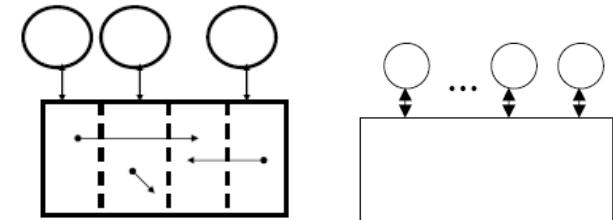
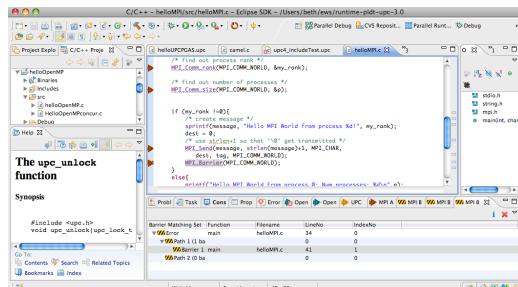
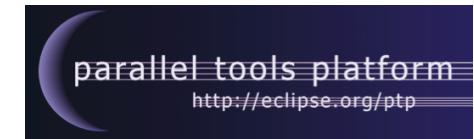




# UPC and OpenMP Parallel Programming and Analysis in PTP with CDT



Beth R. Tibbitts  
IBM Corp.  
[tibbitts@us.ibm.com](mailto:tibbitts@us.ibm.com)  
March, 2010



## EclipseCon 2010, Santa Clara

This material is partly based upon work supported by the Defense Advanced Research Projects Agency under its Agreement No. HR0011-07-9-0002



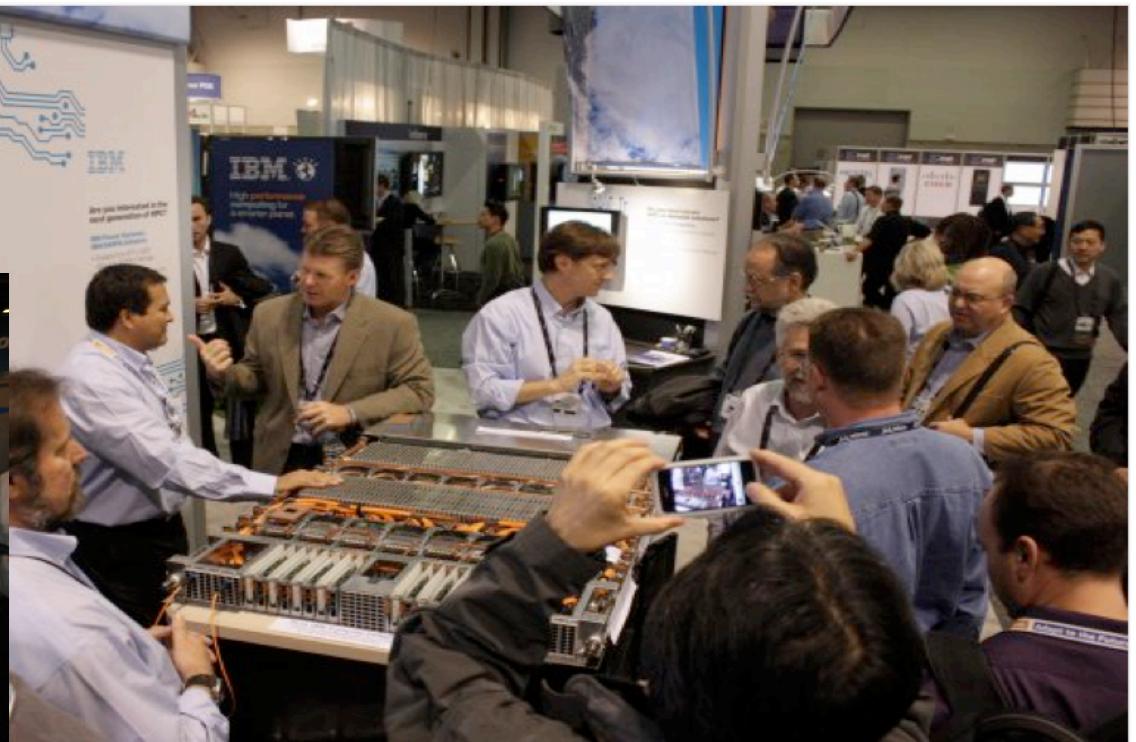
# IBM Power7 and Blue Waters at Supercomputing '09



IBM P7: 8 cores=1 chip; 4 chips=1 QCM; 8 QCMs= 1 drawer  
4 drawers= 1 supernode (1024 cores)

Total 300,000 cores; 10 petaflops peak; 1 petaflop sustained

Source: <http://www.ncsa.illinois.edu/BlueWaters/pdfs/snir-power7.pdf>



# NCSA Blue Waters



facebook 1 10 Search

Links on "National Center for Supercomputing Applications - NCSA"

**NCSA** Back to National Center for Supercomputing Applications - NCSA

Displaying 41 - 50 out of 117 links.

**National Center for Supercomputing Applications - NCSA** It's all about the science. Read some quick takes on how prospective users of the Blue Waters sustained-petascale computing system will improve our world and our understanding of it.

