

Лабораторная работа 2: Выполнение заданий под управлением Microsoft Compute Cluster Server 2003

Лабораторная работа 2: Выполнение заданий под управлением Microsoft Compute Cluster Server 2003.....	1
Цель лабораторной работы.....	1
Общая схема выполнения заданий под управлением Microsoft Compute Cluster Server 2003.....	1
Упражнение 1 – Компиляция программы для запуска в CCS 2003.....	3
Задание 1 - Установка Microsoft Compute Cluster Pack SDK.....	4
Задание 2 – Настройка интегрированной среды разработки Microsoft Visual Studio 2005.....	7
Задание 3 – Компиляция параллельной программы в Microsoft Visual Studio 2005.....	10
Упражнение 2 – Запуск последовательной задачи.....	15
Запуск программы через графический пользовательский интерфейс.....	15
Запуск программы с использованием шаблона.....	25
Запуск программы из командной строки.....	29
Упражнение 3 – Запуск параллельного задания.....	31
Упражнение 4 – Запуск множества задач.....	38
Упражнение 5 – Запуск потока задач.....	45
Дополнительное упражнение. Задача определения характеристик сети передачи данных.....	54
Описание характеристик, определяющих производительность сети.....	54
Общая характеристика алгоритма.....	55
Компиляция программы.....	55
Выполнение программы.....	55
Контрольные вопросы.....	60

Для эффективной эксплуатации высокопроизводительных кластерных установок необходимо использовать сложный комплекс программных систем. Долгое время пользователям Windows кластеров приходилось одновременно использовать программное обеспечение нескольких производителей, что могло быть причиной проблем с совместимостью различных программ друг с другом. С выходом Compute Cluster Server 2003 (CCS) можно говорить о том, что компания Microsoft предоставляет полный спектр программного обеспечения, необходимый для эффективной эксплуатации кластера и разработки программ, в полной мере использующих имеющиеся вычислительные мощности.

Цель лабораторной работы

Цель данной лабораторной работы – научиться компилировать и запускать программы под управлением **Microsoft Compute Cluster Server 2003**.

- Упражнение 1 – Компиляция программы для запуска в CCS 2003
- Упражнение 2 – Запуск последовательного задания
- Упражнение 3 – Запуск параллельного задания
- Упражнение 4 – Запуск множества заданий
- Упражнение 5 – Запуск потока задач

Примерное время выполнения лабораторной работы: **90 минут**.

Общая схема выполнения заданий под управлением Microsoft Compute Cluster Server 2003

Для эффективного использования вычислительного ресурса кластера необходимо обеспечить не только непосредственный механизм запуска заданий на выполнение, но и предоставить среду управления ходом выполнения заданий, решающую, в том числе, задачу эффективного распределения ресурсов. Эти задачи эффективно решаются с использованием встроенных в CCS 2003 средств.

Дадим определение важнейшим понятиям, используемым в CCS 2003:

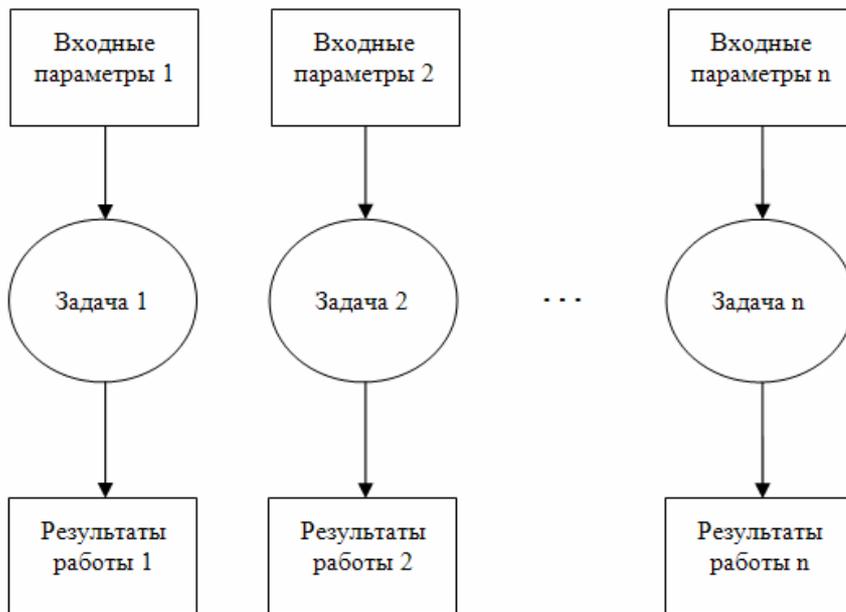
- **Задание (job)** – запрос на выделение вычислительного ресурса кластера для выполнения задач. Каждое задание может содержать одну или несколько задач,
- **Задача (task)** – команда или программа (в том числе, параллельная), которая должна быть выполнена на кластере. Задача не может существовать вне некоторого задания, при этом задание может содержать как несколько задач, так и одну,
- **Планировщик заданий (job scheduler)** – сервис, отвечающий за поддержание очереди заданий, выделение системных ресурсов, постановку задач на выполнение, отслеживание состояния запущенных задач,
- **Узел (node)** – вычислительный компьютер, включенный в кластер под управлением CCS 2003,
- **Процессор (processor)** – один из, возможно, нескольких вычислительных устройств узла,
- **Очередь (queue)** – список заданий, отправленных планировщику для выполнения на кластере. Порядок выполнения заданий определяется принятой на кластере политикой планирования,
- **Список задач (task list)** – эквивалент очереди заданий для задач каждого конкретного задания. Порядок запуска задач определяется правилом FCFS (первыми будут выполнены задачи, добавленные в список первыми), если пользователь специально не задаст иной порядок.

Планировщик заданий CCS 2003 работает как с последовательными, так и с параллельными задачами. Последовательной называется задача, которая использует ресурсы только 1 процессора. Параллельной же называется задача, состоящая из нескольких процессов (или потоков), взаимодействующих друг с другом для решения одной задачи. Как правило, параллельным задачам для эффективной работы требуется сразу несколько процессоров. При этом, в случае использования MPI в качестве интерфейса передачи сообщений, процессы параллельной программы могут выполняться на различных узлах кластера. CCS 2003 включает собственную реализацию стандарта MPI2: библиотеку Microsoft MPI (MS MPI). В случае использования MS MPI в качестве интерфейса передачи сообщений необходимо запускать параллельные задачи с использованием специальной утилиты **mpiexec.exe**, осуществляющей одновременный запуск нескольких экземпляров параллельной программы на выбранных узлах кластера. Важно отметить, что непосредственным запуском задач занимается планировщик, а пользователь может лишь добавить задачу в очередь, так как время ее запуска выбирается системой автоматически в зависимости от того, какие вычислительные ресурсы свободны и какие задания ожидают в очереди выделения им ресурсов. Таким образом, для исполнения программы в CCS 2003 необходимо выполнить следующие действия:

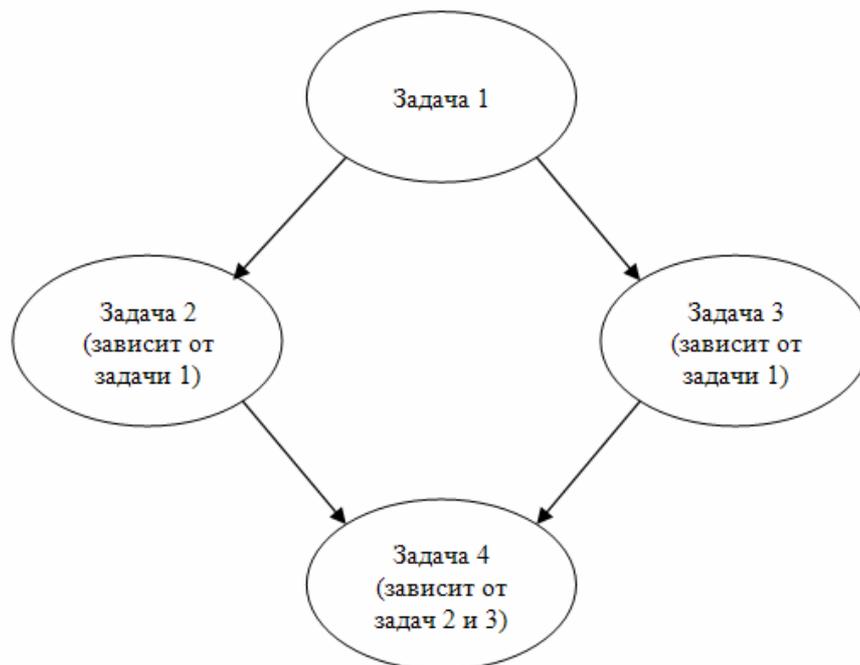
- Создать задание с описанием вычислительных ресурсов, необходимых для его выполнения,
- Создать задачу. Задача определяется при помощи той или иной команды, выполнение которой приводит к запуску на кластере последовательных или параллельных программ. Например, параллельная задача описывается при помощи команды **mpiexec.exe** с соответствующими параметрами (список узлов для ее запуска, имя параллельной программы, аргументы командной строки программы и др.),
- Добавить задачу к созданному ранее заданию.

Выделяют два особых вида заданий:

- **Параметрическое множество задач (parametric sweep)** – одна и та же программа (последовательная или параллельная), несколько экземпляров которой запускается (возможно, одновременно) с разными входными параметрами и разными файлами вывода,



- **Поток задач (task flow)** – несколько задач (возможно, одна и та же программа с разными входными параметрами) запускаются в определенной последовательности. Последовательность запуска объясняется, например, зависимостью некоторых задач последовательности от результатов вычислений предыдущих.



Далее в лабораторной работе на примерах будет показано, как компилировать и запускать последовательные и параллельные задачи в CCS 2003. Кроме того, будут приведены примеры параметрического множества заданий и потока заданий.

