



Microsoft® Research

FacultySummit 2011

Cartagena, Colombia | May 18-20 | In partnership with COLCIENCIAS

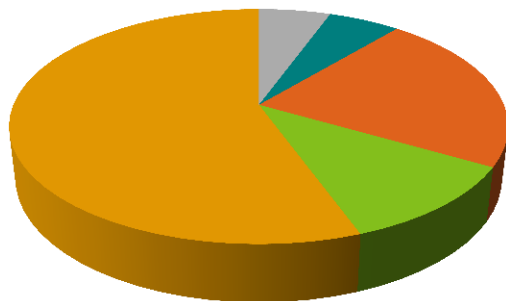
Tools of the trade: Technical Computing on the OS
... that is not Linux!

Or how to leverage everything you've learned, on a Windows box as well

Sean Mortazavi & Felipe Ayora

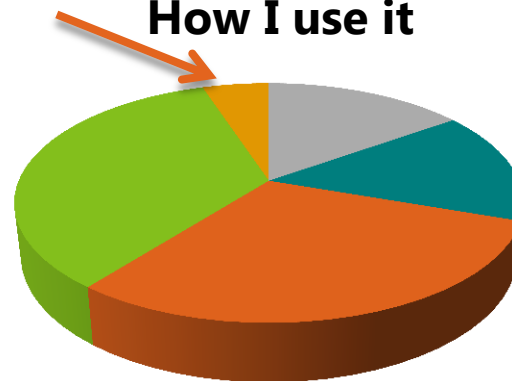
Typical situation with TC/HPC folks

Why I have a Windows box



- It was in the office when I joined
- IT forced me
- I couldn't afford a Mac
- Because I LIKE Windows!
- It's the best gaming machine

How I use it



- Outlook / Email
- PowerPoint
- Excel
- Gaming
- Technical/Scientific computing

Note: Stats completely made up!

The general impression



- ☞ “Enterprise community”
- ☞ Guys in suits
- ☞ Word, Excel, Outlook
- ☞ Run prepackaged stuff



- ☞ “Hacker community”
- ☞ Guys in jeans
- ☞ Emacs, Python, gmail
- ☞ Builds/runs OSS stuff

Common complaints about Windows

- I have a Windows box, but Windows ...
 - Is hard to learn... ☹️
 - Doesn't have a good shell ☹️
 - Doesn't have my favorite editor ☹️
 - Doesn't have my favorite IDE ☹️
 - Doesn't have my favorite compiler or libraries ☹️
 - Locks me in ☹️
 - Doesn't play well with OSS ☹️
 - ☹️
- In summary:



My hope ...

- I have a Windows box, and Windows ...
 - Is easy to learn... 😊
 - Has excellent shells 😊
 - Has my favorite editor 😊
 - Supports my favorite IDE 😊
 - Supports my compilers and libraries 😊
 - Does not lock me in 😊
 - Plays well with OSS 😊
 - 😊
- In summary:



(or at least 😊)

How?

- Recreating a Unix like veneer over windows to minimize your learning curve
- Leverage your investment in know how & code
- Showing what key codes already run natively on windows just as well
- Kicking the dev tires using cross plat languages

Objective is to:

Help you ADD to your toolbox,
not take anything away from it!

At a high level...

- Cygwin
- SUA
- Windowing systems
- Standalone shell/utills

- IDE's
- Editors
- Compilers / languages / Tools
- make
- Libraries

- CAS environments



"The Unix look & feel"

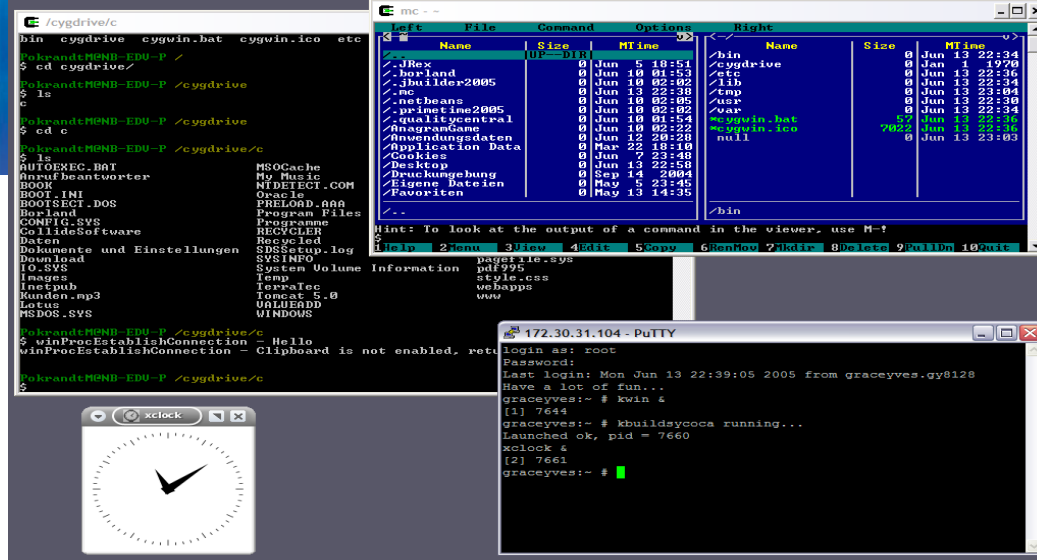
General purpose development

Dedicated CAS / IDE's

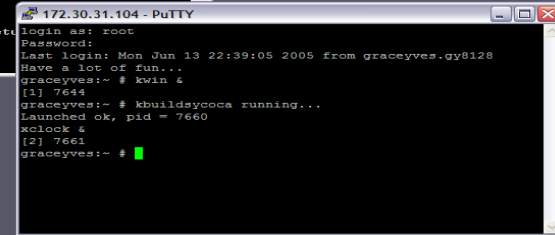
And if there is time, a couple of demos...

Cygwin

- What is it?
 - A Unix like environment for Windows. Native integration of familiar Unix tools & apps built from source for Windows.
- How does it work?
 - POSIX support is provided by the cygwin.dll library which enables code migration w minimal changes
 - The usual shells, utilities, etc are compiled as native Windows binaries against cygwin.dll
- What it's not
 - Cygwin doesn't provide emulation of Unix programs under windows – apps must be built from src
- Integration
 - All std tools, shells, mounts, file conversions, symlinks, ACL's, various langs + gcc for windows, ssh, telnet, ftp, ...
 - You can use win32 API's *and* POSIX



- Sample ports
 - Openoffice, Sun Java, ...
- License, who
 - GPL v2, redhat
- Useful links
 - www.cygwin.com



SUA

Subsystem for Unix Applications

What is it?

- A Unix-like environment for Windows as a subsystem. Native integration of familiar Unix tools & apps built from source for Windows.

How does it work?

- Similar to Cygwin, but implemented as subsystem
- All tools, shells, utils are built from source, link to Windows C runtime. Support for msft compilers via wcc wrapper.

What it's not

- SUA doesn't provide emulation of Unix programs under windows

Integration

- All std tools/shells + better NFS, Oracle/MSFT SQL, AD/user mgmt support
- With later version can mix win32 & posix calls

The image shows two terminal windows. The top window is a Korn Shell (ksh) terminal with a blue title bar. It displays the message 'Welcome to the Interix UNIX utilities.' and the environment variable 'DISPLAY=aplxp:0.0'. The prompt '\$ ls' is followed by a directory listing of files and folders such as 'Application Data', 'Cookies', 'Desktop', 'Favorites', 'Local Settings', 'My Documents', 'NTUSER.DAT', 'NetHood', 'PrintHood', 'Recent', 'SendTo', 'Start Menu', 'Templates', 'ntuser.dat.LOG', and 'ntuser.ini'. The bottom window is a C Shell (csh) terminal with a blue title bar. It shows the command '% ps -ef | head -5' and its output, which is a table of process information including UID, PID, PPID, STIME, TTY, TIME, and CMD. The processes listed include +SYSTEM, <nistrator, and <nistrator running various system services like /usr/sbin/zzInterix, /usr/sbin/init, /usr/sbin/syslogd, and /usr/sbin/inetd. The prompt '% ps -ef | grep notepad' is also shown, along with its output listing several instances of notepad running with different PIDs and times.