**What will scientists do with Blue Waters?**  
[www.ncsa.illinois.edu](http://www.ncsa.illinois.edu)

Many scientists are working now with the Blue Waters team so they are ready to use the massive sustained-petaflop supercomputer when it comes online in 2011. These teams will use Blue Waters to improve ...

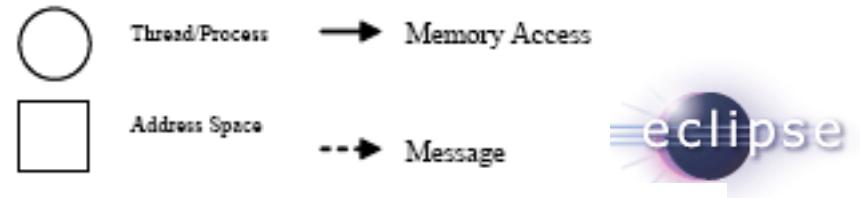
December 16, 2009 at 10:30am · Comment · Like · Share · Report

PTP is the basis for NCSA BW  
HPC workbench  
How to program a massively  
parallel machine?

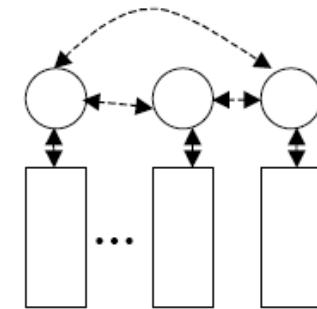


PTP kiosk at IBM booth at SC09

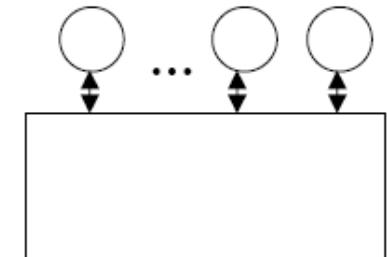
## Parallel Programming Models



1. Message passing - Most parallel programming so far  
Language: MPI (C/C++/Fortran)

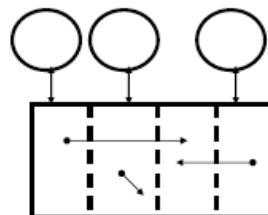


2. Shared Memory (Global Address Space)  
Language: C/Fortran with OpenMP



*Hybrid: Shared + Msg passing: MPI+OpenMP*

3. Partitioned Global Address Space (PGAS)  
Languages: UPC, X10, Chapel, CAF



Berkeley Unified Parallel C (UPC) Project

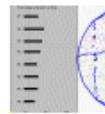
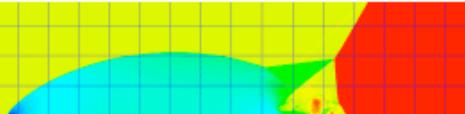
http://upc.lbl.gov/ Google

Maps of Jerusalem Google Calendar Google Reader Gmail - Contacts WA4DAC findu aprs.fi PTP Wiki >

  **Berkeley UPC - Unified Parallel C** (A joint project of LBNL and UC Berkeley) 

[Home](#) [Downloads](#) [Documentation](#) [Bugs](#) [Publications](#) [Demos](#) [Contact](#) [Internal](#)

*NEW -- Berkeley UPC version 2.10.0 released!*

 **The UPC Language**    [About this UPC program](#) [About this UPC program](#)

Unified Parallel C (UPC) is an extension of the C programming language designed for high performance computing on large-scale parallel machines. The language provides a uniform programming model for both shared and distributed memory hardware. The programmer is presented with a single shared, partitioned address space, where variables may be directly read and written by any processor, but each variable is physically associated with a single processor. UPC uses a Single Program Multiple Data (SPMD) model of computation in which the amount of parallelism is fixed at program startup time, typically with a single thread of execution per processor.

In order to express parallelism, UPC extends ISO C 99 with the following constructs:

- [An explicitly parallel execution model](#)
- [A shared address space](#)
- [Synchronization primitives and a memory consistency model](#)
- [Memory management primitives](#)



