



Productivity for Domain Experts

David Gibson
Vice President of Business Development
dgibson@interactivesupecomputing.com



INTERNATIONAL
SUPERCOMPUTING CONFERENCE

'08

INTERACTIVE
supercomputing

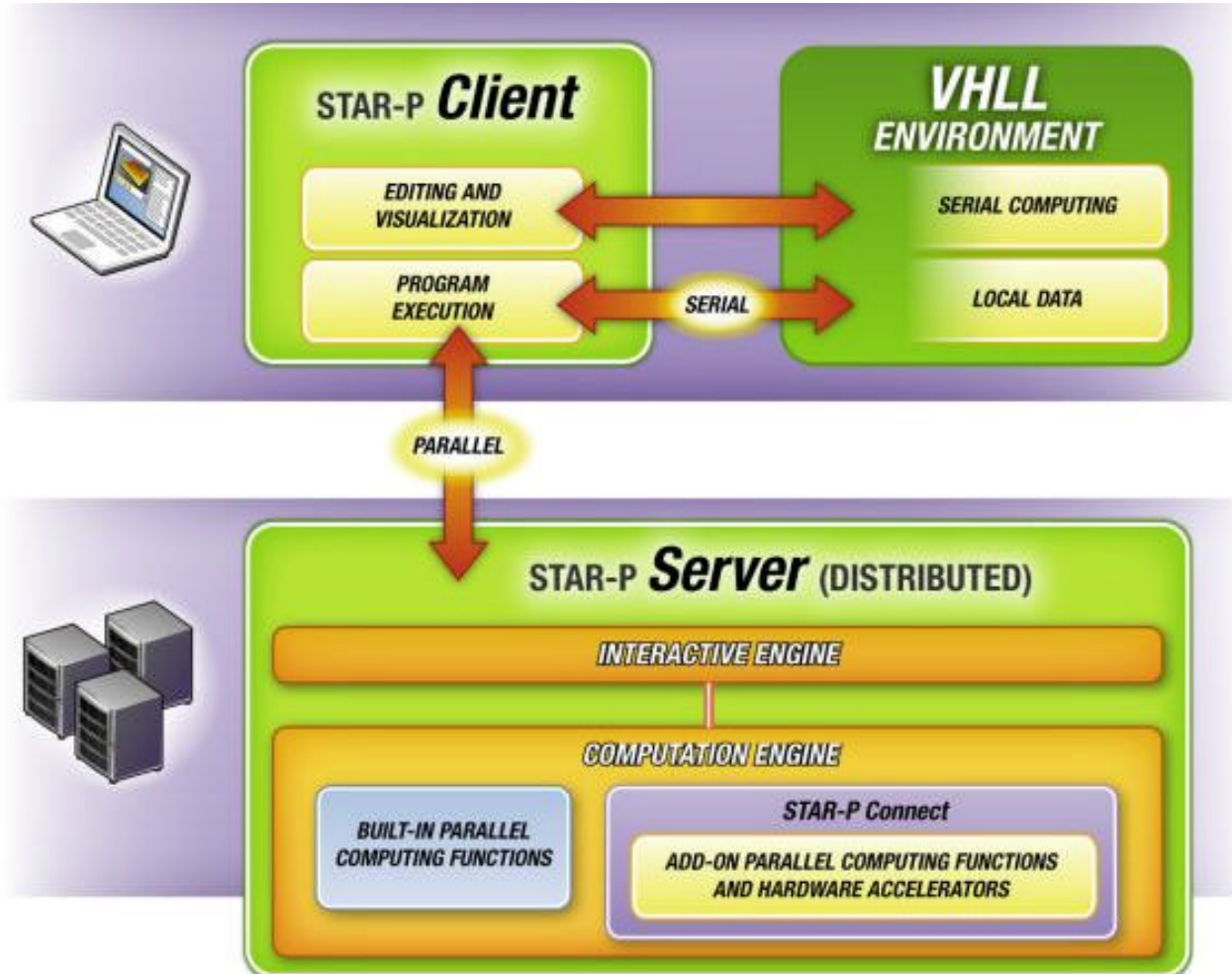
Star-P™ from Interactive Supercomputing, Inc.

