Directions for CS455: Introduction to High Performance Computing

Running an OpenMP Program on UIC Systems Servers

This guide explains compiling and running an OpenMP program on **UIC Systems** servers (`systems1`, `systems2`, `systems3`, and `systems4.cs.uic.edu`).

1. After connecting to any of the servers via SSH, create a sample program, for example, *ompRanks.cc*:

```cpp
#include <iostream>
#include <omp.h>

int main() {
    // Parallel region with OpenMP
    #pragma omp parallel
    {
        int threadId = omp_get_thread_num();
        int numThreads = omp_get_numThreads();
        // Print thread information
        std::cout << "Hello from thread " << threadId << " out of " << numThreads << std::endl;
    }
    return 0;
}
```

2. Create a Makefile:

```
CXX = g++
CXXFLAGS = -std=c++17 -Wall -fopenmp

# Source files and targets
SOURCES = ompRanks.cc
TARGETS = $(SOURCES:.cc=)

all: $(TARGETS)

%: %.cc
	$(CXX) $(CXXFLAGS) -o $@ $<

clean:
	rm -f $(TARGETS)
```

3. Build the program:

```
make
```

4. Execute the compiled program:

```
./ompRanks
Hello from thread 5 out of 8
Hello from thread 0 out of 8
Hello from thread 4 out of 8
Hello from thread 7 out of 8
Hello from thread 1 out of 8
Hello from thread 3 out of 8
Hello from thread 2 out of 8
Hello from thread 6 out of 8
```

5. To set the number of OpenMP threads:

```
export OMP_numThreads=4
```

6. Run again to observe the effect of setting thread count.

Additional Resources

1. *Chapter07: OpenMP*, R. Robey and Y. Zamora, **Parallel and High Performance Computing**.
2. List of tutorials on OpenMP.