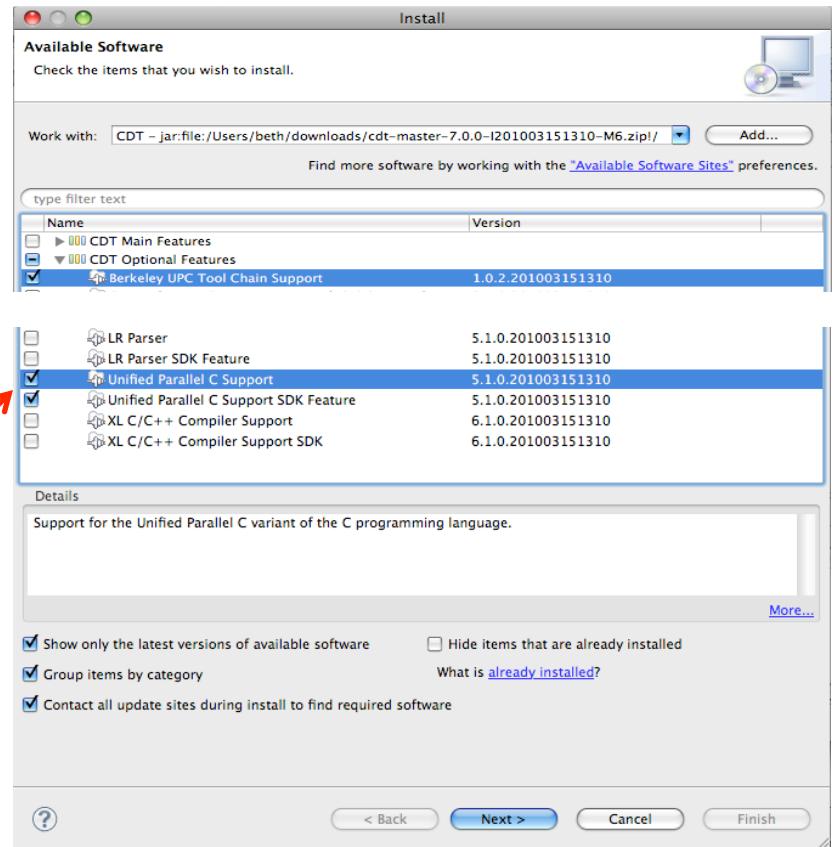
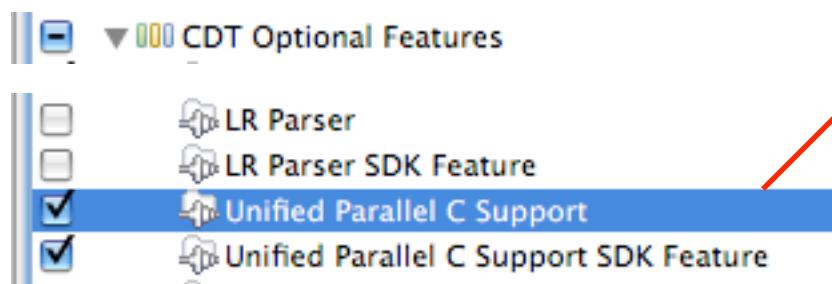
# UPC: Installation from CDT



C/C++ Development Tools: [eclipse.org/cdt](http://eclipse.org/cdt)

## Installation:

Must install from  
CDT Optional features  
(not in main update site  
or in C/C++ IDE )

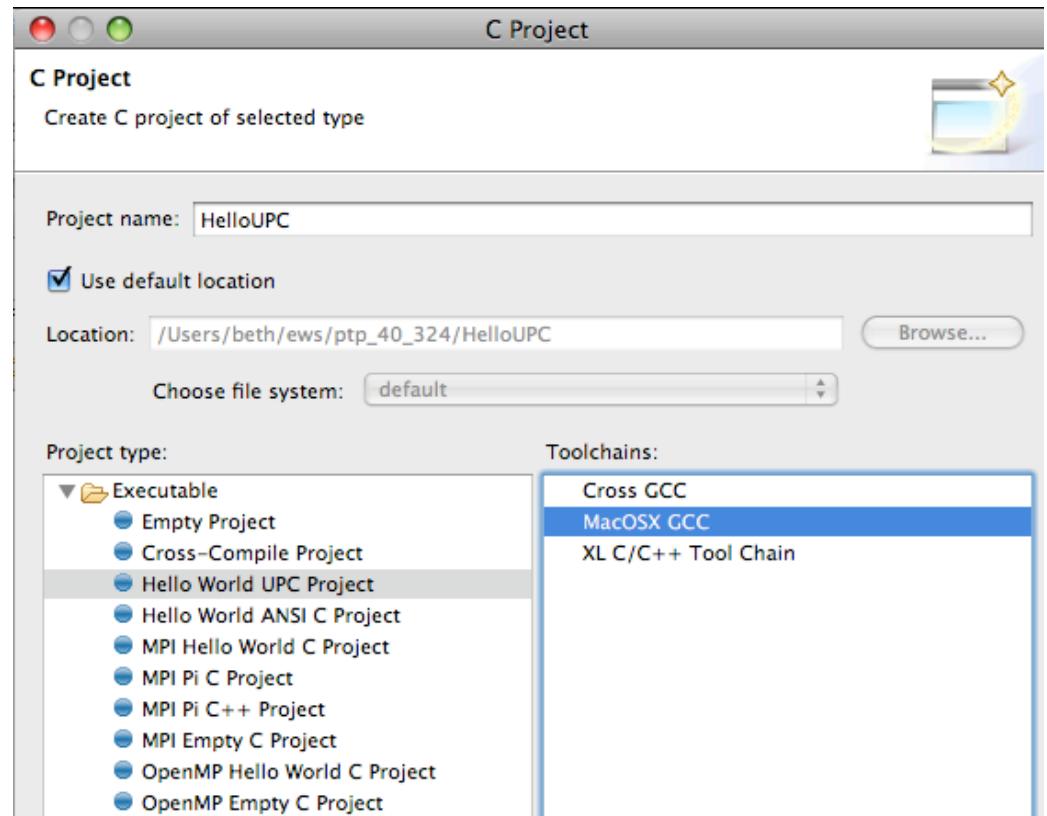


# UPC New Project Wizard



New in CDT 7.0  
Berkeley UPC

IBM xIC includes UPC support  
and is already in CDT.