Упражнение 1 – Компиляция программы для запуска в CCS 2003

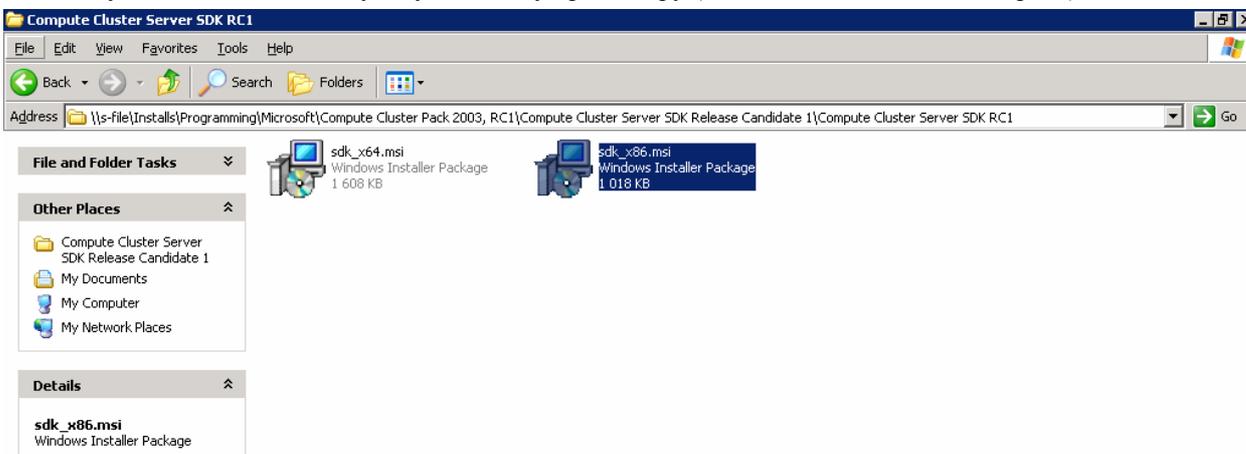
Как было отмечено в предыдущем пункте, Microsoft Compute Cluster Server 2003 позволяет управлять ходом выполнения как последовательных, так и параллельных задач. При этом параллельные MPI задачи не обязательно должны быть скомпилированы для MS MPI (хотя в случае MPI использование реализации от Microsoft является предпочтительным). Кроме того, возможно использование других технологий поддержки параллельного программирования (например, программирование с использованием OpenMP).

Данный пункт описывает только компиляцию параллельных программ для MS MPI в Microsoft Visual Studio 2005.

Задание 1 - Установка Microsoft Compute Cluster Pack SDK

Для компиляции параллельных программ, работающих в среде MS MPI, необходимо установить **SDK (Software Development Kit)** – набор интерфейсов и библиотек для вызова MPI-функций:

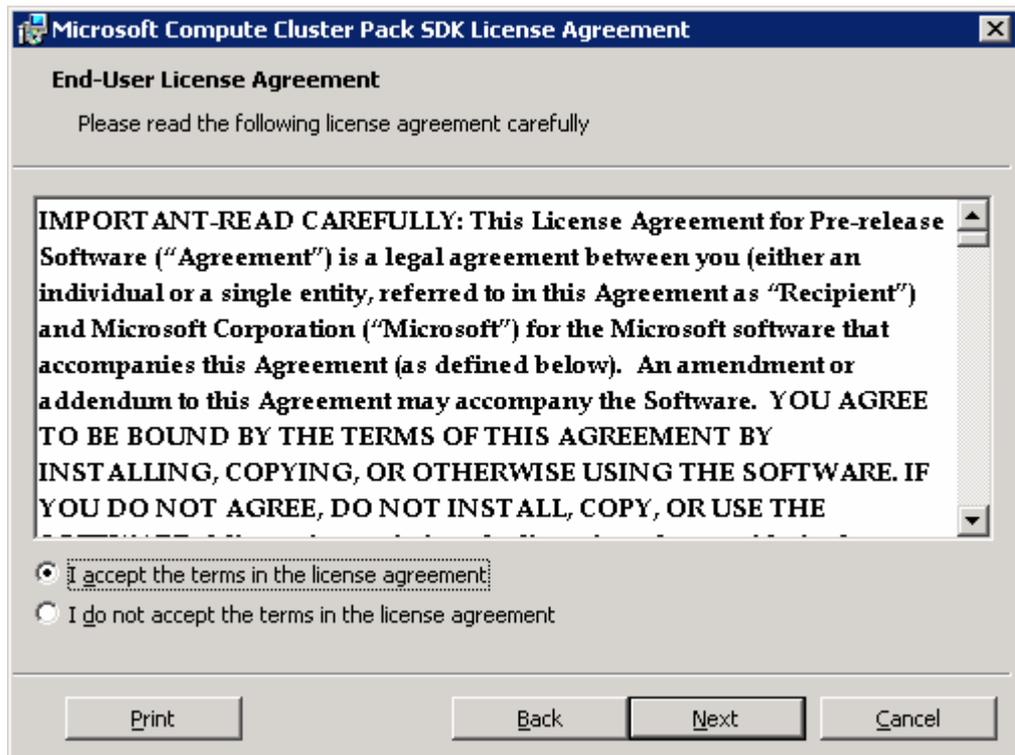
- Откройте директорию со скачанной версией SDK (описание процедуры скачивания можно найти в лабораторной работе “Установка Microsoft Compute Cluster Server 2003”) и запустите программу установки, соответствующую Вашему процессору (32-битная или 64-битная версия)



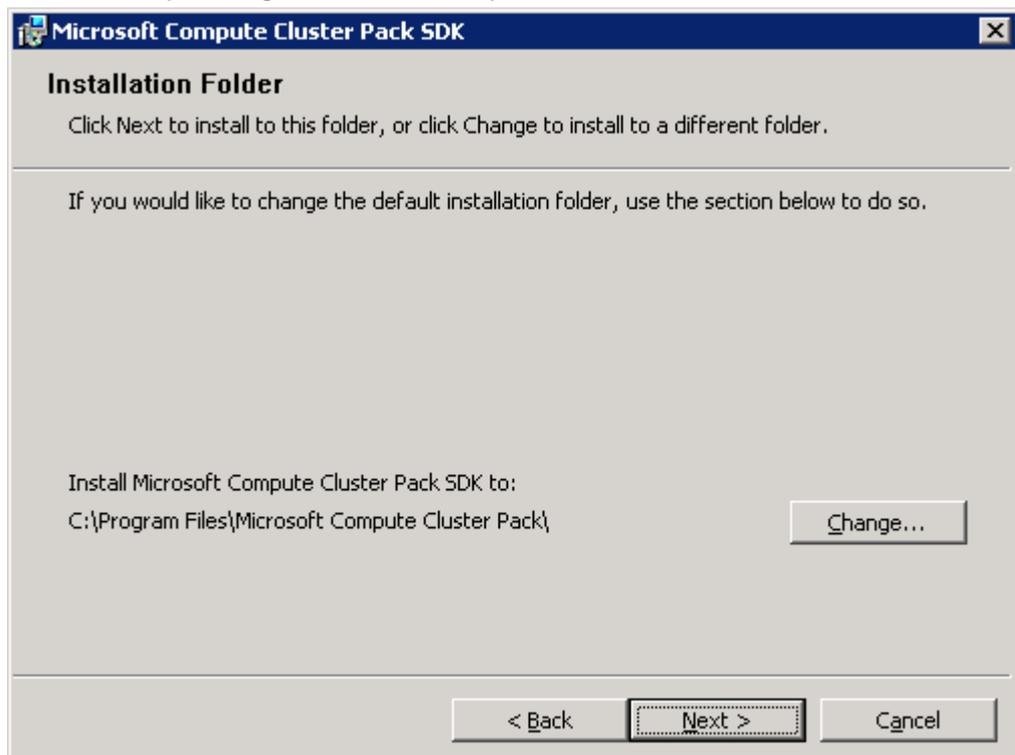
- В открывшемся окне нажмите кнопку “**Next**” для начала установки



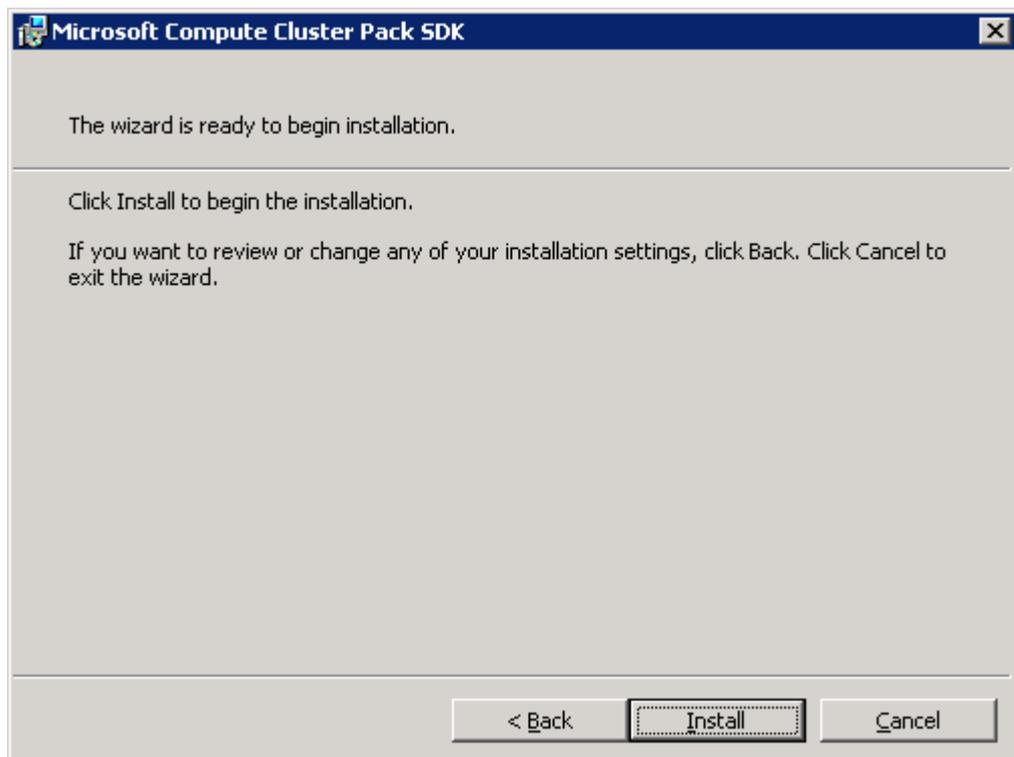
- Внимательно прочитайте лицензионное соглашение. Выберите пункт "**I accept the terms in the license agreement**" в случае согласия с лицензионным соглашением об использовании системы CCS 2003 и нажмите кнопку "**Next**"



- Выберите директорию, в которую будет установлен SDK. Для изменения стандартной директории нажмите кнопку "Change". Нажмите кнопку "Next"