- License, who
 - Free with versions of Windows, MSFT

- Useful links:
 - <http://bit.ly/fTVRtu> for the SDK tools
 - <http://bit.ly/jwGCpD> unix/windows dictionary
 - <http://bit.ly/kK3f15> SUA community

MKS Toolkit

• What is it?

- Another Unix-like environment for Windows .
Native integration of familiar Unix tools & apps built from source for Windows.

• How does it work?

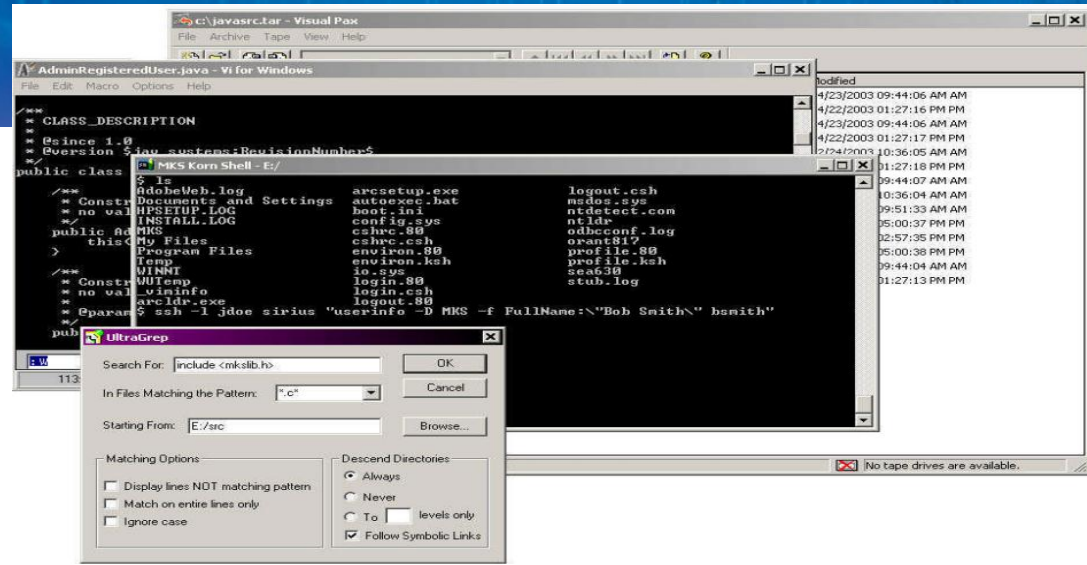
- Native ports of unix shell/utils

• What it's not

- MKS doesn't provide emulation of Unix programs under windows
- MKS is not free

• Integration

- Posix support DLL like cygwin
- Good Enterprise support for AD, users, pwd sync, WMI, ...
- Best of the bunch for sysadmin'ing heterogeneous farms



- License, who
 - Proprietary, MKS

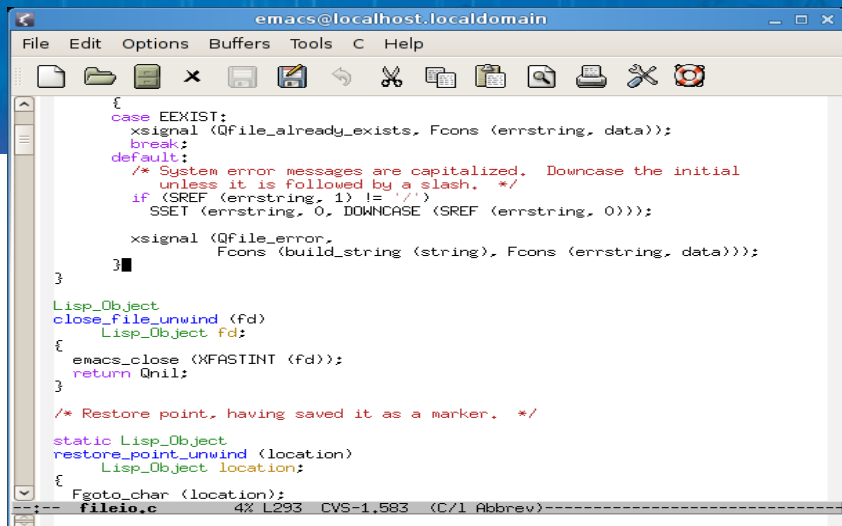
- Useful links:
 - <http://mkssoftware.com> general
 - <http://www.mkssoftware.com/products/tk/commands.asp?product=tkdev> developers

Shells & Utilities options

- All common shells are available on Windows:
 - sh, bash, csh, tcsh, zsh, kshell, ...
- Getting them - Option 1: "Distros"
 - MKS
 - SUA
 - Cygwin
 - MSYS
 - ...
- Getting them - Option 2: "Just the basics please"
 - GnuWin – windows versions of gnutools
 - Native ports of 150+ utils – using mingw / MSVC, no emulation
 - Distributed w gnuemacs, KDE, ...
 - GPL

Editors

- The classics
 - Vi
 - Vim
 - Emacs
- The newer batch
 - Notepad2
 - Notepad++
 - E (Textmate)
 - TotalEdit
 - UltraEdit
 - Pspad
 - EditPlus
 - ...
- Emulation support in Visual Studio
 - ViEmu
 - VsVim
 - Emacs



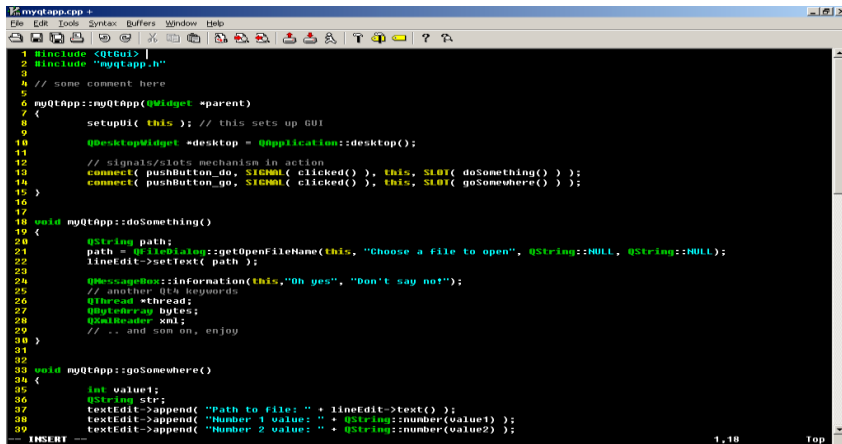
```
emacs@localhost.localdomain
File Edit Options Buffers Tools C Help
{
  case EEXIST:
  xsignal (Qfile_already_exists, Fcons (errstring, data));
  break;
  default:
  /* System error messages are capitalized. Downcase the initial
  unless it is followed by a slash. */
  if (SREF (errstring, 1) != '/')
    SSET (errstring, 0, DOWNCASE (SREF (errstring, 0)));
  xsignal (Qfile_error,
          Fcons (build_string (string), Fcons (errstring, data)));
}
}

Lisp_Object
close_file_unwind (fd)
  Lisp_Object fd;
{
  emacs_close (XFASINT (fd));
  return Qnil;
}

/* Restore point, having saved it as a marker. */

static Lisp_Object
restore_point_unwind (location)
  Lisp_Object location;
{
  Fgoto_char (location);
}

fileio.c      4% L293 CVS-1,583 (C/I Abbrev)
```



```
myQtApp.cpp
File Edit Tools Syntax Buffers Window Help
1 #include <QtGui>
2 #include "myQtApp.h"
3
4 // some comment here
5
6 myQtApp::myQtApp(QWidget *parent)
7 {
8     setupUI( this ); // this sets up GUI
9
10    QDesktopWidget *desktop = QApplication::desktop();
11
12    // signals/slots mechanism in action
13    connect( pushButton_go, SIGNAL( clicked() ), this, SLOT( doSomething() ) );
14    connect( pushButton_go, SIGNAL( clicked() ), this, SLOT( goSomewhere() ) );
15 }
16
17 void myQtApp::doSomething()
18 {
19 }
20
21 QString path;
22 path = QFileDialog::getOpenFileName(this, "Choose a file to open", QString::NULL, QString::NULL);
23 QLineEdit->setText( path );
24
25 QMessageBox::information(this, "Oh yes", "Don't say no!");
26 // another Qt keywords
27 QLineEdit *thread;
28 QTextReader xml;
29 // .. and so on, enjoy
30 }
31
32
33 void myQtApp::goSomewhere()
34 {
35     int value1;
36     QString str;
37     textEdit->append( "Path to file: " + QLineEdit->text() );
38     textEdit->append( "Number 1 value: " + QString::number(value1) );
39     textEdit->append( "Number 2 value: " + QString::number(value2) );
40 }
INSERT 1,18 Top
```

