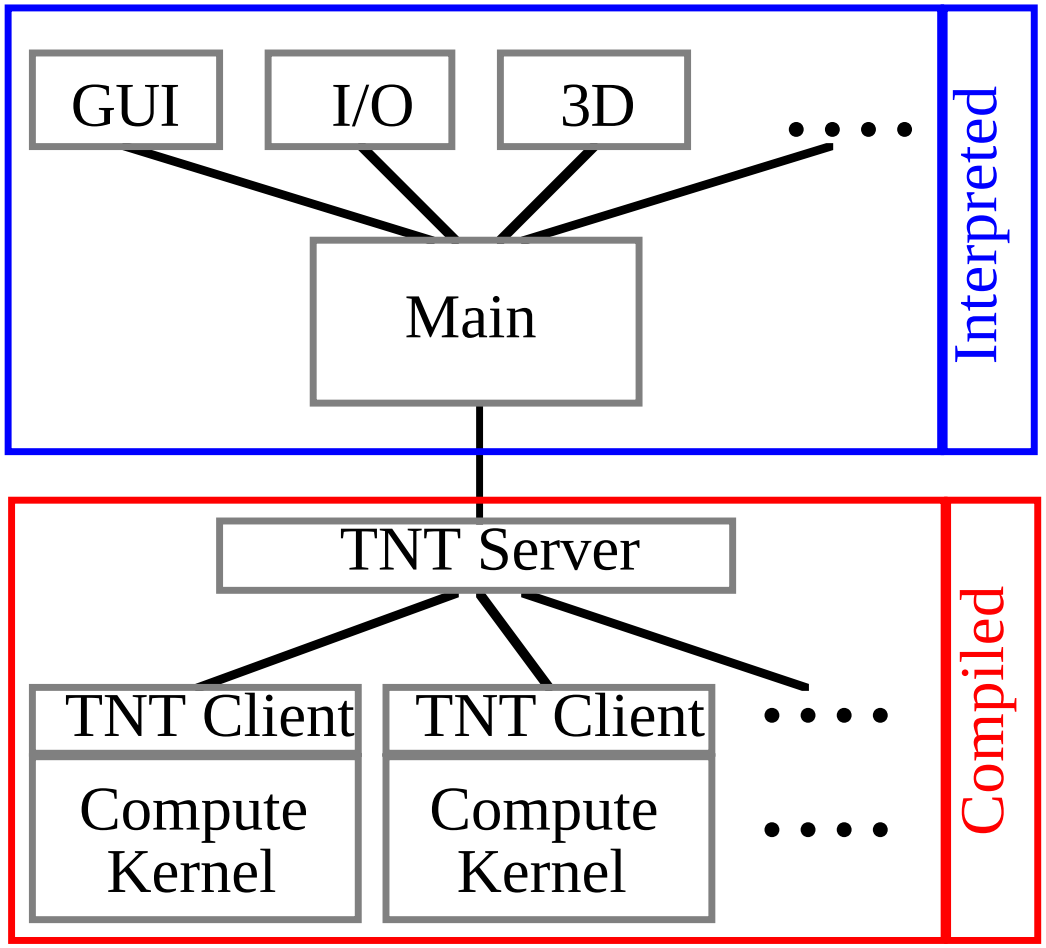
Speedup on a Single CPU



Single CPU Application Architecture

- Interpreted languages easily handle I/O, GUIs, graphics, and code management.
- Interpreted languages have simple mechanisms for calling external compiled libraries.
- Compiled computational kernel in this context is easy to port (no graphics or I/O).

Speedup on a MP-NOW



MP-NOW Application Architecture

- Uses TNT (The Next generation Taskbag) library.
- TNT client/server manages independent instances of compute kernel on a MP-NOW.
- Interpreted layer and computational kernel are unchanged.
- TNT client and server templates are adapted by programmer to problem.

TNT (The Next generation Taskbag) Library

- Consists of a Server with many clients, communicating via TCP/IP.

Server

– Places tasks into Taskbag.
– Listens on a specific port for requests for tasks from clients.
– Dispatches tasks to requesting clients.
– Accepts results from clients.
– Monitors status of clients; re-assigns tasks of dropped clients.
– When all tasks are completed, returns results back to Main.