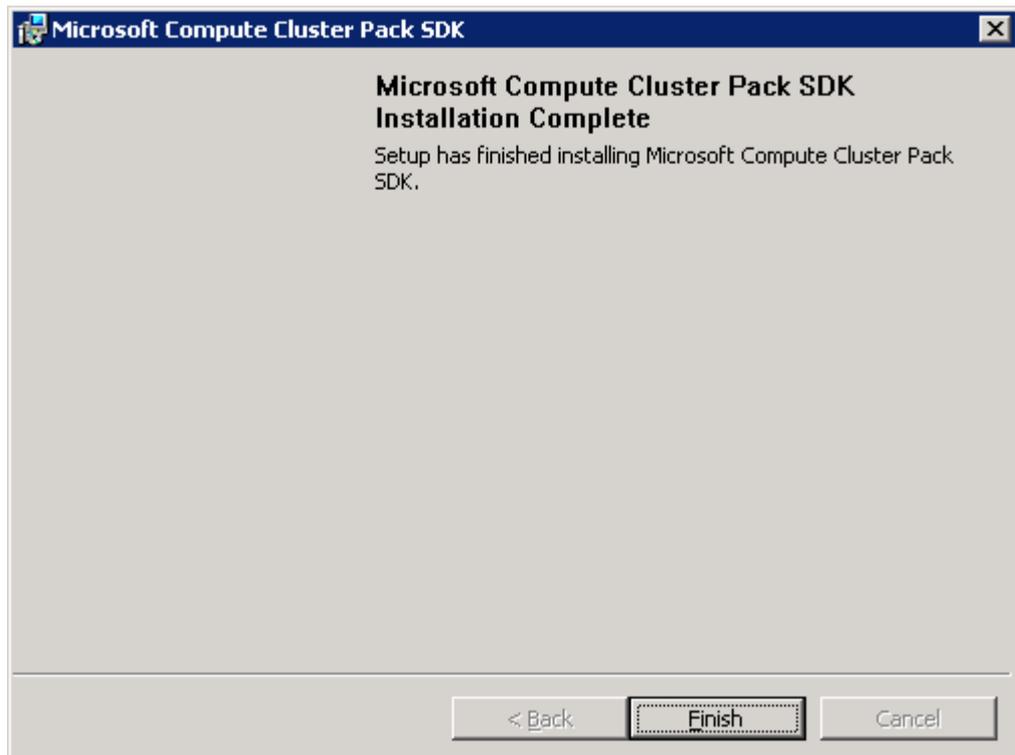
- В открывшемся окне нажмите кнопку "Install" для начала установки SDK



- Дождитесь пока программа установки SDK скопирует требуемые файлы



- По окончании копирования необходимых файлов нажмите кнопку "Finish"

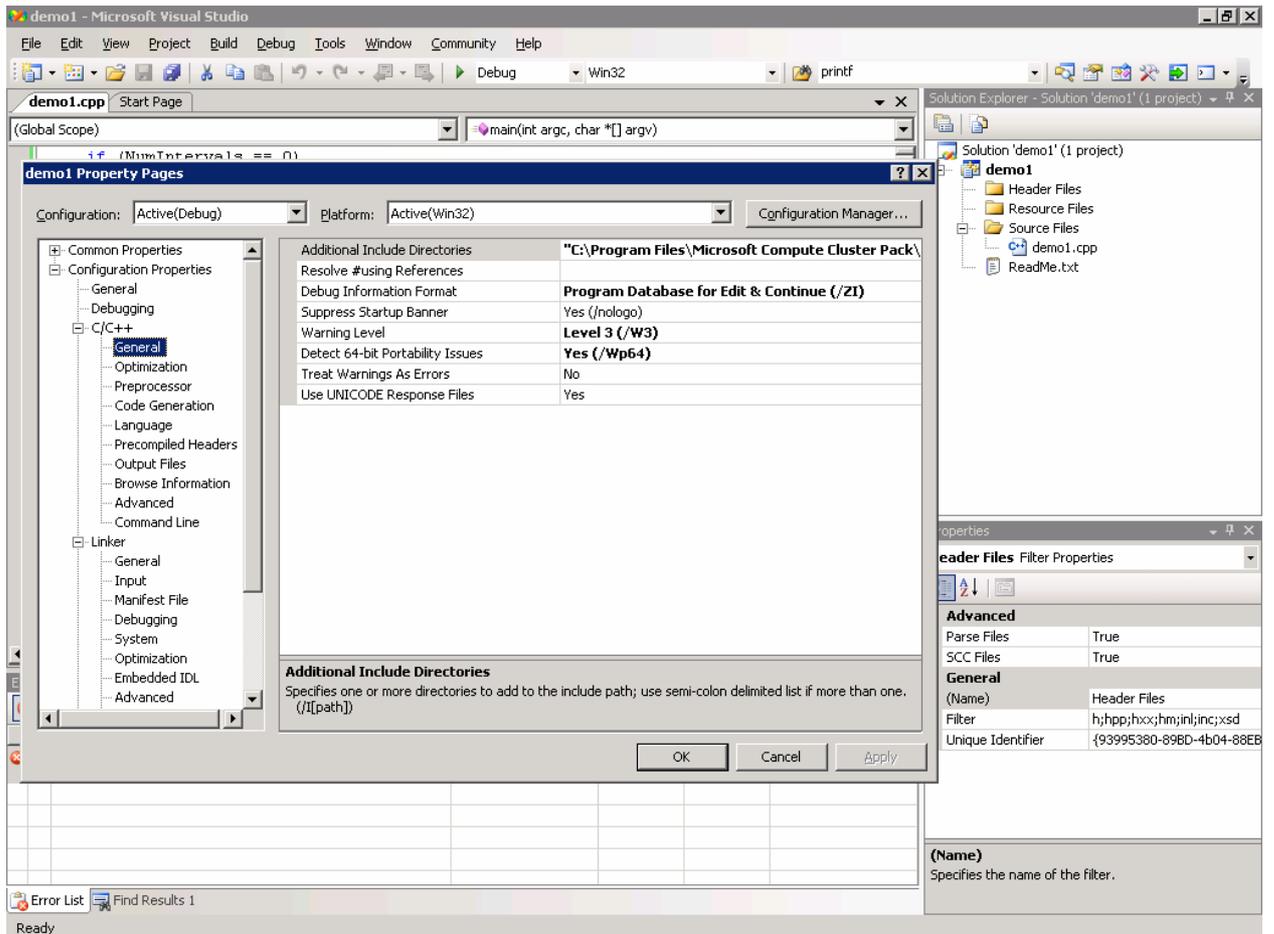


- Поздравляем! Установка Microsoft Compute Cluster Server 2003 SDK завершена.

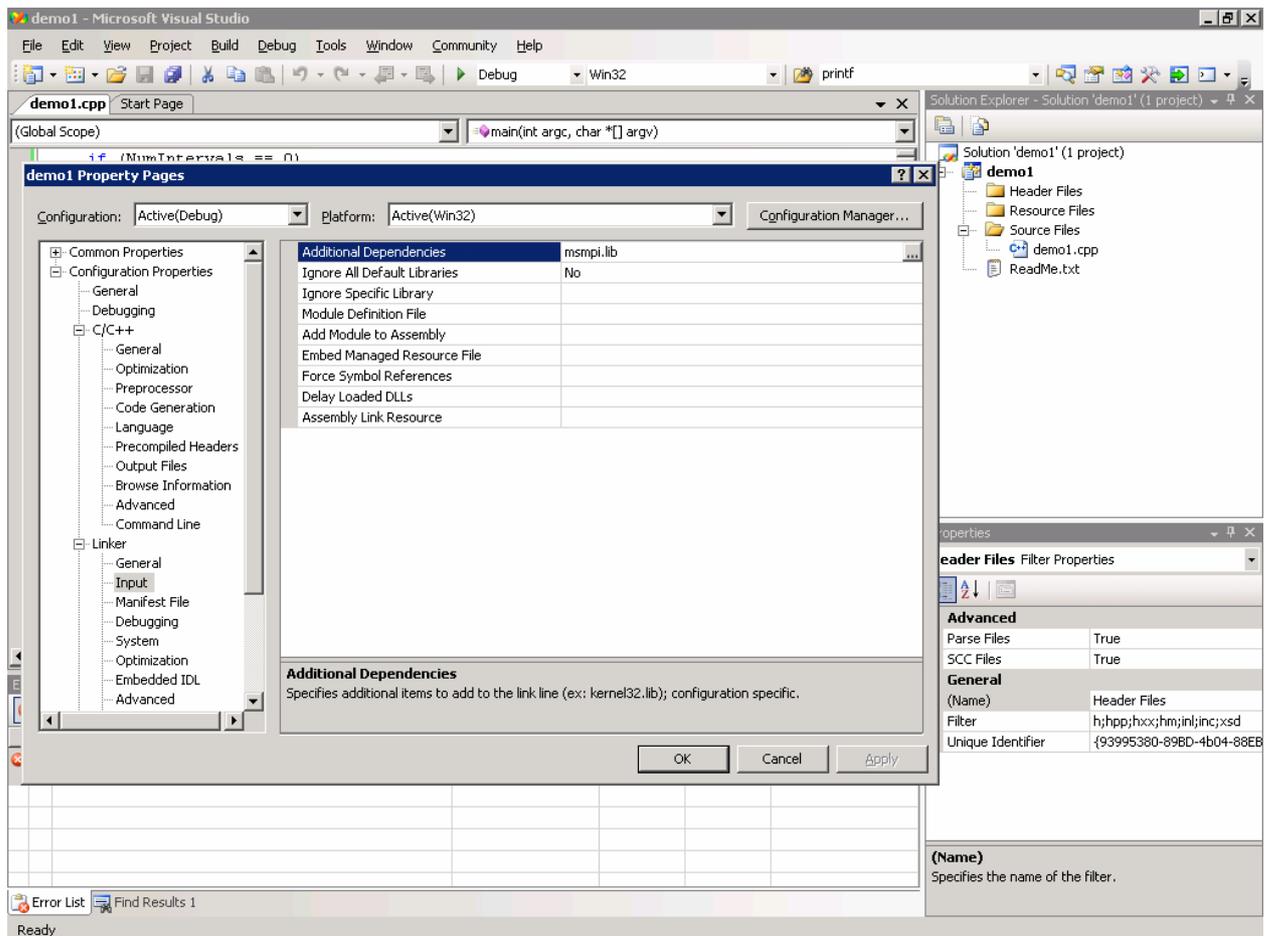
Задание 2 – Настройка интегрированной среды разработки Microsoft Visual Studio 2005