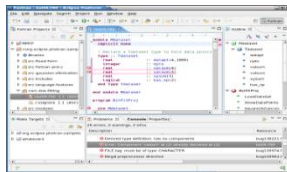
Demo: Shells

- Installation & usage of Unix shells / utilities
- PowerShell: the new msft shell – mixing & matching shells
- Vi / Emacs

IDE examples

- **Eclipse**

- Started a Java env, now w C++ and various other plug-ins



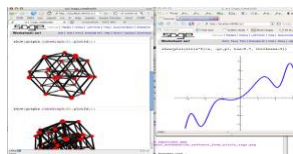
- **IPython**

- Interactive Python REPL w support for parallel computing



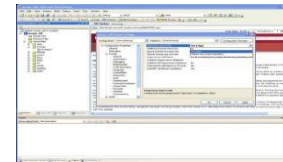
- **Sage**

- Symbolic math IDE w Python as the scripting language



- **Visual Studio**

- Various languages w support for cross plat compilers



- **RevoAnalytics R**

- A complete R development environment w debugging and visualization



Build environments

- MKS / CygWin / SUA
- MingW
- Cross-plat compilers
- Scripts, make, nmake, Cmake, ...
- /, vs \, File vs file, drive names, ...

```
# first we add the executable that generates the table
add_executable(MakeTable MakeTable.cxx)

# add the command to generate the source code
add_custom_command (
  OUTPUT ${CMAKE_CURRENT_BINARY_DIR}/Table.h
  COMMAND MakeTable ${CMAKE_CURRENT_BINARY_DIR}/Table.h
  DEPENDS MakeTable
)

# add the binary tree directory to the search path for
# include files
include_directories( ${CMAKE_CURRENT_BINARY_DIR} )

# add the main library
add_library(MathFunctions mysql.cxx ${CMAKE_CURRENT_BINARY_DIR}/Table.h )
```

The Cmake crossplat build utility

Languages, Compilers/interpreters

- Classics

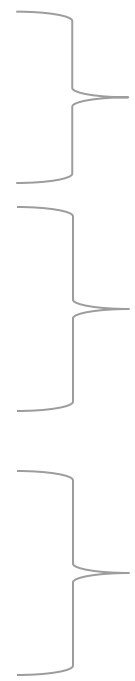
- C/C++
- Fortran

- Newer on the block

- C#/Java
- Python
- R
- F#

- Dedicated CAS langs

- M
- Mathematica
- Maple
- ...

- 
- C: gcc, VC++, Intel*, clang, ...
 - Fortran: Intel*, PGI*, ...

* generally the best FP optimizations

- JVM, CLR langs
- CPython, IronPython, Pypy, Jython, ...
- Revolution R, R-Studio

- IDE + language combos

Library/runtime Examples

- Boost
- Python: SciPy, Numpy, scikits, ...

- From text to image processing to large graphs & MPI
- Numeric & Scientific libraries

- MKL
- IMSL
- NAG
- Visual Numerics
- LibFlame
- ATLAS

- Highly optimized Math libraries for native & .Net number crunching

- MPI, OpenMP, CUDA, ...

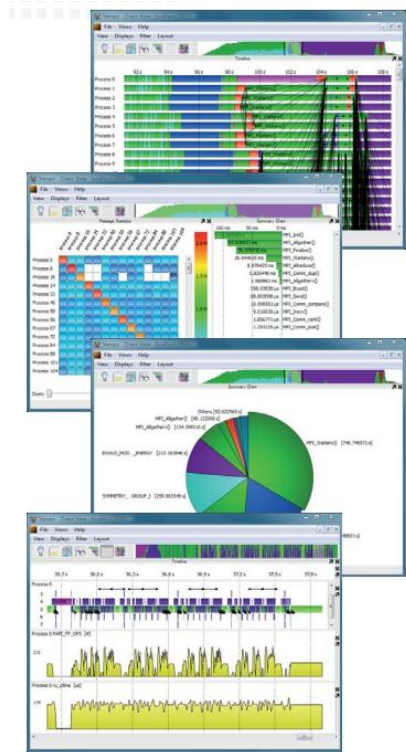
- Optimized || libraries for multicore, cluster and GPUs

- NetCDF, HDF5, FITS, ...

- OS & language neutral scientific data formats & libraries

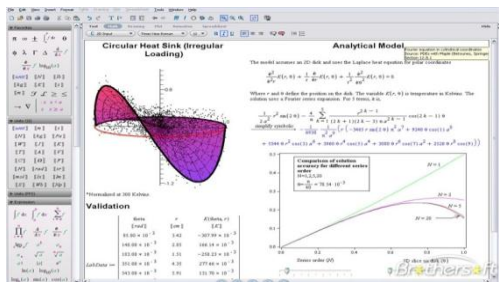
Debugging, Profiling, Analysis

- VS plug-ins from Intel, PGI, Allinea (MPI)
- VS plug-in for Python/mipi4py
- MPI Tracing
 - JumpShot
 - Vampir (native windows port)
- ETW (“DTrace”)
 - Event Tracing for Windows
- Marmmot for VS
 - MPI call / param analysis (runtime)
- ISP
 - MPI program verification

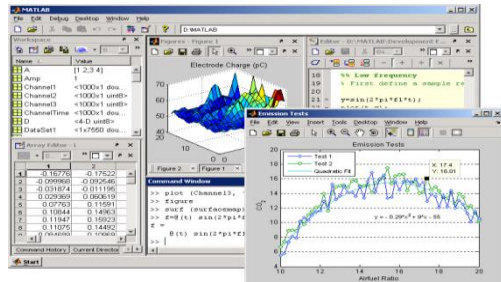


Computer Algebra Systems

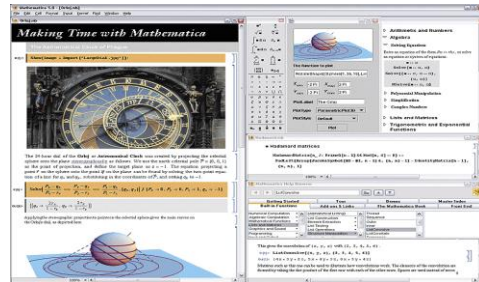
Commercial



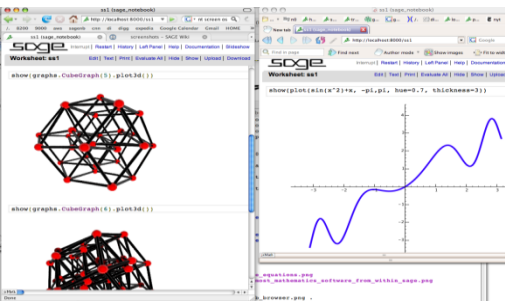
Maple *



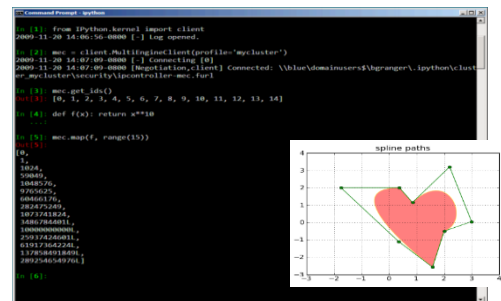
MATLAB *



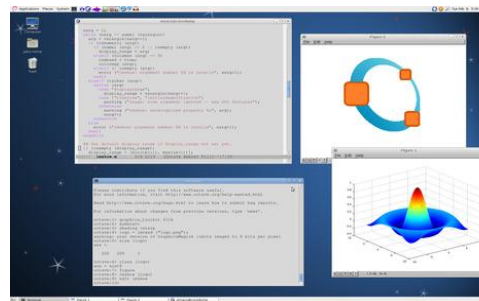
Mathematica *



Sage



IPython *



Octave

Free/OSS

* Can parallelize using Windows HPC

Now that you're here... stuff you really should try!