# UPC Syntax Awareness in CDT Editor



Without Language  
Mapping setting  
for UPC

```
#include <upc.h>
#define NUMPAIRS 200000
int main(int argc, char *argv[]) {
    int input;
    printf("Hello, I am %d of %d.\n", MYTHREAD, THREADS);
    upc_forall(input = 0; input < NUMPAIRS; input++; input)
        // grab all crypts that match up
        docrypt(key32, input, 0, &R1X, &R1Y, &C, &C2); // perform 2
```

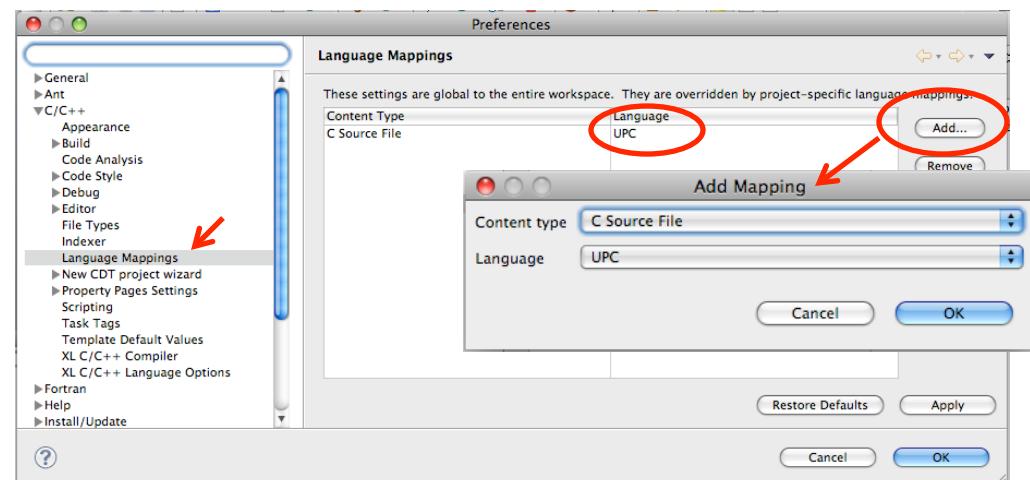
```
#include <upc.h>
#define NUMPAIRS 200000
int main(int argc, char *argv[]) {
    int input;
    printf("Hello, I am %d of %d.\n", MYTHREAD, THREADS);
    upc_forall(input = 0; input < NUMPAIRS; input++; input)
        // grab all crypts that match up
        docrypt(key32, input, 0, &R1X, &R1Y, &C, &C2); // perform 2
```

With Language  
Mapping setting  
for UPC:  
UPC Syntax  
awareness

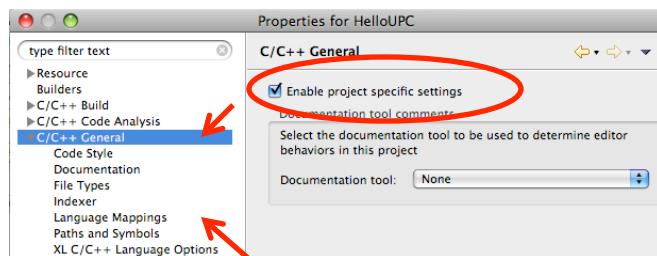
## UPC content types



UPC syntax awareness on \*.upc files - To recognize it on \*.c files, modify Language Mappings in Preferences



Or enable as project-specific setting in Project Properties



Then change Language Mappings here

## UPC Demo



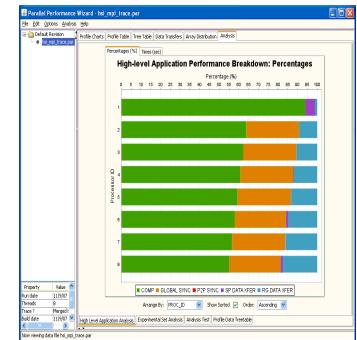
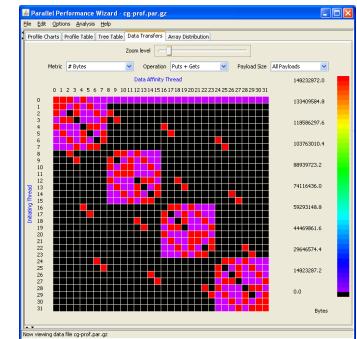
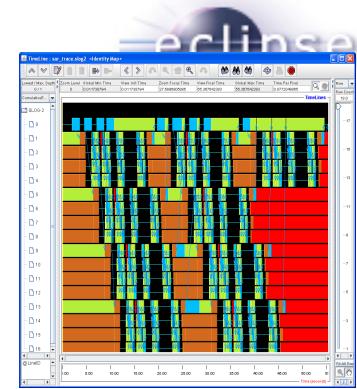
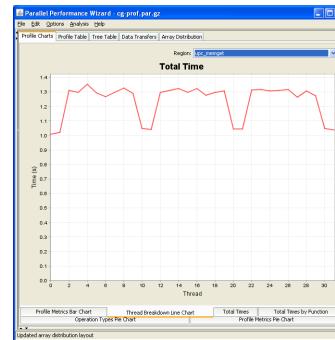
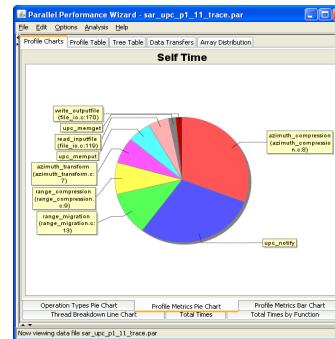
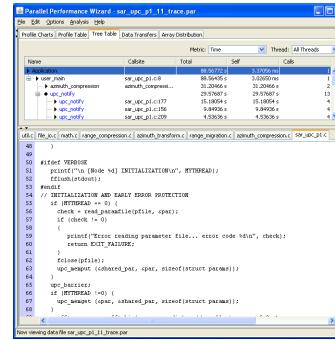
- Project setup: New project, Language Mapping
- Content Assist: added descriptions
- Hover Help
- Code Templates