Для того, чтобы скомпилировать Вашу программу для использования в среде MS MPI, необходимо изменить следующие настройки проекта по умолчанию в Microsoft Visual Studio 2005:

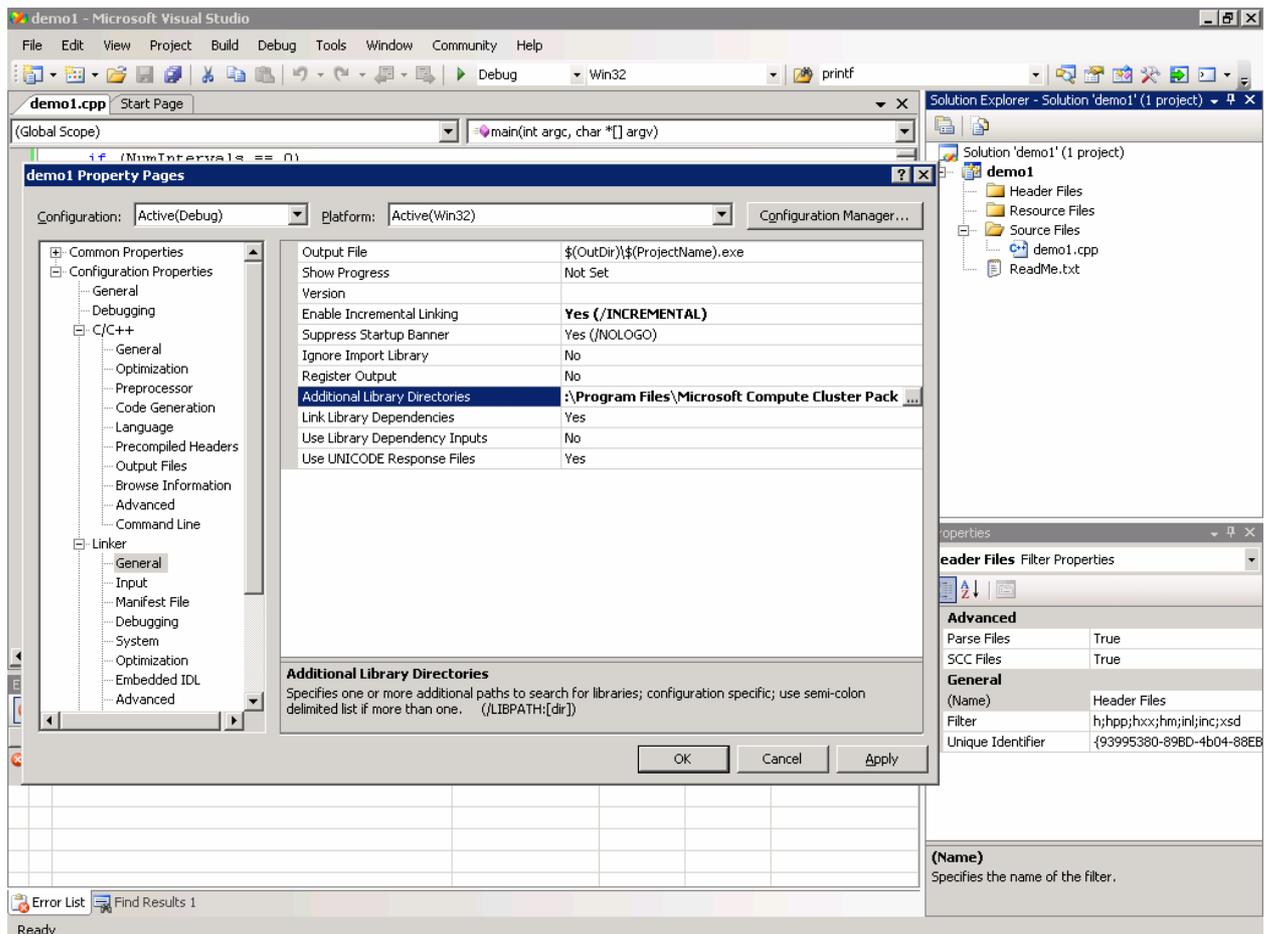
- **Путь до заголовочных файлов объявлений MPI.** Выберите пункт меню **Project->Project Properties**. В пункте **Configuration Properties->C++->General->Additional Include Directories** введите путь до заголовочных файлов MS MPI: **<Директория установки CCS SDK>\Include**



- Библиотечный файл с реализацией функций MPI. Выберите пункт меню **Project->Project Properties**. В пункте **Configuration Properties->C++->Linker->Input->Additional Dependencies** введите название библиотечного файла **msmpi.lib**



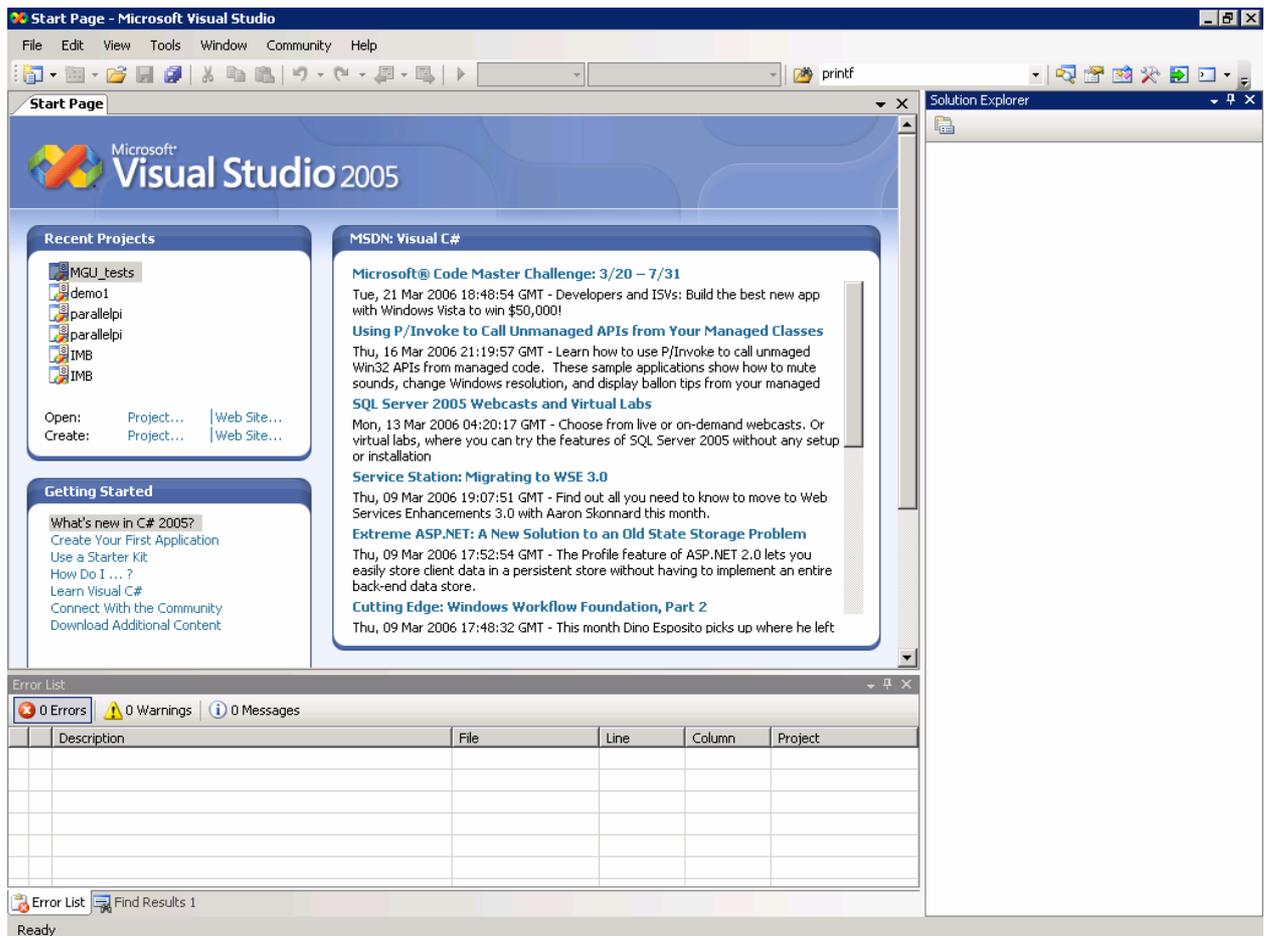
- **Путь до библиотечного файла msmpi.lib.** Выберите пункт меню **Project->Project Properties**. В пункте **Configuration Properties->C++->Linker->General->Additional Library Directories** введите путь до библиотечного файла **msmpi.lib**: **<Директория установки CCS SDK>\Lib\i386** или **<Директория установки CCS SDK>\Lib\AMD64** в зависимости от используемой Вами архитектуры процессоров.