- Visual Studio
- F#
- MPI.Net
- Python Tools for VS
- PowerShell
- CUDA

Visual Studio

- State of the art development environment
 - Development, debugging, profiling, life-cycle mgmt, ...
- Technical Computing related
 - Best host for key compilers: C++, Fortran, PGI
 - Free full featured Python dev plug-in
 - Best host for AMD, Nvidia plug-ins: see Nsight
 - Support for Vim & Emacs 😊
- Academic / hobbyist related
 - Essentially free to schools via www.dreamspark.com program
 - Free to startups via www.bizspark.com program

F#

- Multi-paradigm language
- Succinct & powerful
- Interactive mode
- Runs on MacOS & Linux too!
- Free / open source
- Check out “Units of Measure”, async
- See Christophe’s **demo** at the demofest!

Python Tools for VS

- Free & Open source plug-in for writing Python code
- Intellisense, browsing, ...
- Standard REPL + IPython REPL
- Cluster support: MPI & IPython
- Debugging
- Profiling
- CPython, IronPython, Jython, ...
- Soon: Big Data, Big Compute support

The screenshot shows the PythonMPI1 debugging interface in Microsoft Visual Studio. The main editor displays the following Python code:

```
computePi
mypi = (inside * 1.0) / numSamples
sndBuf[0] = mypi
comm.Allreduce((sndBuf, MPI.DOUBLE), (recvBuf, MPI.DOUBLE), op=MPI.SUM)
pi = (4.0 / size) * recvBuf[0]

delta = abs(pi - oldPi)
if rank == 0:
    print("pi: %f - delta: %f\r\n" % (pi, delta))

if delta < 0.00001:
    break
return pi

if __name__ == '__main__':
    pi = computePi(10000)
    if comm.Get_rank() == 0:
        print("Computed value of pi on %d processors is %f" % (size, pi))
```

The Locals window shows the following variables:

Name	Value	Type
rank	14	int
oldPi	3.140849999999999995	float
pi	3.147549999999999998	float
mypi	0.786900000000000004	float
sndBuf	array('d', [0.786900000000000004])	array
recvBuf	array('d', [18.8853000000000001])	array
inside	[0]	float
inside	7869	int
i	9999	int
x	0.21626256737742355	float
y	0.8718039304687516	float
delta	0.006700000000000003723	float
numSamples	10000	int

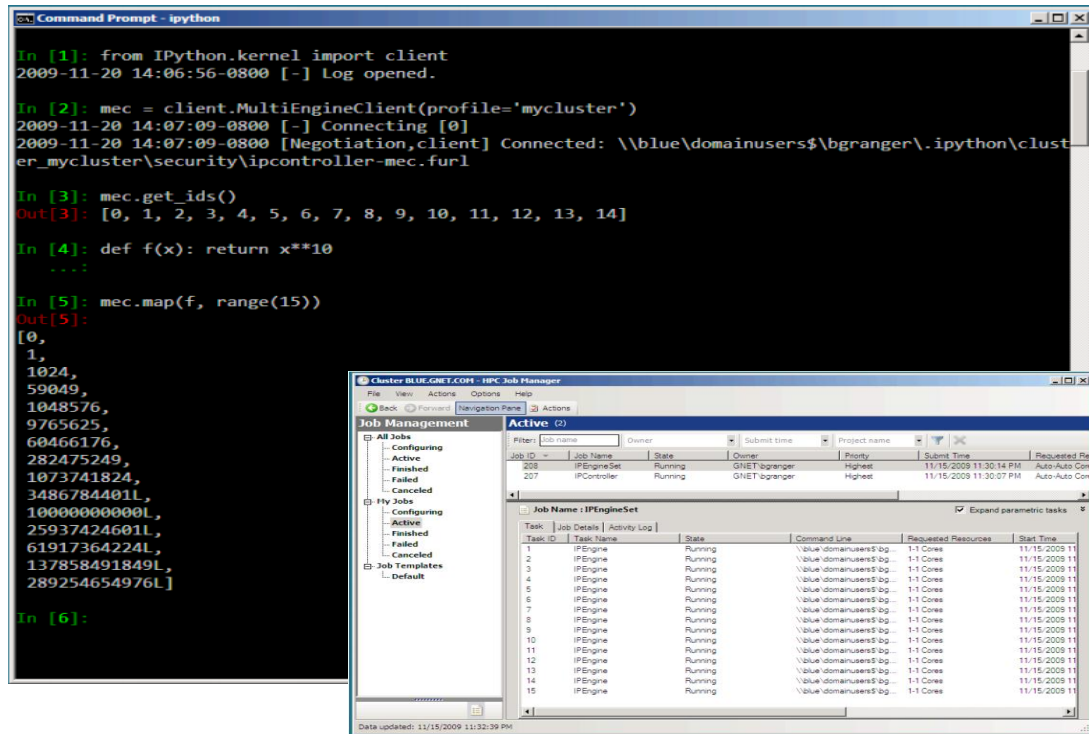
The Python 2.6 Interactive window shows a histogram of pi values and a loop for i in range(1, N).

Demo

- Python Tools for Visual Studio
 - Intellisense
 - Profiling
 - Cluster MPI debugging

Noteworthy: IPython

- Interactive computing using Python
 - Advanced REPL with History, completion, ...
 - Capture 'var = !ls -la'
 - Inline images
- Interactive `||` computing
 - Specify cluster headnode,
 - Start # of desired engines
 - Compute!
- Included in all major Python distro's
- Open source & available on Windows, Linux, Mac



```
Command Prompt - ipython

In [1]: from IPython.kernel import client
2009-11-20 14:06:56-0800 [-] Log opened.

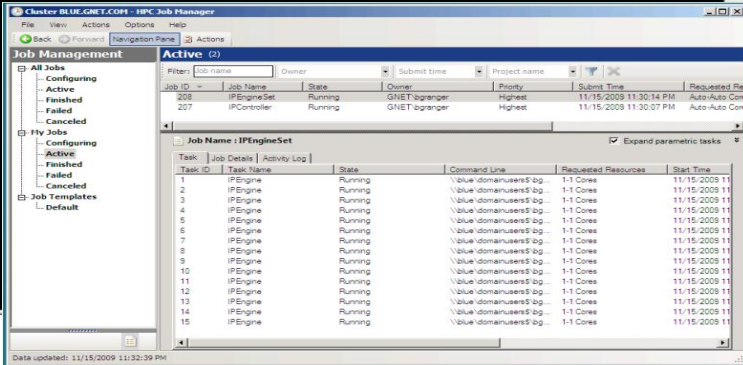
In [2]: mec = client.MultiEngineClient(profile='mycluster')
2009-11-20 14:07:09-0800 [-] Connecting [0]
2009-11-20 14:07:09-0800 [Negotiation_client] Connected: \\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl

In [3]: mec.get_ids()
Out[3]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

In [4]: def f(x): return x**10
...:

In [5]: mec.map(f, range(15))
Out[5]:
[0,
 1,
1024,
59049,
1048576,
9765625,
60466176,
282475249,
1073741824,
3486784401L,
10000000000L,
25937424601L,
61917364224L,
137858491849L,
289254654976L]


In [6]:
```



Job ID	Job Name	State	Owner	Priority	Submit Time	Requested Resources
208	IPythonSet	Running	GHEtBgranger	Highest	11/15/2009 11:30:14 PM	Auto-Auto Core
207	IPController	Running	GHEtBgranger	Highest	11/15/2009 11:30:07 PM	Auto-Auto Core

Task ID	Task Name	State	Command Line	Requested Resources	Start Time
1	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
2	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
3	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
4	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
5	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
6	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
7	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
8	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
9	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
10	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
11	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
12	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
13	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
14	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM
15	IPython	Running	\\blue\domainusers\bgranger\ipython\cluster_mycluster\security\ipcontroller-mec.furl	1-1 Cores	11/15/2009 11:30:14 PM

Why Python

- Well suited to Technical & Scientific Computing
 - Isn't it interpreted? (and slow?!)
 - Easy ramp up, yet powerful language
 - Incredibly rich ecosystem of high quality libraries
 - Healthy developer eco-system
 - Various implementations
 - Free, open source w quality distro's providing support
- 
- Interactive by design
 - Easily mix in native code, even CUDA, MPI, ... see Cython, swig, ctypes
 - From CS101 to PhD thesis
 - Web, numerics, symbolics, Bio, astronomy, ...
 - 2D, 3D viz, See scipy, scikits
 - #3 most popular in some surveys
 - PyCon is the main conference
 - CPython, IronPython, Jython, PyPy, ...
 - BSD like license
 - Enthought Python Distro, ActivePython, ...