Run: via PTP Parallel Runtime

# Parallel Performance Wizard (PPW)

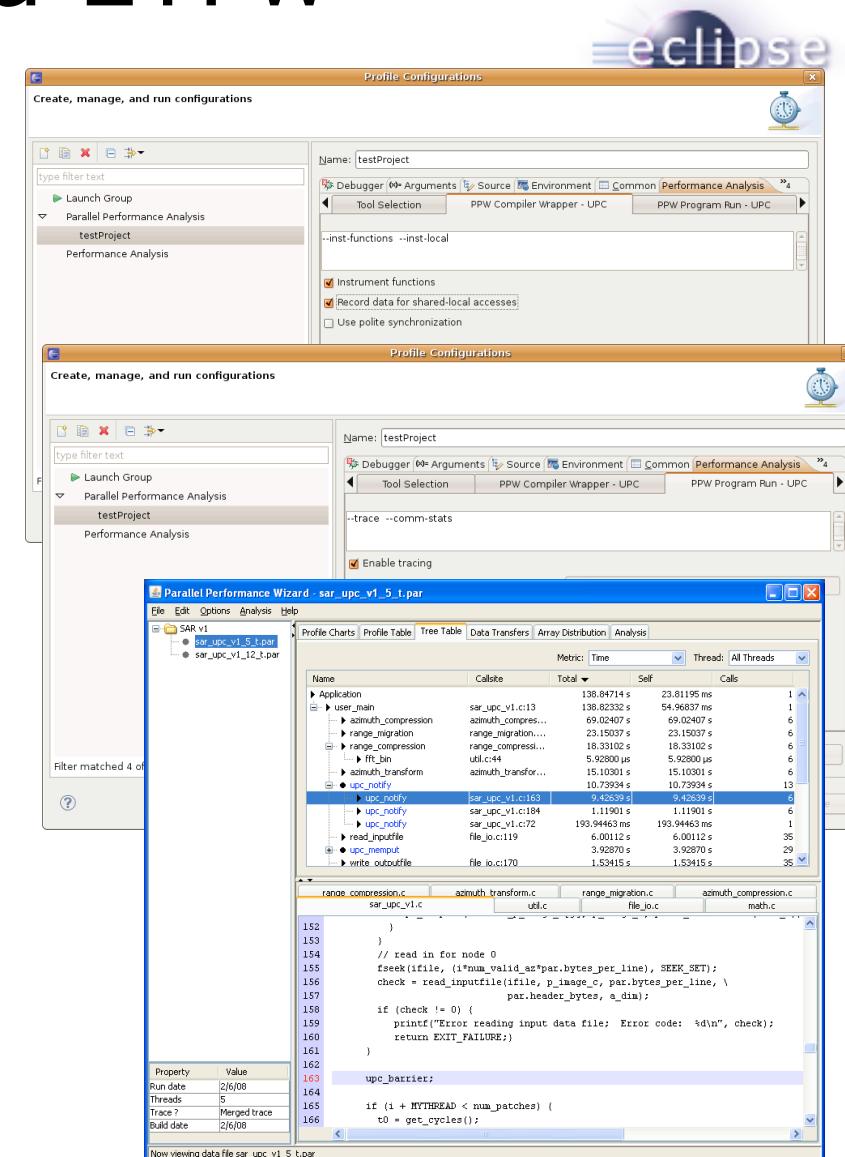
- ◆ Full-featured performance tool for PGAS programming models
    - ◆ Currently supports UPC, SHMEM, and MPI
    - ◆ Extensible to support other models
    - ◆ PGAS support by way of Global Address Space Performance (GASP) interface (<http://gasp.hcs.ufl.edu>)

- ◆ PPW v2.4 features:
    - ◆ Easy-to-use scripts for backend data collection
    - ◆ User-friendly GUI with familiar visualizations
    - ◆ Advanced automatic analysis support
    - ◆ Eclipse Integration with PTP 4.0
  - ◆ More information and free download: <http://ppw.hcs.ufl.edu>



# PPW Integration via ETFw

- ❖ We implement the ETFw to make PPW's capabilities available within Eclipse
  - ❖ Compile with instrumentation, parallel launch with PPW
  - ❖ Generates performance data file in workspace, PPW GUI launched
- ❖ PPW is often used for UPC application analysis
  - ❖ ETFw extended to support UPC
  - ❖ Many UPC features in PTP
- ❖ For more information:
  - ❖ <http://ppw.hcs.ufl.edu>
  - ❖ [ppw@hcs.ufl.edu](mailto:ppw@hcs.ufl.edu)