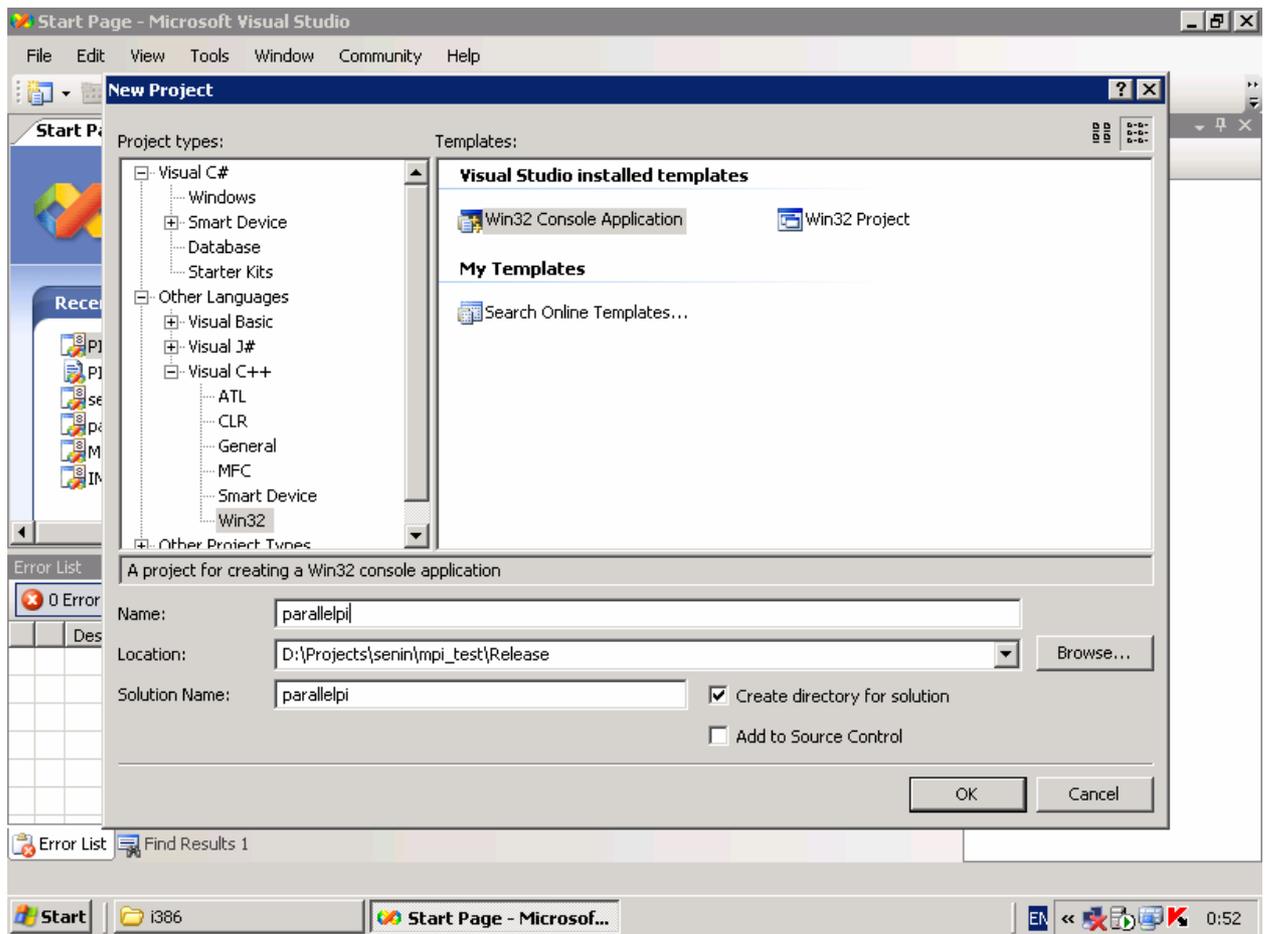
Задание 3 – Компиляция параллельной программы в Microsoft Visual Studio 2005

В качестве примера параллельной программы для этого задания будет использоваться параллельный алгоритм вычисления числа π . В данной работе рассматриваются только технические вопросы использования Microsoft Compute Cluster Server 2003; описание алгоритма и вопросы его реализации рассмотрены в лабораторной работе "Параллельный метод вычисления числа π ". В данном задании будут рассмотрены вопросы использования Visual Studio 2005 для компиляции параллельной MPI программы для использования в среде MS MPI:

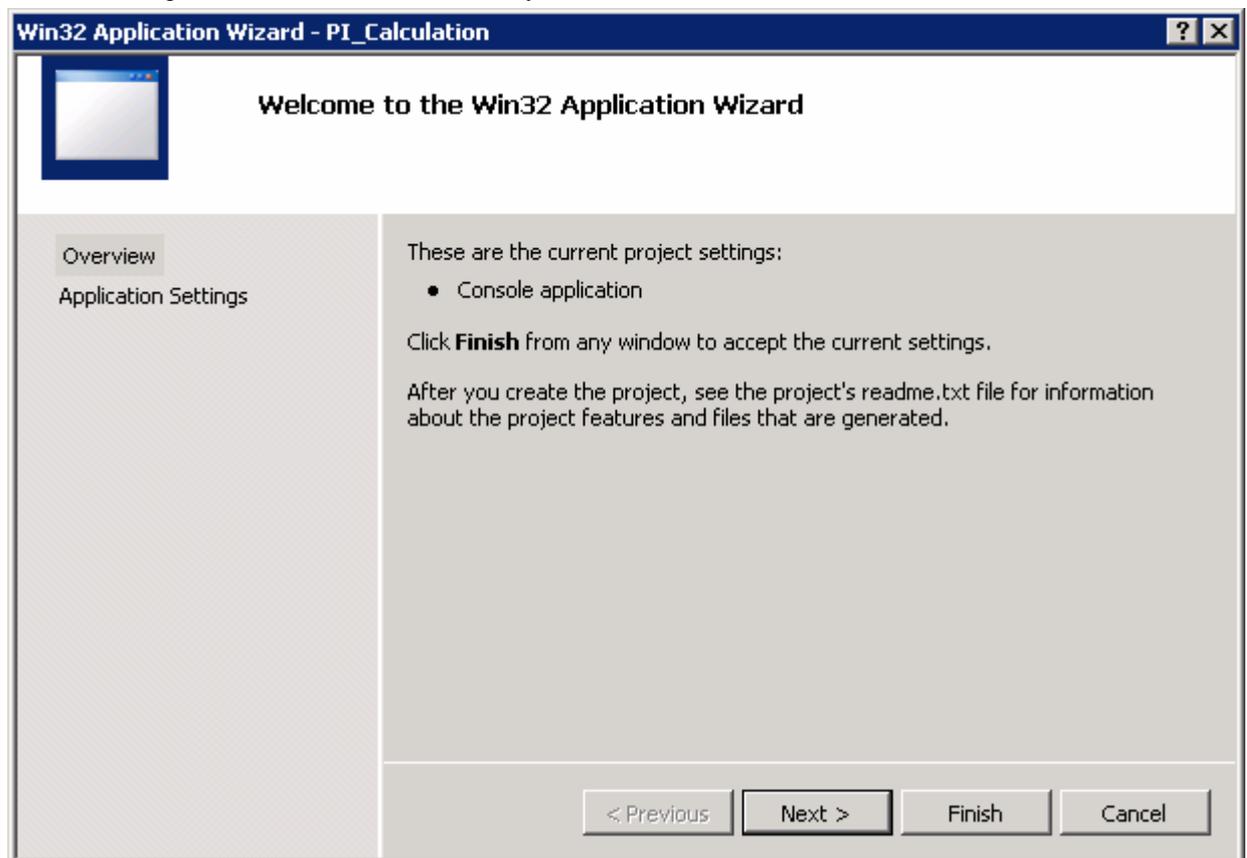
- Запустите **Microsoft Visual Studio 2005**



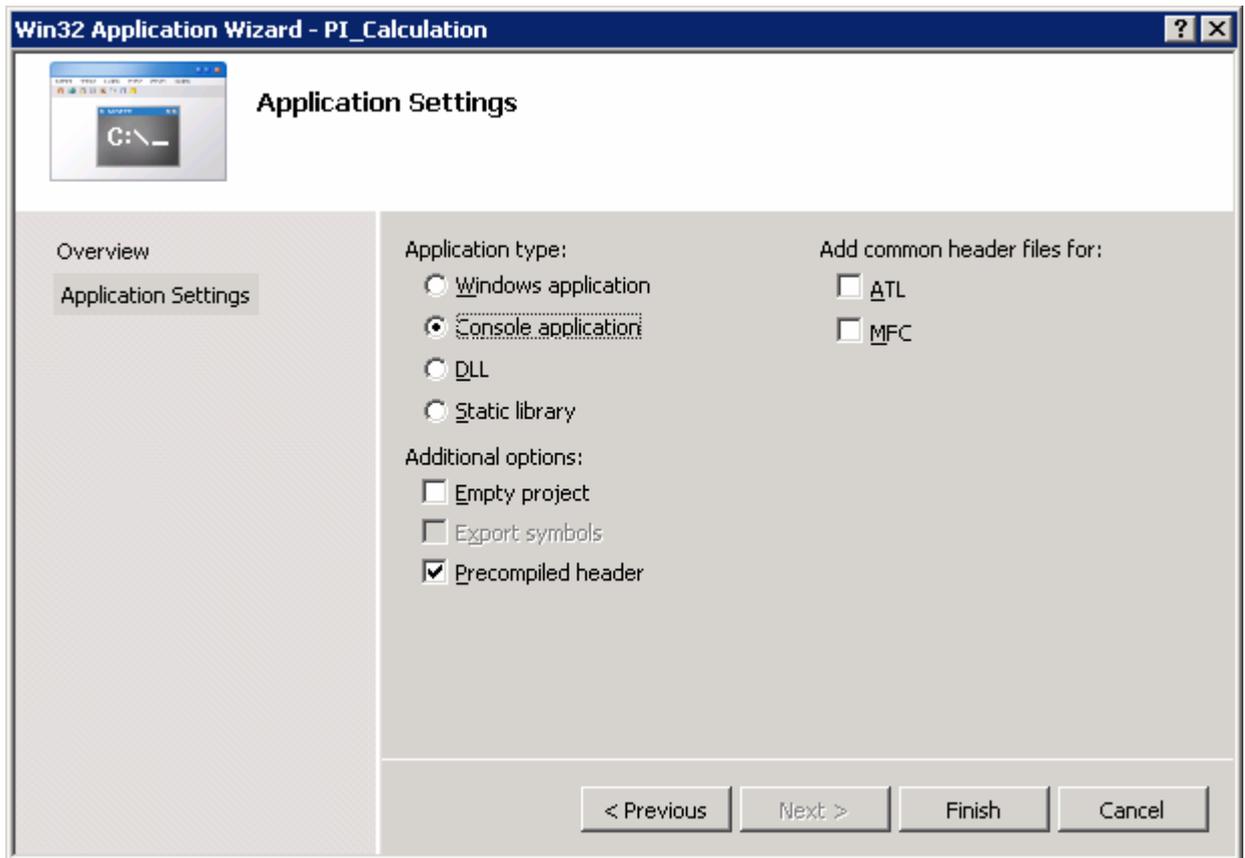
- Создайте новый проект: выберите пункт меню **File->New->Project**. В окне выбора нового проекта выберите **консольное Win32 приложение (Other Languages->Visual C++->Win32->Win32 Console Application)**, введите имя проекта в поле "Name" (например, "parallelp1") и убедитесь, что путь до проекта выбран правильно (поле "Location"). Нажмите кнопку "OK" для выбора остальных настроек создаваемого проекта