- A parallel programming software platform designed to bridge desktop technical computing tools with parallel systems and clusters for numerical and scientific computing.
- Design goals include:
 - Leverage *easy to use* Very High Level languages:
 - **MATLAB®**, arguably the most-widely used interactive system for numerical/scientific computing;
 - **Python**, a popular scripting system; or
 - **R**, an increasingly important public-domain statistical system.
 - Empower Domain Experts with a “*command set*” that abstracts the complexity of parallel systems, servers, clusters, accelerators, ...
 - Extension of desktop memory pool and available processors/cores to large scale systems and clusters – “cluster memory looks like extension to desktop”
 - Server-side *serial and parallel libraries* maximize application performance:
 - broad coverage of math, statistics, imaging, signal, financial, optimization and other numerical and scientific tasks and/or functions.
 - Server-side software development kit (SDK) and application programming interface (API) enabling customers/users, ISC partners, and the open source community to add custom libraries, solvers and algorithms to the Star-P server-side library portfolio
 - SDK/API enables Clearspeed accelerator technology to be leveraged via the Star-P platform
 - Performance and scalability to 100’s of sockets/processors



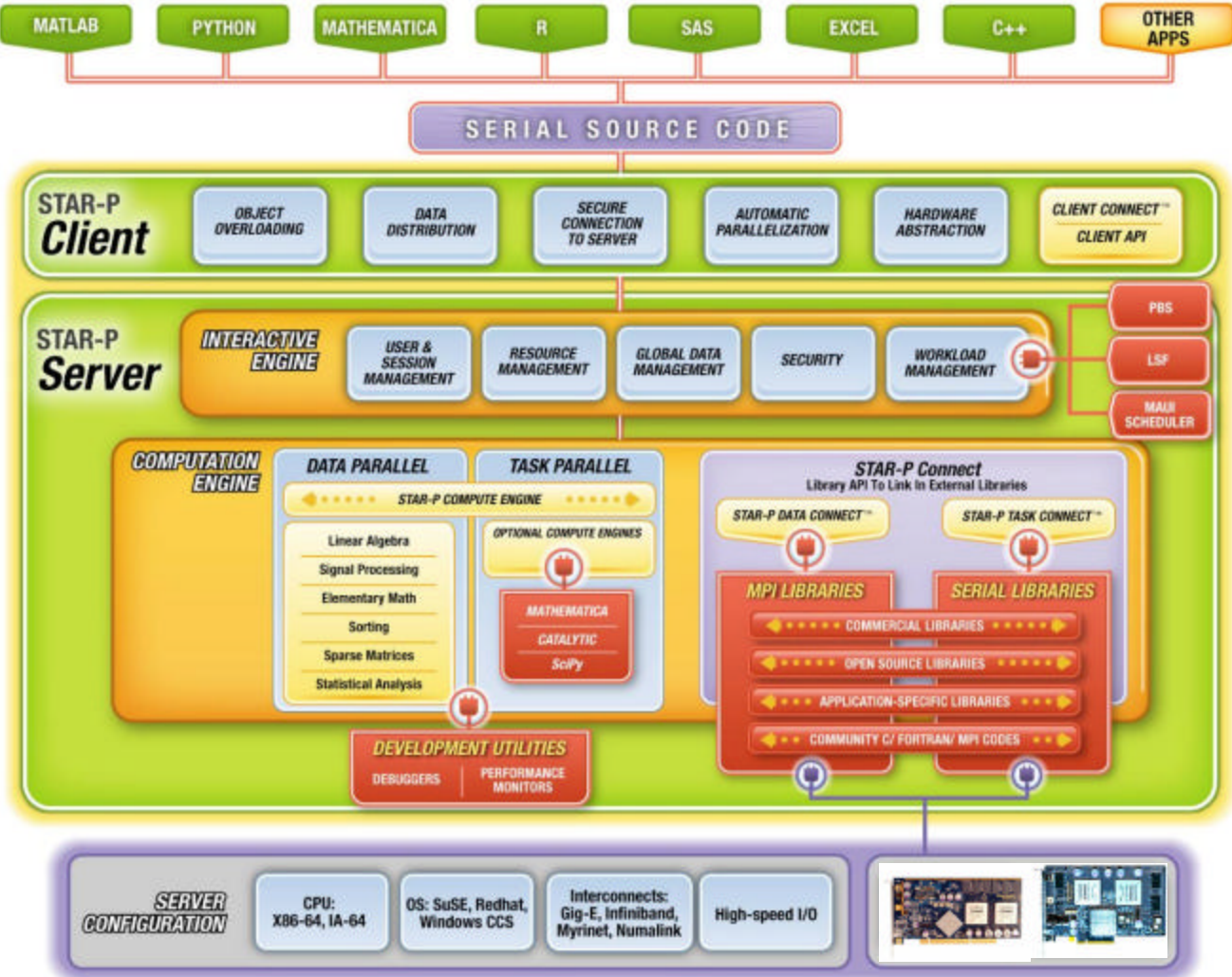
Core Star-P Innovations

Patented Technology with Distributed, Parallel & OO Techniques





Star-P Platform Overview



ClearSpeed™

Star-P Use Modes

- Data parallel
- Task Parallel

```
% explicitly parallel with p processes
n=10000*p

% implicitly parallel
A = rand(n, n);

% implicitly parallel
x = randn(n, 1);

% implicitly parallel
y = zeros(size(x));

while norm(x-y) / norm(x) > 1e-11
    y = x;
    x = A*x;
    x = x / norm(x);
end;
```

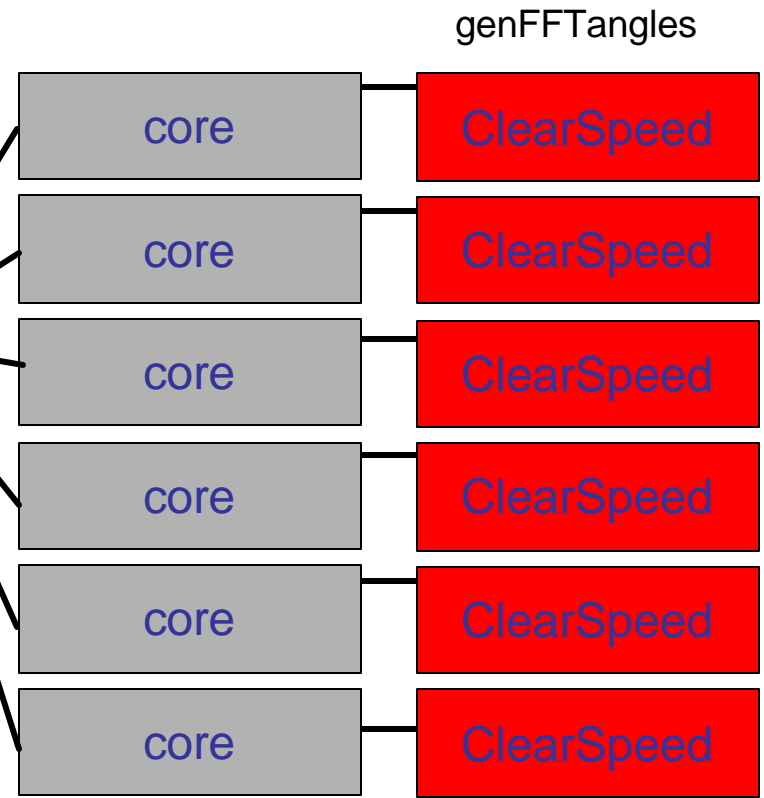
```
1 %Generate the Fourier Transform on 10 degree spacing
2 angles = linspace(0,360,37);
3 %Serial Version
4 load('brain.mat','A');
5 for i = 1:length(angles);
6     FFTangles(:, :, i) = genFFTangles(angles(i), A);
7 end
8
9
10
```