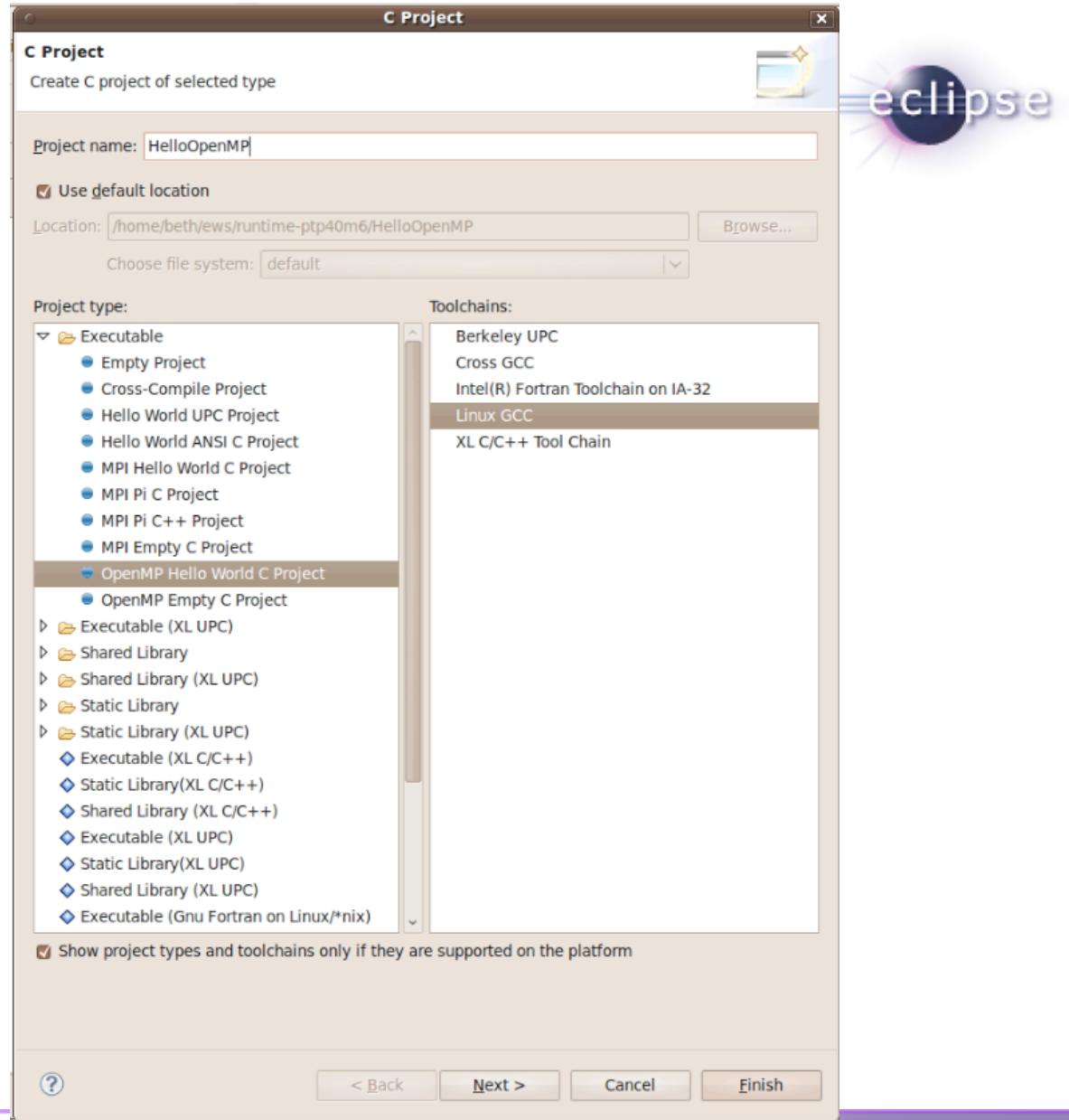


# OpenMP support in PTP



OpenMP is a set of directives and APIs/library on top of C  
Makes suggestions to compiler for what can be parallelized  
Often used in Multicore development