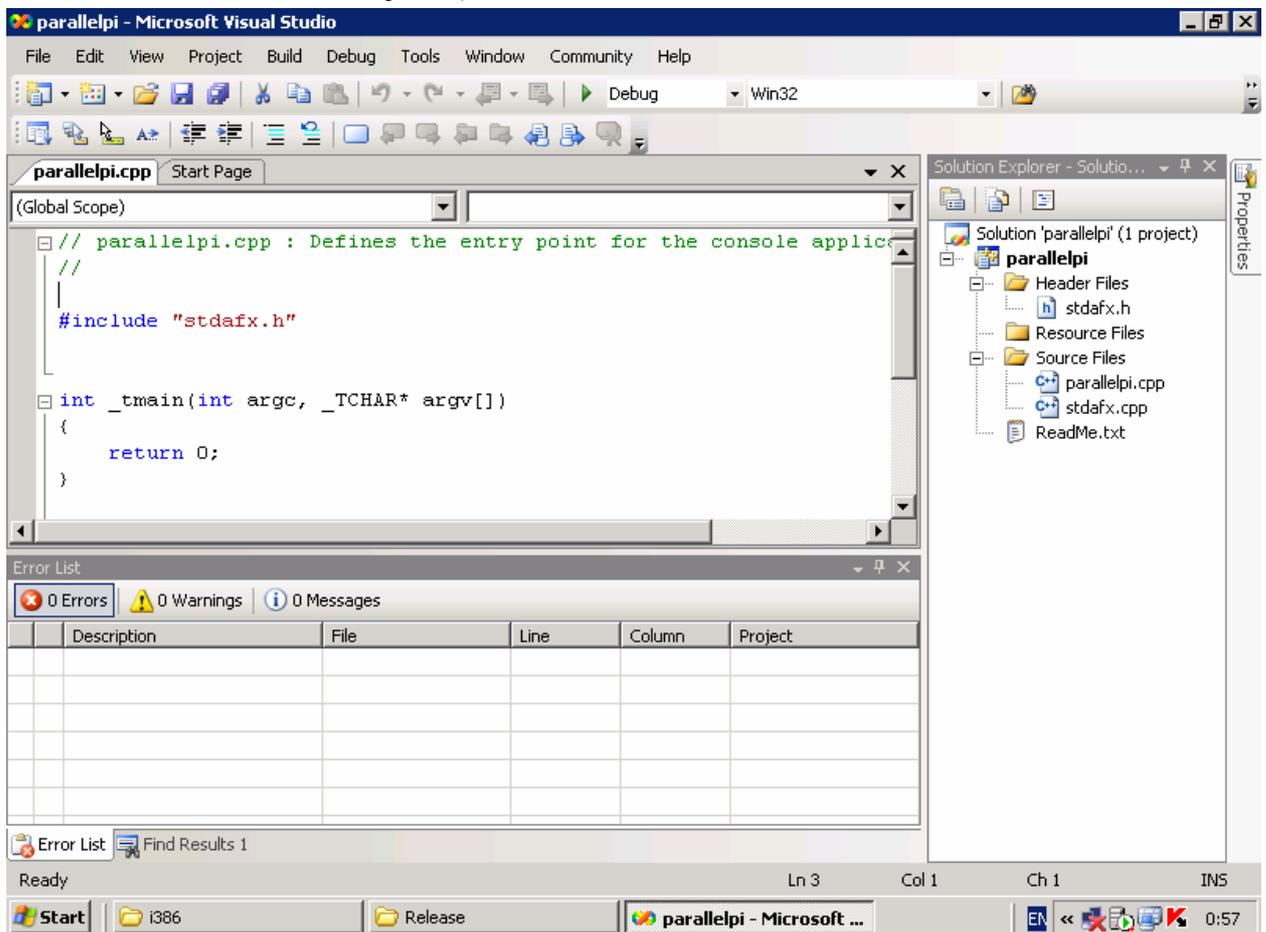
- В открывшемся окне нажмите кнопку “Next”



- В открывшемся окне выберите настройки проекта (можно оставить все настройки по умолчанию). Нажмите кнопку “Finish”



- В окне **Solution Explorer** щелкните 2 раза на файле **parallepi.cpp** (имя файла совпадает с введенным названием проекта)



- Удалите содержимое файла и замените его следующим (см. лабораторную работу "Параллельный метод вычисления числа Пи"):

```

#include "stdafx.h"
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <mpi.h>

void main(int argc, char *argv[]) {
    int    NumIntervals    = 0;    // num intervals in the domain [0,1]
    double IntervalWidth  = 0.0;  // width of intervals
    double IntervalLength = 0.0;  // length of intervals
    double IntrvlMidPoint = 0.0;  // x mid point of interval
    int    Interval       = 0;    // loop counter
    int    done           = 0;    // flag
    double MyPI           = 0.0;  // storage for PI approximation results
    double ReferencePI    = 3.141592653589793238462643; // value for comparison
    double PI;
    char   processor_name[MPI_MAX_PROCESSOR_NAME];
    char   (*all_proc_names)[MPI_MAX_PROCESSOR_NAME];
    int    numprocs;
    int    MyID;
    int    namelen;
    int    proc = 0;

    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD,&MyID);
    MPI_Get_processor_name(processor_name,&namelen);

    all_proc_names = (char(*)[128]) malloc(numprocs * MPI_MAX_PROCESSOR_NAME);

    MPI_Gather(processor_name, MPI_MAX_PROCESSOR_NAME, MPI_CHAR,
        all_proc_names, MPI_MAX_PROCESSOR_NAME, MPI_CHAR, 0, MPI_COMM_WORLD);
    if (MyID == 0) {
        for (proc=0; proc < numprocs; ++proc)
            printf("Process %d on %s\n", proc, all_proc_names[proc]);
    }

    IntervalLength = 0.0;
    if (MyID == 0) {
        if (argc > 1) {
            NumIntervals = atoi(argv[1]);
        }
        else {
            NumIntervals = 100000;
        }
        printf("NumIntervals = %i\n", NumIntervals);
    }

    // send number of intervals to all procs
    MPI_Bcast(&NumIntervals, 1, MPI_INT, 0, MPI_COMM_WORLD);

    if (NumIntervals != 0)
    {
        //approximate the value of PI
        IntervalWidth  = 1.0 / (double) NumIntervals;

        for (Interval = MyID+1; Interval <= NumIntervals; Interval += numprocs){
            IntrvlMidPoint = IntervalWidth * ((double)Interval - 0.5);
            IntervalLength += (4.0 / (1.0 + IntrvlMidPoint*IntrvlMidPoint));
        }
        MyPI = IntervalWidth * IntervalLength;
    }
}

```

```

// Calculating the sum of all local alues of MyPI
MPI_Reduce(&MyPI, &PI, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);

//report approximation
if (MyID == 0) {
    printf("PI is approximately %.16f, Error is %.16f\n",
        PI, fabs(PI - ReferencePI));
}
}

MPI_Finalize();
}

```

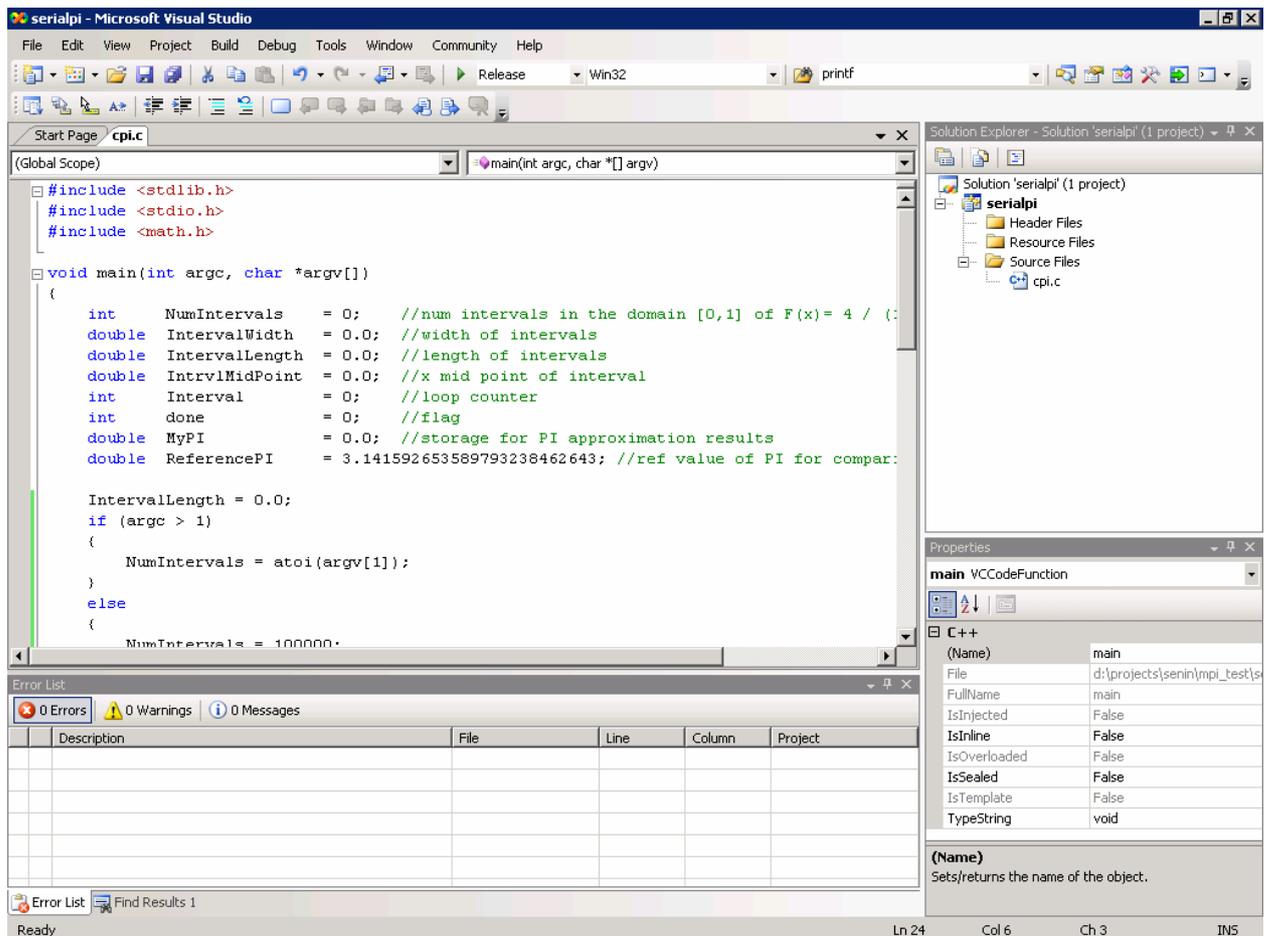
- Выполните настройки проекта Visual Studio 2005 для компиляции MPI части проекта в соответствии с указаниями пункта “**Настройка интегрированной среды разработки Microsoft Visual Studio 2005**”,
- Выполните команду **Build->Rebuild Solution** для компиляции и линковки проекта,
- Поздравляем! Компиляция параллельной программы для MS MPI успешно завершена.