Client

Loops until Taskbag is empty:
– Send requests for work to Server on a specific port.
– Reads data sent by Server over network.
– Calls compute kernel with the data.
– Sends results of computation back to server over network.

TNT API (Application Programmer Interface)

- TNT client/server templates contain calls to TNT library.
- Programmer replaces default subroutines in templates.
- Typical application requires programmer to write four subroutines:

Server

– ParseInput()
  passes data from the interpreted layer to the TNT server

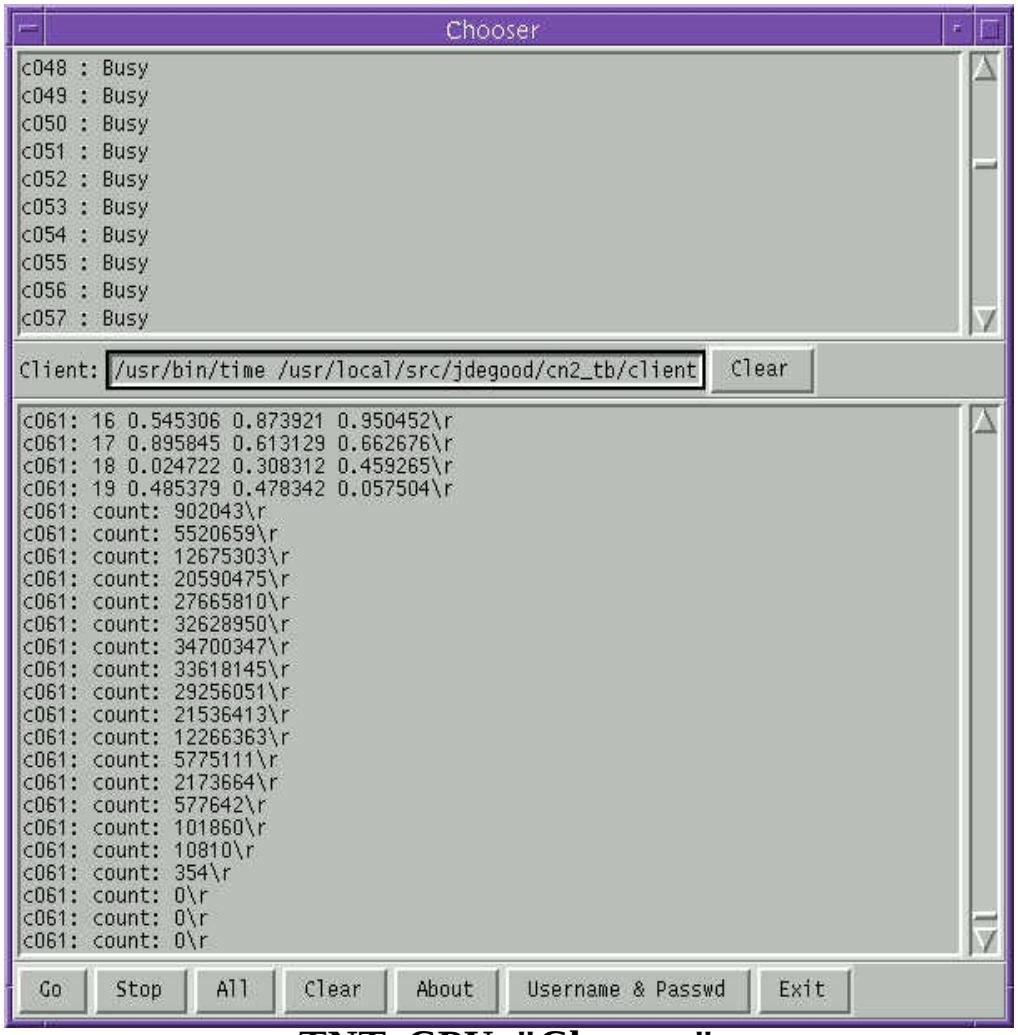
– FillTaskbag()
  passes data to the clients by placing tasks in Taskbag.

– ReturnResult()
  returns result of computation back to Main.

Client

– ProcessTask()
  calls the *unmodified* computational kernel.

TNT library on a MP-NOW



TNT CPU "Chooser"

- "Chooser" tool allows compute nodes to be selected interactively.
- Compute nodes can also be selected automatically when server starts.