MPI.Net : A high performance wrapper for MPI

C vs C#: gather cluster hostnames / sort / print

```
Communicator comm = Communicator.world;
string[] hostnames =
    comm.Gather(MPI.Environment.ProcessorName, 0);
if (Communicator.world.Rank == 0) {
    Array.Sort(hostnames);
    foreach (string host in hostnames)
        Console.WriteLine(host);
}
```

```
int size, rank;
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &size);

char name[MPI_MAX_PROCESSOR_NAME];
int resultlen;
MPI_Get_processor_name(name, &resultlen);

int *rbuf = (int*)malloc(sizeof(int) * size);
MPI_Gather(&resultlen, 1, MPI_INT, rbuf, 1, MPI_INT, 0, MPI_COMM_WORLD);

int *rcounts = (int*)malloc(sizeof(int) * size);
int *rdispls = (int*)malloc(sizeof(int) * size);
int cnt = 0;
for (int i = 0; i < size; i++) {
    rcounts[i] = rbuf[i]+1;
    cnt += rcounts[i]+1;
    if (i) rdispls[i] = rdispls[i-1]+rbuf[i]+1;
    else rdispls[i] = 0;
}
char *rnamebuf = (char*)malloc(cnt * sizeof(char));
MPI_Gatherv(name, resultlen, MPI_CHAR, rnamebuf, rcounts, rdispls, MPI_CHAR, 0,
    MPI_COMM_WORLD);
if (rank == 0) {
    char **hostnames = (char**)malloc(size * sizeof(char*));
    for (int i = 0; i < size; ++i)
        hostnames[i] = rnamebuf + rdispls[i];
    qsort (hostnames, size, sizeof(char*), strcmp);
    for (int i = 0; i < size; ++i)
        printf("%s\n", hostnames[i]);
}
```

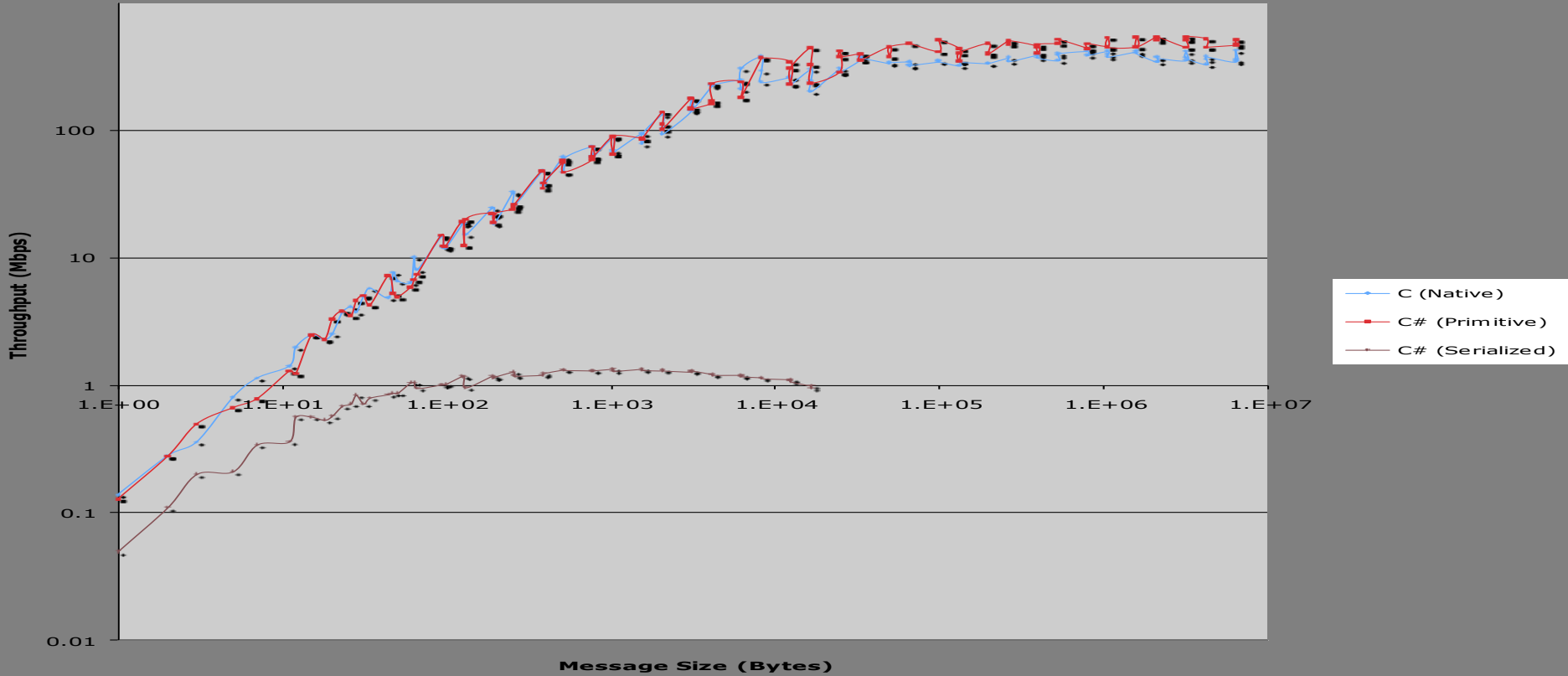
Optimizing MPI.Net performance

An optimal Send?

```
public void Send<T>(T value, int dest, int tag)
{
    if (HasMpiDatatype<T>()) {
        unsafe {
            fixed (T* valuePtr = &value) {
                Unsafe.MPI_Send(new IntPtr(valuePtr), 1,
                    GetMpiDatatype<T>(), dest, tag, comm);
            }
        }
    }
    else {
        // Serialize and transmit
    }
}
```

C#	MPI
short	MPI_SHORT
int	MPI_INT
float	MPI_FLOAT
double	MPI_DOUBLE

NetPIPE Performance



Parallel Dwarfs Project

- The “Dwarfs”
 - Colella’s original 7, now 13 kernels that encapsulate a large spectrum of computing workloads
- Parallel Dwarfs: Visual Studio solutions that implement || versions of the dwarfs (13k+ LOC)
 - Languages: C++, C#, (some F#)
 - Input files: small, medium, large
 - Parallelization technologies:
 - OpenMP, TPL, MPI, MPI.Net, (ClusterSOA, PPL soon)
 - Results gathering & plotting
 - Excel, JumpShot, Vampir
 - “Driver” for selecting & running the benchmarks
 - Open Source on codeplex.com

Dwarf Popularity¹

		HPC	Embed	SPEC	ML	Games	DB
1	Dense Matrix	Red	Red	Red	Red	Red	Yellow
2	Sparse Matrix	Red	Yellow	Yellow	Red	Red	Grey
3	Spectral (FFT)	Red	Yellow	Grey	Yellow	Yellow	Grey
4	N-Body	Red	Grey	Yellow	Grey	Grey	Grey
5	Structured Grid	Red	Red	Red	Grey	Yellow	Grey
6	Unstructured	Red	Grey	Grey	Yellow	Yellow	Grey
7	MapReduce	Red	Grey	Orange	Red	Grey	Red
8	Combinational	Grey	Red	Grey	Orange	Grey	Orange
9	Nearest Neighbor	Grey	Grey	Grey	Yellow	Grey	Red
10	Graph Traversal	Grey	Red	Yellow	Red	Yellow	Yellow
11	Dynamic Prog	Grey	Yellow	Grey	Red	Grey	Red
12	Backtrack/ B&B	Grey	Grey	Grey	Red	Grey	Yellow
13	Graphical Models	Grey	Grey	Grey	Red	Grey	Yellow
14	FSM	Grey	Red	Red	Yellow	Yellow	Red