Упражнение 2 – Запуск последовательной задачи

Последовательной называется задача, которая используется для своей работы ресурсы только одного процессора. Порядок компиляции последовательной программы (а также параллельной программы с использованием технологии OpenMP) для использования на кластере под управлением Compute Cluster Server 2003 не отличается от обычного и не требует использования дополнительных библиотек. В данном задании рассматривается порядок запуска последовательной задачи на кластере.

Запуск программы через графический пользовательский интерфейс

- Откройте проект последовательного алгоритма вычисления числа Пи (**serialpi**), поставляющийся вместе с лабораторной работой "Параллельный метод вычисления числа Пи", и скомпилируйте программу в конфигурации **Release**



- Откройте **Computer Cluster Job Manager (Start->All Programs->Microsoft Compute Cluster Pack->Compute Cluster Job Manager)** для запуска программы на кластере. Если Вы установили клиентскую часть **Compute Cluster Pack** на Вашу рабочую станцию, то постановку задач в очередь можно выполнять непосредственно с Вашего компьютера, иначе необходимо зайти по **Remote Desktop Connection** на головной узел кластера или любой другой узел с установленной клиентской частью

Job Queue at s-cw-head

ID	Name	Priority	Submitted By	Status	Submit Time	Pending Reason
1	hostname	Normal	CCAM\Senin	Finished	21.06.2006 4:10:14	
2	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:19:50	
3	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:24:24	
4	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:25:23	
5	imb	Normal	CCAM\Senin	Finished	21.06.2006 5:13:20	

Select a job to view its tasks

Name	Status	Task Id	Command Line	Processors	End Time	Failure Message

Ready

- В открывшемся окне менеджера заданий выберите пункт меню **File->Submit Job** для постановки нового задания в очередь
- В окне постановки задания в очередь введите имя задания (поле “**Job Name**”), при необходимости измените приоритет задания (пользовательские задания с большим приоритетом будут выполнены раньше заданий с меньшим приоритетом). Перейдите на вкладку “**Processors**”

Submit Job Serial Pi computing

General | Processors | Tasks | Licenses | Advanced

Serial Pi computing

Job Name: Serial Pi computing

Project Name:

Priority: Normal

Submitted By: N/A

Submitted on: N/A

Status: Not Submitted

Save As Template... Submit Cancel

- На вкладке **Processors** введите **минимальное** и **максимальное** числа процессоров, необходимых для выполнения задания (в нашем случае максимальное необходимое число процессоров – один, так как задача последовательная). Под максимальным здесь понимается оптимальное для задания число процессоров (именно столько будет выделено в случае низкой загрузки кластера). Гарантируется, что задание не начнет выполняться, если на кластере доступно менее минимального числа процессоров. Дополнительно можно ввести предполагаемое время работы задания (это поможет планировщику эффективнее распределить системные ресурсы) – панель **Estimate run time for this job**. Если Вы хотите, чтобы для задания вычислительные ресурсы оставались зарезервированными в течение указанного времени, даже после того, как все задачи задания завершили свою работу, то поставьте галочку около пункта **“Run job until end of run time or until canceled”**. Таким образом, Вы сможете запускать новые задачи в рамках задания даже после того, как все задачи, указанные первоначально, выполнены. Перейдите на вкладку **“Tasks”** для добавления в задание новых задач

Submit Job Serial Pi computing

General Processors Tasks Licenses Advanced

Processors required for this job

Processors available in this cluster: 60

Minimum required: 1

Maximum required: 1

Estimate run time for this job

Days: 0 Hours: 0 Minutes: 1

Run job until end of run time or until canceled.

This option lets you run extra tasks after running all tasks already listed in the job if there is time left.

Save As Template... Submit Cancel

- Введите имя задачи (поле “**Task Name**”) и команду, которую необходимо выполнить (поле “**Command Line**”) – имя программы и параметры командной строки. Программу необходимо разместить на сетевом диске, доступном со всех узлов кластера. Нажмите кнопку “**Add**” для добавления задачи в задание

Submit Job Serial Pi Calculation [X]

General | Processors | **Tasks** | Licenses | Advanced

Task Command Line

Task Name:

Command Line:

Use job's allocated processors

Minimum Processors: Maximum Processors:

This job contains the following tasks:

Order	Command Line	Processors

[What is it?](#)

Task Summary

- Добавленная задача появится в списке задач текущего задания (список “**This job contains the following tasks**”). Выделите ее в списке и нажмите кнопку “**Edit**” для редактирования дополнительных параметров задачи

Submit Job Serial Pi Calculation

General | Processors | **Tasks** | Licenses | Advanced

Task Command Line

Task Name:

Command Line:

Use job's allocated processors

Minimum Processors: Maximum Processors:

This job contains the following tasks:

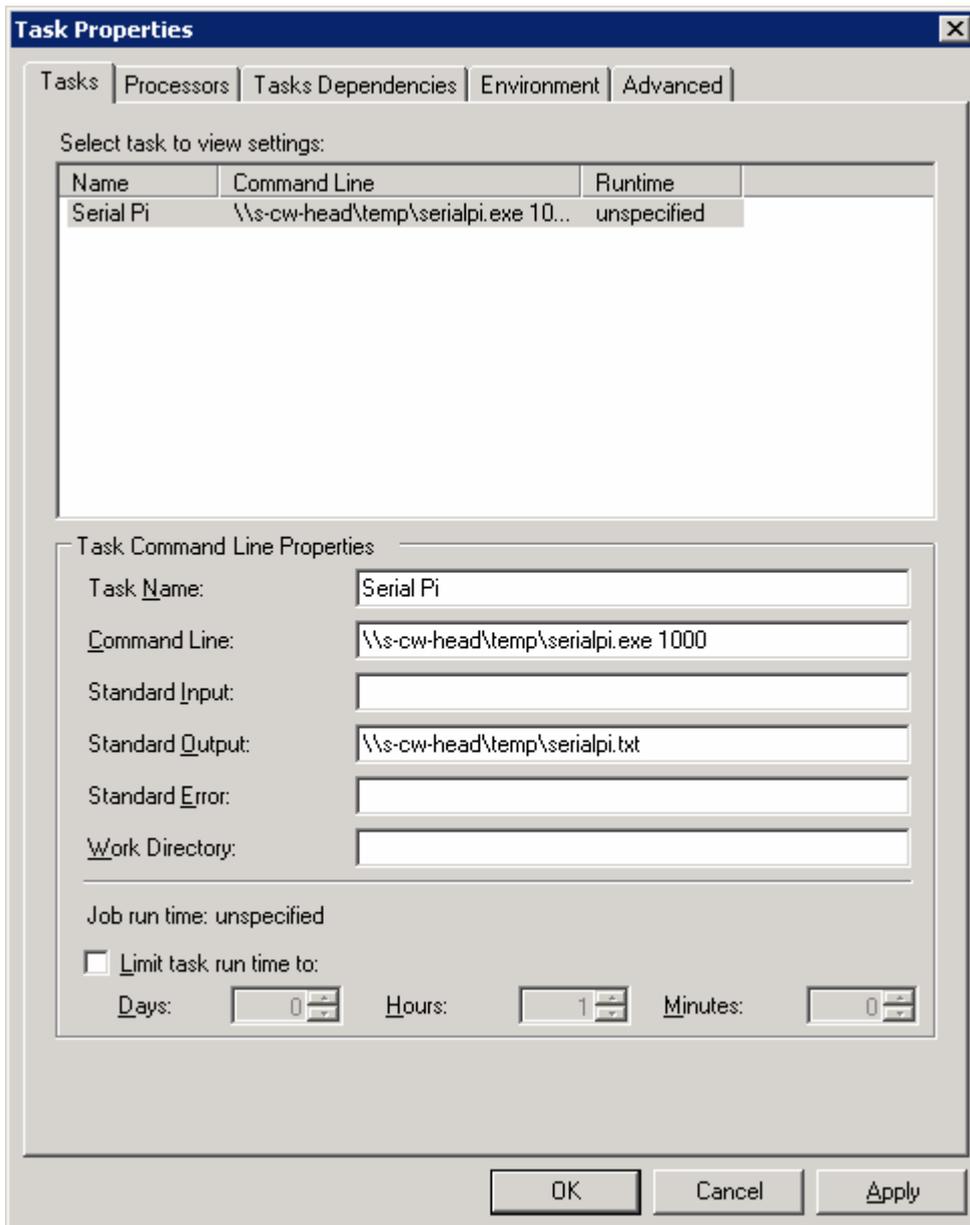
Order	Command Line	Processors
1	\\s-cw-head\temp\serialpi.exe 1000	1

[What is it?](#)