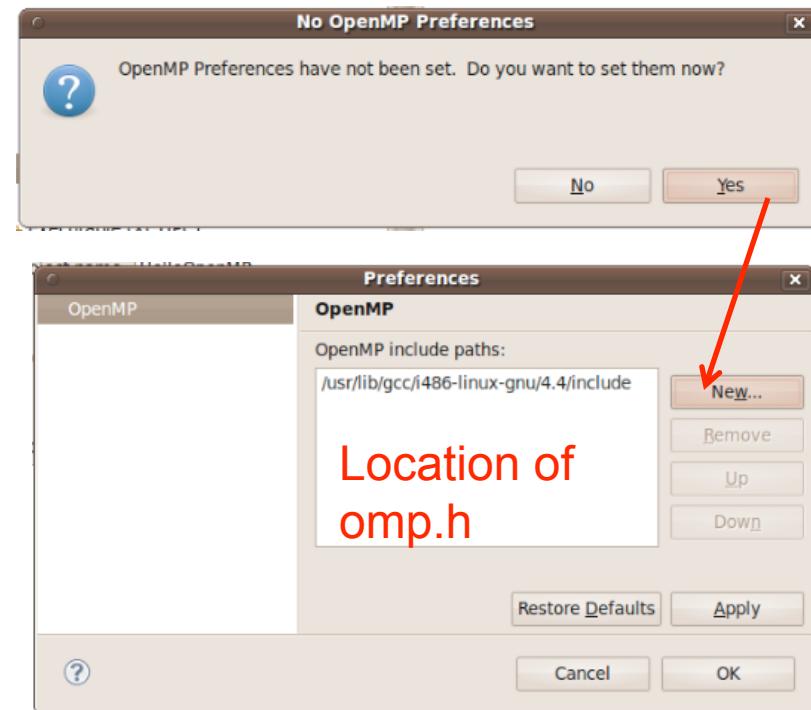
# New OpenMP project



## Set OpenMP Preferences once



Point at OpenMP header file  
Allows recognition  
of artifacts



# OpenMP project builds

A screenshot of the Eclipse CDT (C/C++ Development Tooling) interface. The window title is "C/C++ - HelloOpenMP/src/HelloOpenMP.c - Eclipse SDK - /home/beth/ews/runtime-ptp40m6b".

- Project Explorer:** Shows the project structure: "HelloOpenMP" (Binaries, Includes, src), "src" (HelloOpenMP.c), "Debug" (src, HelloOpenMP - [x86/le], makefile, objects.mk, sources.mk).
- Editor:** Displays the source code of "HelloOpenMP.c". The code includes #include <omp.h>, #include <stdio.h>, #include <stdlib.h>. It contains a main function that prints the number of threads and the current thread ID using #pragma omp parallel private(numThreads, tid) and #pragma omp for.

```
/*
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
*/
/* Hello OpenMP World prints the number of threads and the current thread
 */
int main (int argc, char *argv[]) {
    int numThreads, tid;

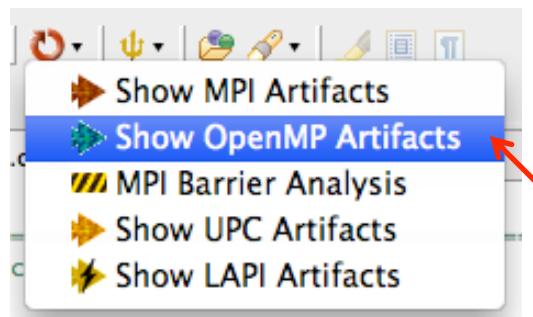
    /* This creates a team of threads; each thread has own copy of variable
    */
# pragma omp parallel private(numThreads, tid)
    {
        tid = omp_get_thread_num();
        printf("Hello World from thread number %d\n", tid);
    }
}
```
- Console:** Shows the build log:

```
**** Build of configuration Debug for project HelloOpenMP ****

make all
Building file: ../src>HelloOpenMP.c
Invoking: GCC C Compiler
gcc -fopenmp -I/usr/lib/gcc/i486-linux-gnu/4.4/include -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"src>HelloOpenMP.d" -MT"src>HelloOpenMP.d" -o"src>HelloOpenMP.o" "../src>HelloOpenMP.c"
Finished building: ../src>HelloOpenMP.c

Building target: HelloOpenMP
Invoking: GCC C Linker
gcc -fopenmp -o"HelloOpenMP" ./src>HelloOpenMP.o
Finished building target: HelloOpenMP
```

# Show OpenMP Artifacts



The screenshot shows the Eclipse CDT interface with the HelloOpenMP.c file open in the editor. The code prints the number of threads and the current thread ID. The OpenMP Artifact view below the editor shows three entries:

OpenMP Artifact	Filename	LineNo	Construct
omp_get_thread_num	HelloOpenMP.c	23	Function Call
omp_get_num_threads	HelloOpenMP.c	29	Function Call
#pragma omp parallel private(numThre	HelloOpenMP.c	21	OpenMP Pragma

Hover help  
F1 help  
Content assist  
Code templates  
... all especially for OpenMP

# OpenMP Artifacts



Look! It even works in Fortran!

Integration with  
Photran in  
PTP 4.0

Demo

The screenshot shows the Eclipse IDE interface with several open windows:

- Editor View:** Displays the source code of a Fortran file named `HelloOpenMP.f90`. The code is a sample OpenMP program that prints the thread ID and total number of threads.
- OpenMP Artifact View:** A table view showing the detected OpenMP artifacts. The table has columns: OpenMP Artifact, Filename, LineNo, and Construct.

OpenMP Artifact	Filename	LineNo	Construct
omp_get_thread_num	HelloOpenMP.c	23	Function Call
omp_get_num_threads	HelloOpenMP.c	29	Function Call
#pragma omp parallel private(numThreads, id)	HelloOpenMP.c	21	OpenMP Pragma
PARALLEL PRIVATE(NUM_THREADS, ID)	openMP.f90	4	value is 3
OMP_GET_THREAD_NUM	openMP.f90	5	Function Call
OMP_GET_NUM_THREADS	openMP.f90	8	Function Call
END PARALLEL	openMP.f90	11	value is 3