Scale: Nearly 300 combinations

- Foreach (managed, unmanaged)
 - Foreach (mpi, mpi.net, openmp, tpl, hybrid)
 - Foreach (input.small, input.medium, input.large)
 - Foreach (one..thirteenth dwarf)
 - Run, Trace
 - Plot Excel, Xperf
 - Plot Vampir, JumpShot
- Support for mixed models:
 - MPI + Openmp
 - MPI.Net + TPL
 - etc

Use the Parallel Dwarfs for:

- Comparing || technologies
- Comparing language features
- Benchmarking
- Best practices
- Starting templates

```
PS1> DwarfBench -Names SpectralMethod -Size medium -Platform managed -Parallel serial,tpl,mpi -PlotExcel
```

```
PS1> DwarfBench -Names DenseAlgebra -Size medium -Platform unmanaged,managed -Parallel mpi -PlotVmampir
```

```
PS1> DwarfBench -Names *grid* -Size Large -Platform unmanaged -Parallel hybrid -PlotVmampir
```

13 Visual Studio Solutions

StructuredGrid code fragment using MPI.NET

The image shows a Visual Studio IDE with a Solution Explorer on the left, a code editor in the center, and three class diagrams on the right. The code editor displays the Solver.cpp file with the following code fragment:

```

delete grid;
delete tempGrid;
}

void Solver::solve()
{
    bool converged = false;
    bool odd = false;

    int i, j;

    while(!converged) {
        converged = true;
        odd = !odd;

        for (i = 1; i < ymax - 1; i++) {
            for (j = 1; j < xmax - 1; j++) {
                if (odd) {
                    tempGrid[i][j] = (grid[i-1][j] +
                    ) else {
                        grid[i][j] = (tempGrid[i-1][j] +
                        ) if (converged && (abs(tempGrid[i][j]
                    ) }
                }
            }
        }

        result = odd ? tempGrid : grid;
    }
}

```

The three class diagrams on the right are:

- Program Class:** Fields: settings; Methods: getSettings, Main.
- Settings Class:** Fields: BUFFER_SIZE, DWARF_NAME, DWARF_NAME..., inputFile, log, outputFile, PROFILE_TXT, RESULT_TXT, ROOT_PROCESS; Methods: Finish, GetLine, Init, parseArguments, parseConfigFile, Settings, start, Start, stop, usage, WriteSettings.
- Solver Class:** Fields: epsilon, grid, tempGrid, xmax, ymax; Methods: Solve, Solver.

The code fragment in the center shows the MPI.NET implementation of the solve method, including the communication of grid data between processes.

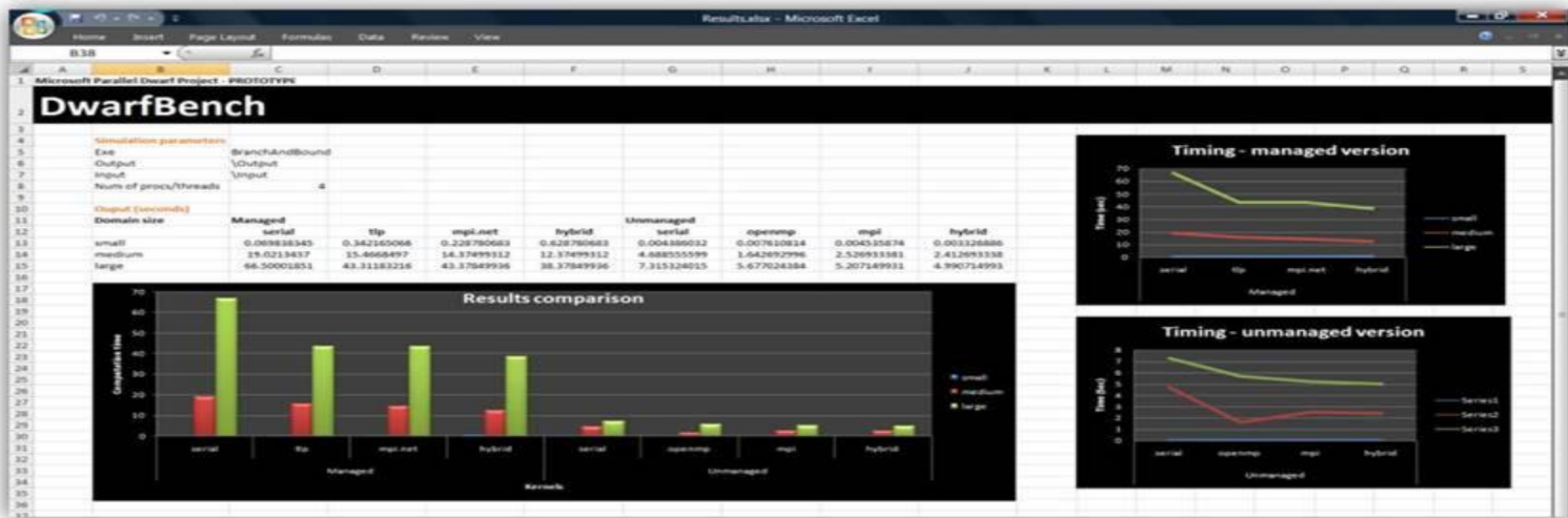
```

if (Communicator.world.Rank == Settings.ROOT_PROCESS)
{
    int count = (ymax - 2) / Communicator.world.Size + 2;
    for (i = 1; i < Communicator.world.Size; i++)
    {
        for (j = 1; j < count - 1; j++)
            Communicator.world.Receive(
                i,
                i + j,
                ref (odd ? grid : tempGrid)[(ymax - 2) %
                Communicator.world.Size + ((ymax - 2) /
                Communicator.world.Size) * i + j]);
    }
}
else
{
    for (i = 1; i < countRows - 1; i++)
        Communicator.world.Send(
            (odd ? grid : tempGrid)[i],
            Settings.ROOT_PROCESS,
            Communicator.world.Rank + i);
}

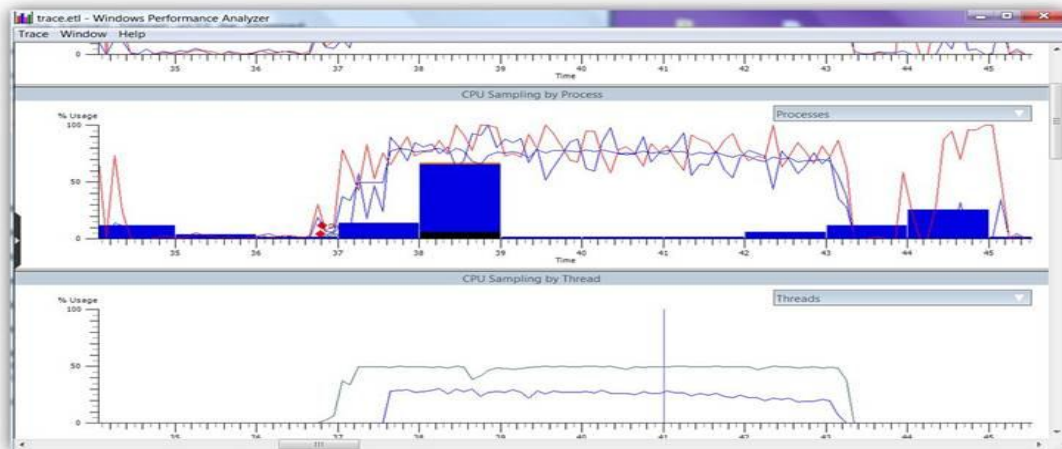
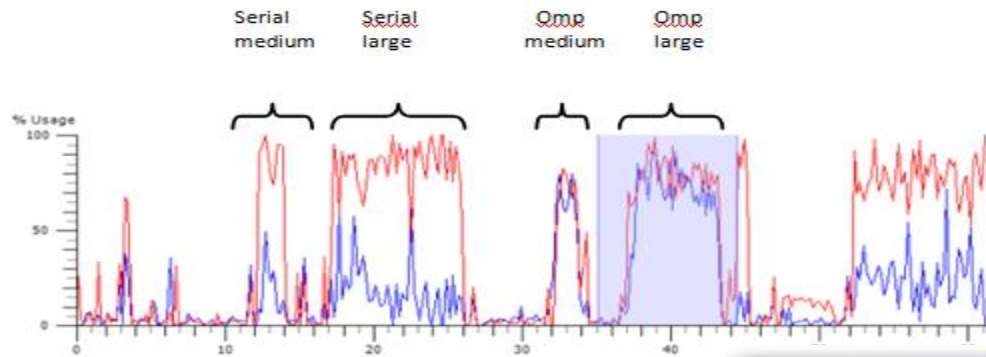
grid = (odd ? grid : tempGrid);