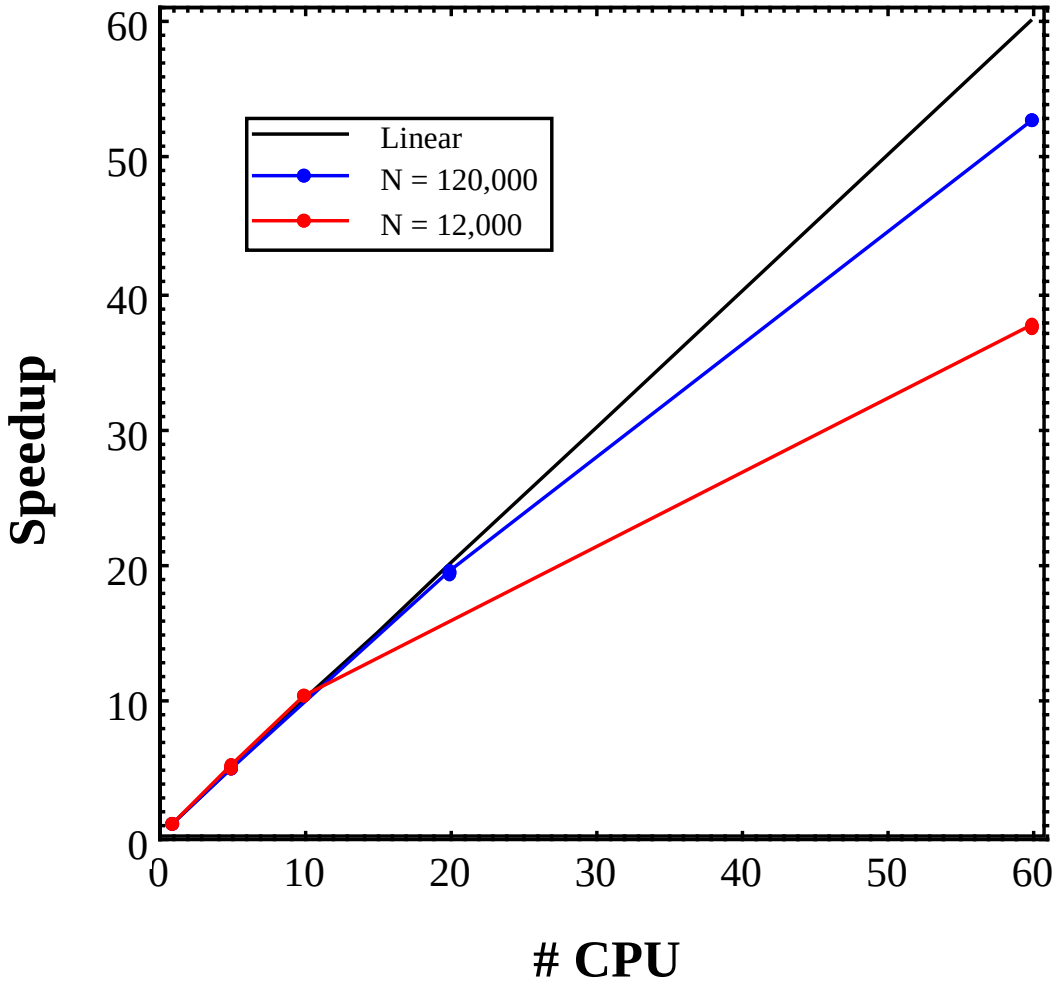
Test Problem: Pattern Recognition

- N vectors ($\mathbf{x}_1,...,\mathbf{x}_i,...\mathbf{x}_N$) each with D elements.
- Vectors composed of real, complex, integer, string or mixed data.
- Distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ given by:

$$d_{ij} = \text{distance}(\mathbf{x}_i , \mathbf{x}_j)$$

- Wish to compute M nearest neighbors to every point.
- Do direct calculation:
  - N^2 distance evaluations.
  - N sorts each requiring O(N log N) operations.
- Trivial to parallelize: do N/#CPU points on each CPU.
- Can explore both CPU dominated and communication dominated regimes by changing N, M and #CPU.

Parallel Performance



- Larger problem size shows nearly linear speedup: a factor of 52 on 60 CPUs.
- Communication overhead is more evident on smaller problem size.

Further Work

- Document and distribute TNT library
- Use TNT/IDL on other compute-intensive applications to further evaluate ease of use and performance.
- Enhance TNT to minimize data copying.