Task parallel with accelerator

```
1 %Generate the Fourier Transform on 10 degree spacing
2 angles = linspace(0,360,37);
3 %Serial Version
4 load('brain.mat','A');
5 for i = 1:length(angles);
6     FFTangles(:, :, i) = genFFTangles(angles(i), A);
7 end
8 %Parallel Version
9 ppload('brain.mat','A');
10 FFTangles = ppeval('genFFTangles', split(angles), bcst(A));
```

Where genFFTangles runs on Accelerator:

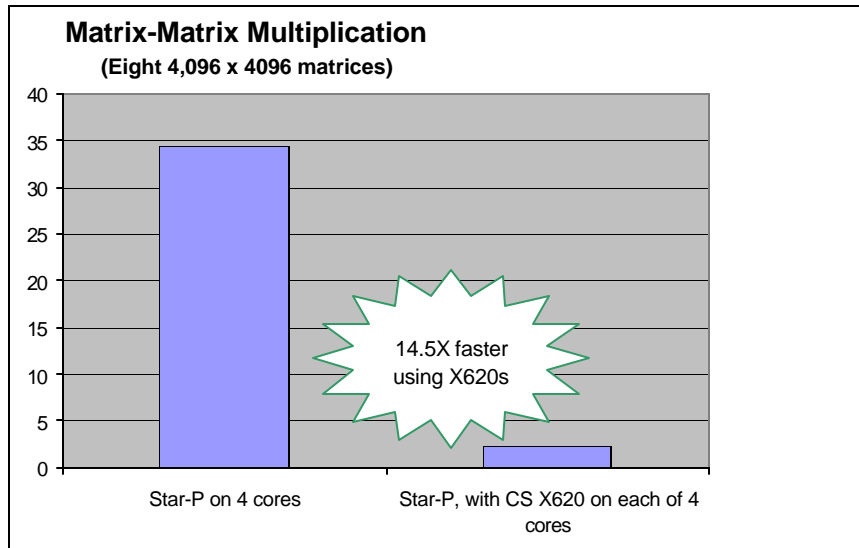
- Task parallel in Star-P via ppeval
 - Passes execution stream to board
 - Provides data distribution
 - Manages the library interfaces
 - Abstraction of details from user.



Sample results

```
1 %Perform matrix multiply on Four planes of 4k by 4k
2 %Distribute data to server or cluster
3 A = rand(4096, 4096, 8*p);
4 %Time and Execute matrix multiply across number of devices
5 tic; B = (ppeval "*", A, A); toc
6
7
8
9
10
```

Extends readily to other ClearSpeed library routines:
BLAS, LAPACK, and FFTs



- Task-parallel, on each plane of the input array
- Using 4 cores and 4 ClearSpeed X620 boards
- 14.5X faster using X620s
- Data-parallel operations that are not purely local (e.g. matmul) will require communication between calls to ClearSpeed routines

Programming Interface

- **For Star-P end-users**

- In shell rc file (*i.e.*, `.bashrc`) on the server system, ‘source’ a file that adds the ClearSpeed library to the search path.
- Star-P then uses the ClearSpeed library automatically for subsequent sessions.
- Zero VHLL source code changes to use accelerators

- **For Star-P internal developers**

- ISC had to add the ability for a user to specify from where libraries (*e.g.*, BLAS) are to be loaded;
 - sufficient for future ClearSpeed library additions
- Otherwise, ClearSpeed interface just worked
- ClearSpeed library makes threshold decisions;
 - *i.e.*, small matrices don’t benefit

Next steps...

- 3 to 5 early adopters/partners
 - Representative applications to do a general release
 - Complete integration of full Clearspeed library suite
 - Custom Libraries or algorithms already on accelerator
- Participation in beta program in Q4 2008 of next major release
 - Early adopters/partners drive the offering



For more information:
<http://www.interactivesupercomputing.com>

White papers
Case Studies
Customer Testimonials
Application Notes



INTERNATIONAL
SUPERCOMPUTING CONFERENCE

'08

INTERACTIVE
supercomputing