```

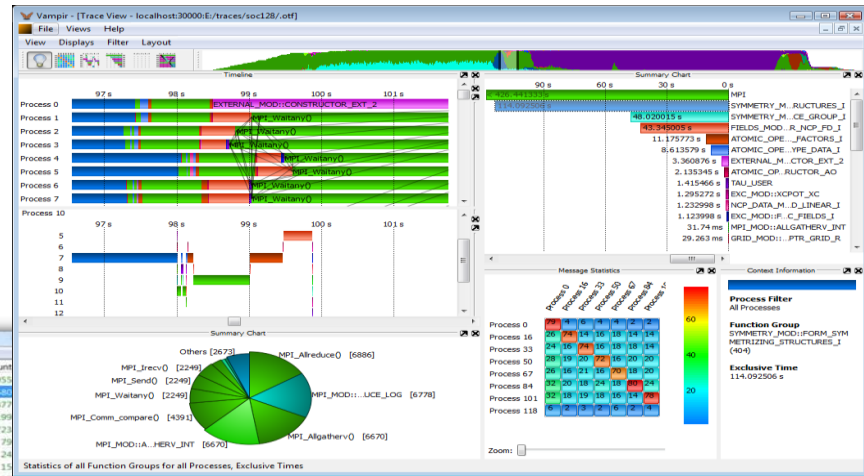
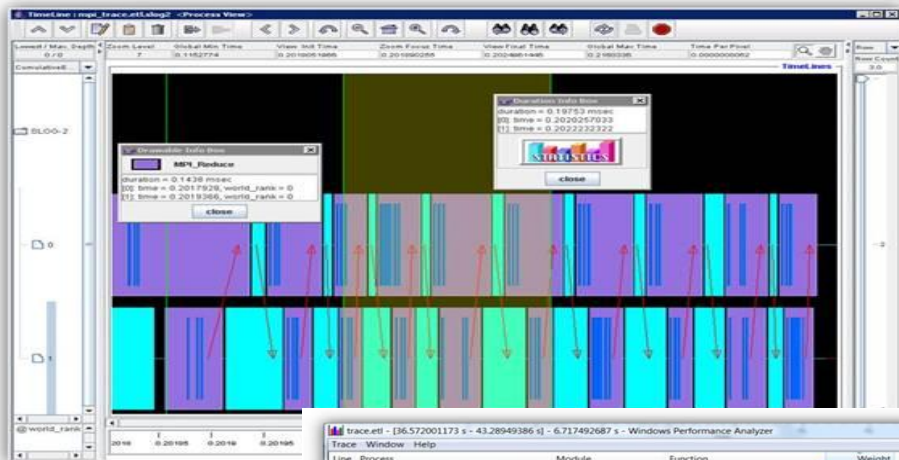
Results Summary: -PlotExcel



Xperf: -PlotXperf



JumpShot: -PlotJumpshot -PlotVampir



trace.exe - [36.572001173 s - 43.28949386 s] - 6.717492687 s - Windows Performance Analyzer

Line	Process	Module	Function	Weight	% Weight	Count
1	Idle	Dwarf.Unmanaged.Omp.exe (1704)	Unknown	8979.140299	66.83	9051
2		Dwarf.Unmanaged	Unknown	5372.51174	4.84	524
3		ntoskrnl.exe	Unknown	1362.383990	10.14	1371
4		ntdll.dll	Unknown	1176.816322	8.76	1199
5		msvc90d.dll	Unknown	696.036322	5.18	723
6		vcomp90d.dll	Unknown	79.019051	0.59	79
7		Unknown	Unknown	22.473111	0.17	24
8		NETwv64.sys	Unknown	14.998271	0.11	11
9		atikmdag.sys	Unknown	11.982878	0.09	10
10		hal.dll	Unknown	10.003715	0.07	10
11		wow64.dll	Unknown	3.004757	0.04	5
12		dxgkml.sys	Unknown	3.001893	0.04	5
13		tcpip.sys	Unknown	4.994608	0.04	5
14		kernel32.dll	Unknown	3.221638	0.02	4
15		wow64cpu.dll	Unknown	3.219055	0.02	4
16		spq.sys	Unknown	2.001092	0.01	2
17		PCIDEX.SYS	Unknown	1.998857	0.01	2
18		usbhub.sys	Unknown	1.998788	0.01	2
19		fltmgr.sys	Unknown	1.220197	0.01	2
20		interppm.sys	Unknown	1.003829	0.01	1
21		ataport.sys	Unknown	0.999149	0.01	1
22		USBPORT.SYS	Unknown	0.998730	0.01	1
23		Nfs.sys	Unknown	0.880909	0.01	2
24		Idle (0)		4018.309357	29.91	3733
25		pschost.exe (936)		130.951628	0.97	131
26		devcon.exe (4660)		84.157205	0.63	86
				Total CPU Usage (Non-Idle) - 70.09%		

Parallel Computing on Azure

- Demo:

Processing astronomical images for a TeraPixel panorama

The 10 Parsec Overview

Collect

Process

View

4TB of raw images

2 X 1791 X 0.5 gigapixel plates

Palomar (US) and UK Schmidt (Australia)
telescopes

Raw images have many artifacts

Data access is challenging

Local + Cloud clusters

Vignetting correction

Astrometric alignment

De-vignetting & color correction

Stich & Smooth

Final rendered image

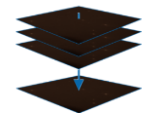
2^{40} (=1,099,511,627,776)
pixels (RGB)

Smooth, zoomable

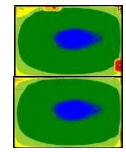
Silverlight or Desktop viewers

Accessible by everyone

Vignetting correction



			0.9	1					
		1	0.9						
		0.9	0.9	0.9	0.9				
		0.7	0.8	0.8	0.8	0.8	0.8	0.8	



Creating Flat Fields

Normalization Matrix

Normalizing Corners

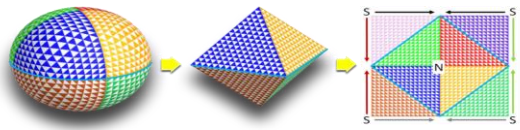
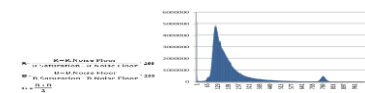
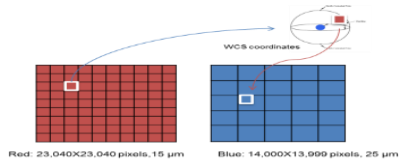
Color Plates Generation

Devignetting

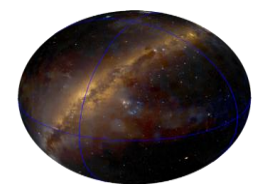
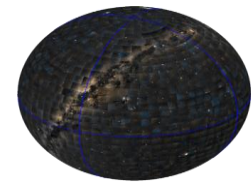
Astrometric Alignment

Color Correction

Stitching & Smoothing



&



Vignetting Correction

- Input: 1791 X 2 images; Output: matrix of correction factors
- Code: Parallel C#, C++, DryadLinq (Map/Reduce)
- HW: Local Cluster



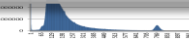
Creating Flat
Fields

Normalization
Matrix

Normalizing
Corners

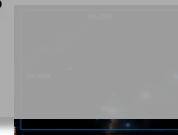
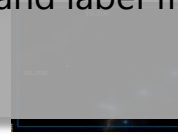
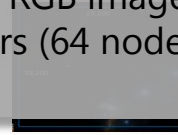
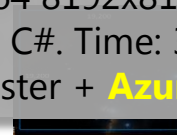
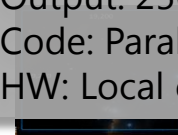
Color Plates Generation