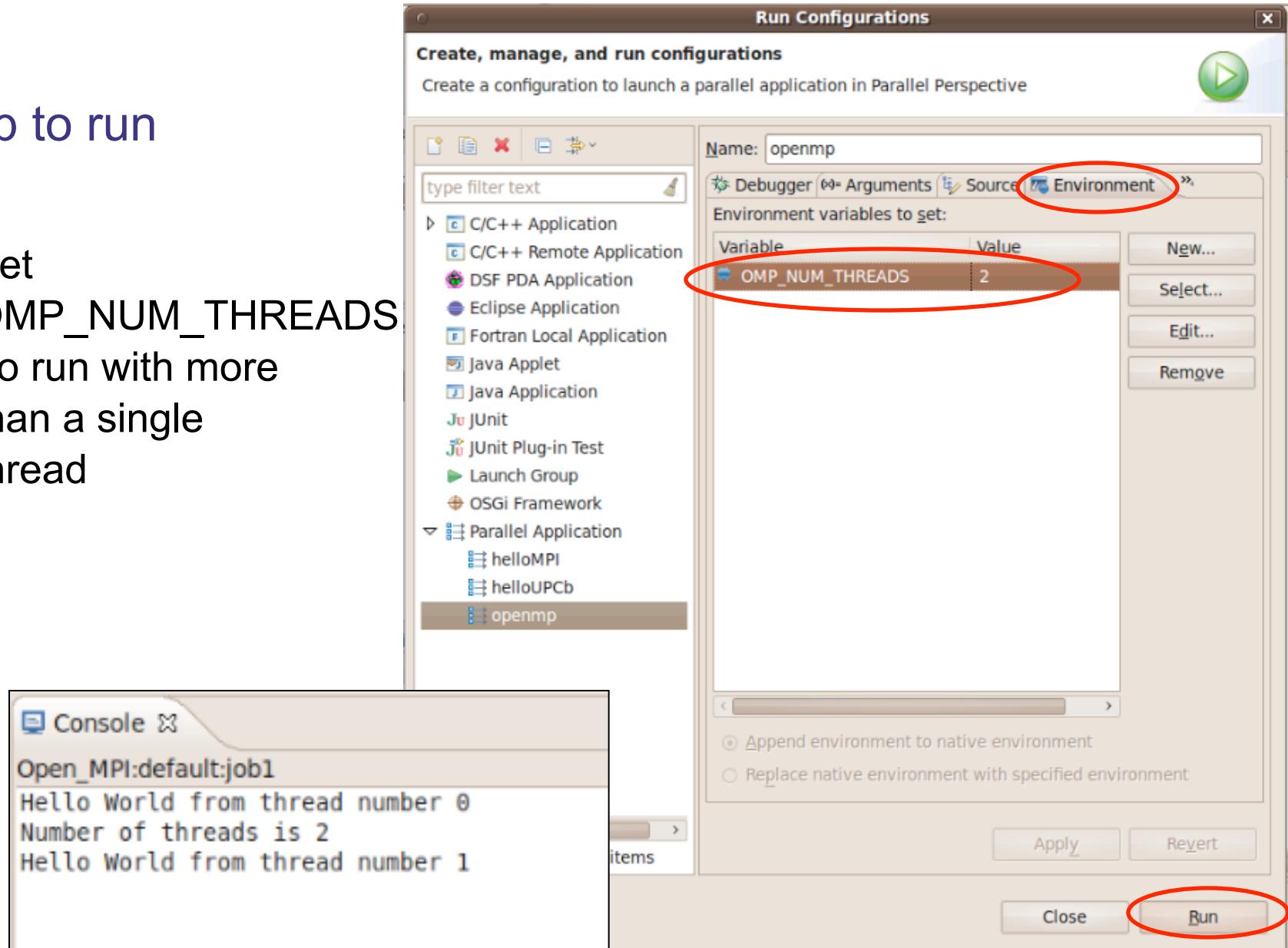
# OpenMP support in PTP



- Need to update OpenMP API info to – 3.0?
- Do multicore shared memory programmers find these tools useful?
- Additional analysis tools
  - Show concurrency – identifies statements that cannot execute concurrently
  - Show #pragma region – scope of the #pragma statements

## Setup to run

Set  
`OMP_NUM_THREADS`  
To run with more  
than a single  
thread



## Issues, Problems, future areas



PTP will be in Helios Release  
Need feedback!  
Debugging



PTP Talk later today:

“The Parallel Tools Platform: A Development Environment For  
High Performance Computing”  
Greg Watson (IBM), Thursday 3:20 PM, Winchester Room

# References



- Parallel Tools Platform: <http://eclipse.org/ptp>
- Beth Tibbitts, "Static Analysis in PTP with CDT", EclipseCon 2008, <http://www.eclipsecon.org/2008/?page=sub/&id=373>
- UPC (Unified Parallel C) <http://upc.lbl.gov/>
  - Implementations include: IBM xlUPC, Berkeley UPC, GCC UPC
- PPW (Parallel Performance Wizard) for PGAS languages: <http://ppw.hcs.ufl.edu/>
- UPC setup in CDT:  
<http://www.eclipse.org/ptp/documentation/3.0/org.eclipse.ptp.pldt.help/html/runUPC.html>
- IBM X10 language for parallel development: <http://x10-lang.org>
  - Includes X10DT Eclipse tooling – based on IMP (<http://eclipse.org/imp>)
- Blue Waters info: <http://www.ncsa.illinois.edu/BlueWaters/pdfs/snir-power7.pdf>
  - <http://www.facebook.com/NCSAatIllinois>