- Input: 1791x2 images (417GB compressed, 4TB uncompressed) ; Output: 1791 color plate files (790GB)
- Code: Parallel C#, C++. Time: 5Hrs (64 nodes).
- HW: Local Cluster + **Azure**



Stitching

- Output: 256x64 8192x8192 RGB images and label files
- Code: Parallel C#. Time: 3Hrs (64 nodes)
- HW: Local cluster + **Azure**



Smoothing

- Final output: 1025 image pyramids (802G);
- Code: C++, MPI, Parallel C#; Time: Smooth=4Hrs; Zoom=40m
- HW: Local Cluster (64 nodes); 2.5 Hrs to move final data off cluster (1Gpbs)



- Would require 500,000 HDTVs to view
- Stretched out, would fill an American football field

Technical Details/Demo

- Acquisition & node mgmt on Azure
- Visual Studio : DryadLinq
- Visual Studio : .Net Parallel Extensions
- Visual Studio : MPI

The DryadLinq Query

```
DryadLinq.UsePLINQ = false;
DryadLinqTools.RemoveUnwantedDllsFromResourceSet();

var pixelRows = folders.SelectMany(image => ImageToRows(image, options));
var stackedPixelRows = pixelRows.GroupBy(pixelRow => pixelRow.Position);
var finalRows = stackedPixelRows.Select(x => ReduceStackedRows(x));
var b = finalRows.Apply(x => SaveFlatField(x, options));
return b.Single();
```

.Net Parallel Extensions

```
Parallel.For(0, plate.Height, (y) =>
{
    //Flip the y-index because a DSS plate has y=0 at the bottom of
    //the image but a bitmap has the origin at the top
    int iy = plate.Height - 1 - y;
    int pos = y * imageData.Stride;
    for (int ix = 0; ix < plate.Width; ix++)
    {
        Color c = scale.Map(plate.Data[iy][ix]);
        rgb[pos++] = c.B;
        rgb[pos++] = c.G;
        rgb[pos++] = c.R;
    }
});
```


What's actually nicer on windows for TC work

(in my humble opinion)

- Visual Studio, C#, F#, C++ IDE
- MPI.net
- Graphics & GPGPU drivers, Nvidia Nsight, ...
- Python MPI
- CAS packages
- TC / Domain Specialist support: eg run Excel at scale
- HPC cluster setup / mgmt / multi-discipline usage
- Intel & PGI's compiler integration
- .Net in some ways > JVM

Conclusion...

- If you haven't already, give your Windows box some love!
- Leverage your investments: tools, code, muscle memory, ...
- Cross-platform languages/runtimes enable kicking the tires without lock-in

Winux™ : The best features of Linux & Windows !



Microsoft[®]

© 2010 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.
The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.
MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

Backup screen shots

Python Tools for Visual Studio

Free/OSS

The screenshot displays the Visual Studio interface for a project named 'PythonMPI1'. The 'Debug' configuration is selected, and the 'Launch mode' is set to 'MPI Cluster launcher'. The 'Run Environment' section shows the following configuration:

- Run Environment: LAYTON00.redmond.corp.microsoft.com/8/Cc
- Target platform: X64
- Publish before Run: True
- Python Interpreter: C:\Python26_x64\python.exe
- Interpreter Arguments: (empty)
- Deployment Directory: (empty)
- Working Directory: (empty)
- MPIExec Command: (empty)

The 'Cluster Selector' dialog box is open, showing the following settings:

- Head Node: LAYTON00.redmond.corp.microsoft.com
- Number of processes: 8
- Schedule one process per: Core
- Pick nodes from: ComputeNodes
- Manually select nodes to include in the allocation

The dialog box contains a table of available nodes:

Node	CPU (MHz)	Memory (MB)	Cores	State
<input type="checkbox"/> LAYTON00	2992	16383	8	Online
<input type="checkbox"/> LAYTON04	3000	16383	8	Online
<input type="checkbox"/> LAYTON06	3000	16383	8	Online
<input type="checkbox"/> LAYTON03	3000	16383	8	Online
<input type="checkbox"/> LAYTON02	3000	16383	8	Online
<input type="checkbox"/> LAYTON07	3000	16383	8	Online
<input type="checkbox"/> LAYTON05	3000	16383	8	Online

The 'Output' window at the bottom shows 'Show output from:' and is currently empty. The system tray at the bottom right indicates the time is 8:41 PM on 5/17/2011, and the taskbar shows 'Python 2.6 I...', 'Solution...', and 'Team Ex...'.

Burst to Azure

Cluster JEREMYHPC-DEV - HPC 2008 R2 Cluster Manager

File View Actions Options Go Help

Back Forward Navigation Pane Actions Filter: By Group By Health Search: Node Name Clear All

Node Management

Nodes (41)

- By Node Health
 - OK (41)
 - Warning (0)
 - Error (0)
 - Transitional (0)
 - Unapproved (0)
- By Node State
 - Online (41)
 - Offline (0)
 - Draining (0)
 - Provisioning (0)
 - Rejected (0)
 - Not-Deployed (0)
- By Group
 - HeadNodes
 - ComputeNodes
 - WCFBrokerNodes
 - WorkstationNodes
 - AzureNodes
 - AzureWorkerNodes
 - Terapixel
- By Node Template
 - Azure Worker Template
 - Default AzureNode Template
 - Default ComputeNode Template
 - HeadNode Template
- By Location
- Pivot View
- Operations
 - Archived
 - Committed
 - Executing
 - Failed

Configuration

Node Management

Job Management

Diagnostics

Charts and Reports

Nodes (41)

List Heat Map

Node	CPU Usage (%)	Disk Throughput (Bytes/second)
AzureCN-0007	95.85	12424.02
AzureCN-0008	95.25	13657.34
AzureCN-0009	95.54	9774.12
AzureCN-0010	95.52	12010.93
AzureCN-0011	94.90	15785.66
AzureCN-0012	95.35	14956.97
AzureCN-0013	95.11	10960.28
AzureCN-0014	95.60	15435.26
AzureCN-0015	95.52	11272.86
AzureCN-0016	95.46	15580.53
AzureCN-0017	94.97	13304.02
AzureCN-0018	95.44	10215.57
AzureCN-0019	95.54	16089.92
AzureCN-0020	95.43	13585.52
AzureCN-0021	95.24	10065.13
AzureCN-0022	94.63	15715.05
AzureCN-0023	95.55	15367.84
AzureCN-0024	95.02	11602.30
AzureCN-0025	95.65	9724.08
AzureCN-0026	95.16	15041.91
AzureCN-0027	95.07	15690.97
AzureCN-0028	95.02	12853.03
AzureCN-0029	94.80	14767.85
AzureCN-0030	95.77	15181.49
AzureCN-0031	95.45	14706.52
AzureCN-0032	95.37	14095.37
AzureCN-0033	95.30	14791.04
AzureCN-0034	95.35	13683.23
AzureCN-0035	95.27	10648.85
AzureCN-0036	95.58	9683.72
AzureCN-0037	94.51	16787.41
AzureCN-0038	95.17	13820.76
AzureCN-0039	94.48	12716.35
AzureCN-0040	95.70	13547.64
AzureCN-0041	95.53	15983.25

Node AzureCN-0010 CPU Usage (%) 95.55 Disk Throughput (Bytes/second) 14276.26 33%

Actions

Pivot To

Jobs for the Selected Nodes

Failed Diagnostics for the Nodes

Operations for the Nodes

Node Actions

- Bring Online
- Take Offline
- Start
- Stop
- Reboot
- Shut Down
- Run Command ...
- Add Node ...
- Reimage
- Maintain
- Change Role ...
- Delete
- Reject
- Assign Node Template ...
- Edit Properties...
- Export Node XML File ...
- Run Diagnostics ...
- View Performance Charts
- Remote Desktop

Tab Actions

- New Tab
- Customize Tab...
- Delete Tab

Help Resources

- Node Management
- Node Health and States
- Grouping Nodes
- Applying Software Updates

Data updated: 5/17/2011 3:47:06 PM

Development

- Porting vs developing new code
- Examine headers, code, platform dependencies
- UI layer
- Support libraries
- make
- Compile/link/build, debug, iterate
- Profile, optimize
- Package & Deploy