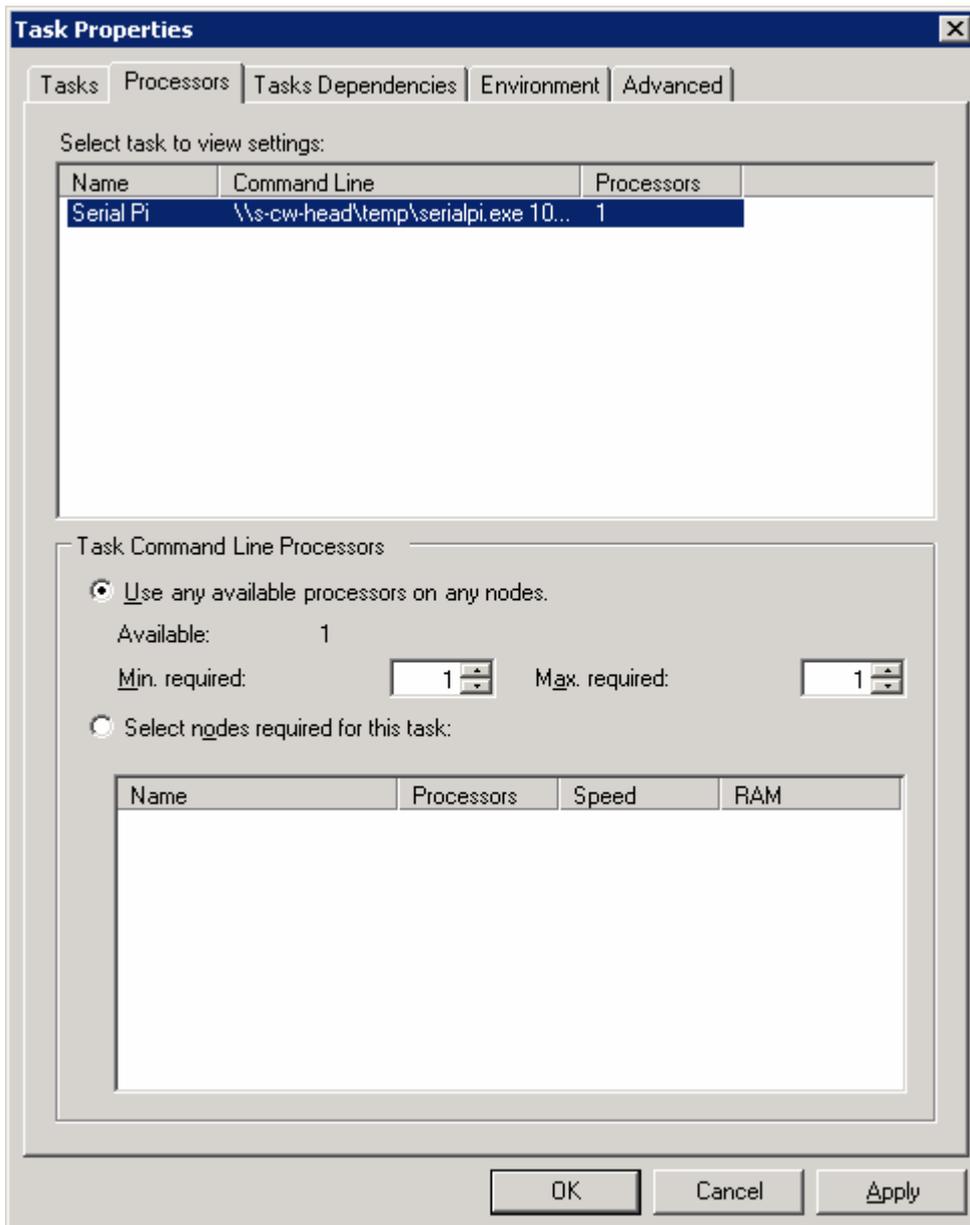
Task Summary

Name :Serial Pi
 Command Line :\\s-cw-head\temp\serialpi.exe 1000
 Standard Input :
 Standard Output :
 Standard Error :
 Work Directory :
 Number of processors requested :1
 RunTime :Infinite
 Preceding tasks(dependent tasks) :
 Exclusive :False

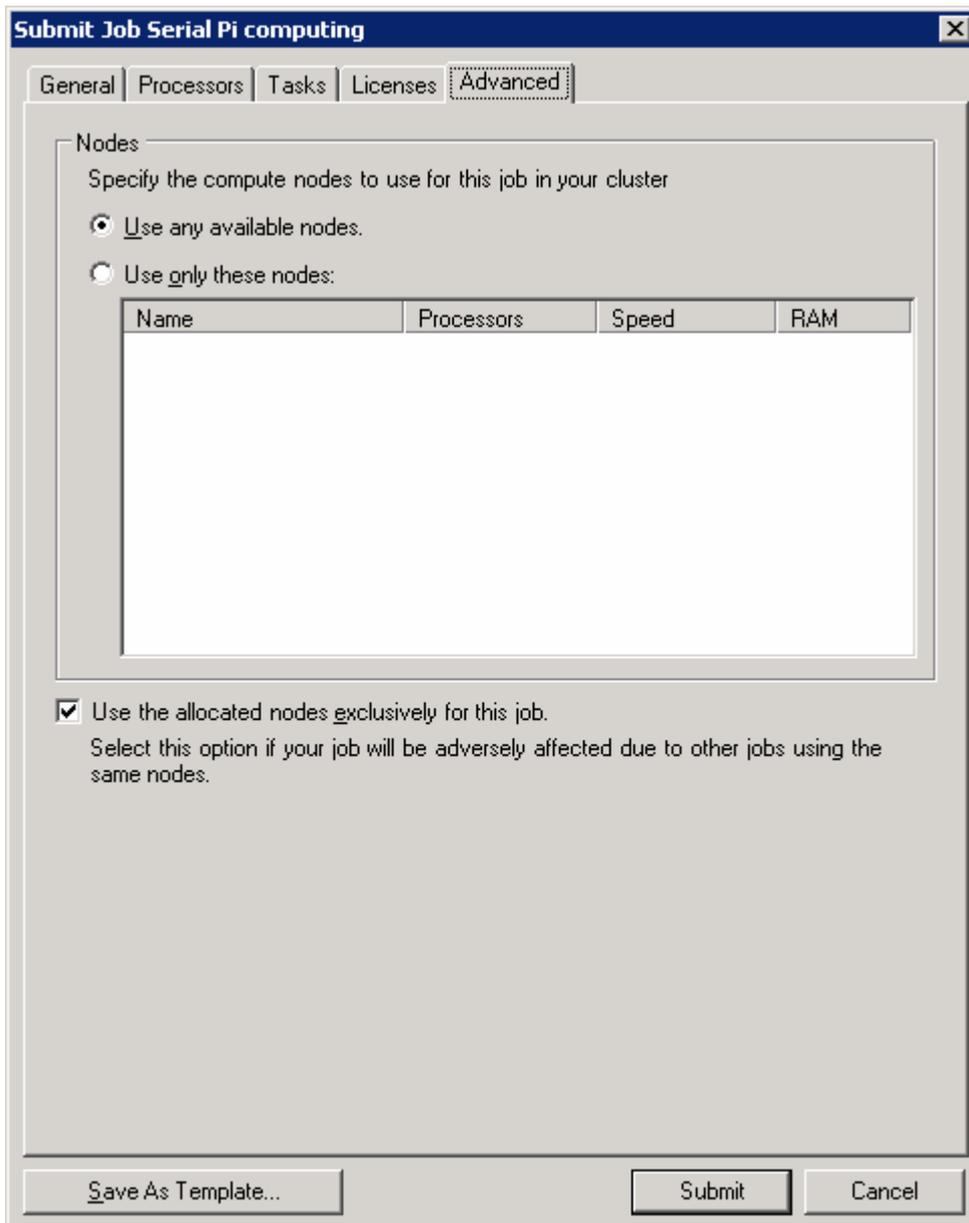
- В открывшемся окне введите файл, в который будет перенаправлен стандартный поток вывода консольного приложения (поле **“Standard Output”**). Кроме того, можно указать файл стандартного потока ввода (поле **“Standard Input”**), файл стандартного потока ошибок (поле **“Standard Error”**), рабочую директорию запускаемой программы (поле **“Work Directory”**) и ограничение по времени на продолжительность выполнения задачи (суммарное время выполнения задач не должно превышать оценки времени выполнения задания) – поле **“Limit task run time to”**. Выберите вкладку **“Processors”**,



- На вкладке “**Processors**” в верхнем списке (“**Select task to view settings**”) выделите задачу, настройки которой Вы хотите изменить, и укажите минимальное и максимальное числа процессоров для выбранной задачи (поля “**Min. required**” и “**Max. required**”) в случае, если Вы хотите, чтобы планировщик выбрал узлы для запуска автоматически (“**Use any available processors on any nodes**”). Если Вы хотите вручную указать узлы для запуска, выберите пункт “**Select nodes required for this task**” и поставьте флажки около требуемых узлов в нижнем списке. Так как задача последовательная, то для ее выполнения требуется только 1 процессор. На этом настройка параметров задачи закончена. Нажмите “**OK**” для сохранения внесенных изменений и возвращения к настройкам задания



- Перейдите на вкладку “**Advanced**” и выберите пункт “**Use any available nodes**” для автоматического выбора узлов для задания. Если Вы хотите вручную указать узлы, на которых будет выполняться задание, выберите пункт “**Use only these nodes**”. Учтите, что в том случае, если Вы вручную указали узлы для запуска задачи, то они также должны быть выбраны для всего задания (в случае неавтоматического распределения). Установите флаг “**Use the allocated nodes exclusively for this job**” для того, чтобы запретить запуск нескольких заданий на одном узле. Нажмите кнопку “**Submit**” для добавления задания в очередь



- Введите имя и пароль пользователя, имеющего право запуска задач на кластере, и нажмите “OK”



- Задание появится в очереди. По окончании выполнения его состояние изменится на “**Finished**”

The screenshot shows a window titled "Job Queue at s-cw-head" with a menu bar (File, View, Job, Help) and a "Show:" dropdown set to "All Jobs". Below is a table of jobs:

ID	Name	Priority	Submitted By	Status	Submit Time	Pending Reason
1	hostname	Normal	CCAM\Senin	Finished	21.06.2006 4:10:14	
2	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:19:50	
3	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:24:24	
4	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:25:23	
5	imb	Normal	CCAM\Senin	Failed	21.06.2006 5:13:20	
6	Serial Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 1:59:22	
7	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 2:00:46	
8	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 3:50:22	
9	Serial Pi Calculation	Normal	CCAM\Senin	Finished	25.06.2006 4:08:39	

Below the job list is a section titled "Tasks for Serial Pi Calculation" with a sub-table:

Name	Status	Task Id	Command Line	Processors	End Time	Failure Message
Serial Pi	Finished	1	\\s-cw-head\temp\serialpi...	1	25.06.2006 ...	

The status bar at the bottom of the window reads "Ready".

- В файле, указанном в настройках задачи для перенаправления стандартного потока вывода, содержится результат работы программы

The screenshot shows a Notepad window titled "serialpi.txt - Notepad" with a menu bar (File, Edit, Format, View, Help). The text content is:

```
NumIntervals = 1000
PI is approximately 3.1415927369231227, Error is 0.0000000833333296
```

Запуск программы с использованием шаблона

Если в будущем Вы захотите еще раз запустить задачу последовательного вычисления числа Пи (например, с другими параметрами), то Вам будет полезна функция сохранения всех параметров задания, запущенного ранее, в xml файл с возможностью последующего быстрого создания копии:

- Откройте **Computer Cluster Job Manager (Start->All Programs->Microsoft Compute Cluster Pack->Compute Cluster Job Manager)** и дважды щелкните левой кнопкой мыши по заданию, параметры которого Вы хотите сохранить в xml-файл

Job Queue at s-cw-head

File View Job Help Show: All Jobs

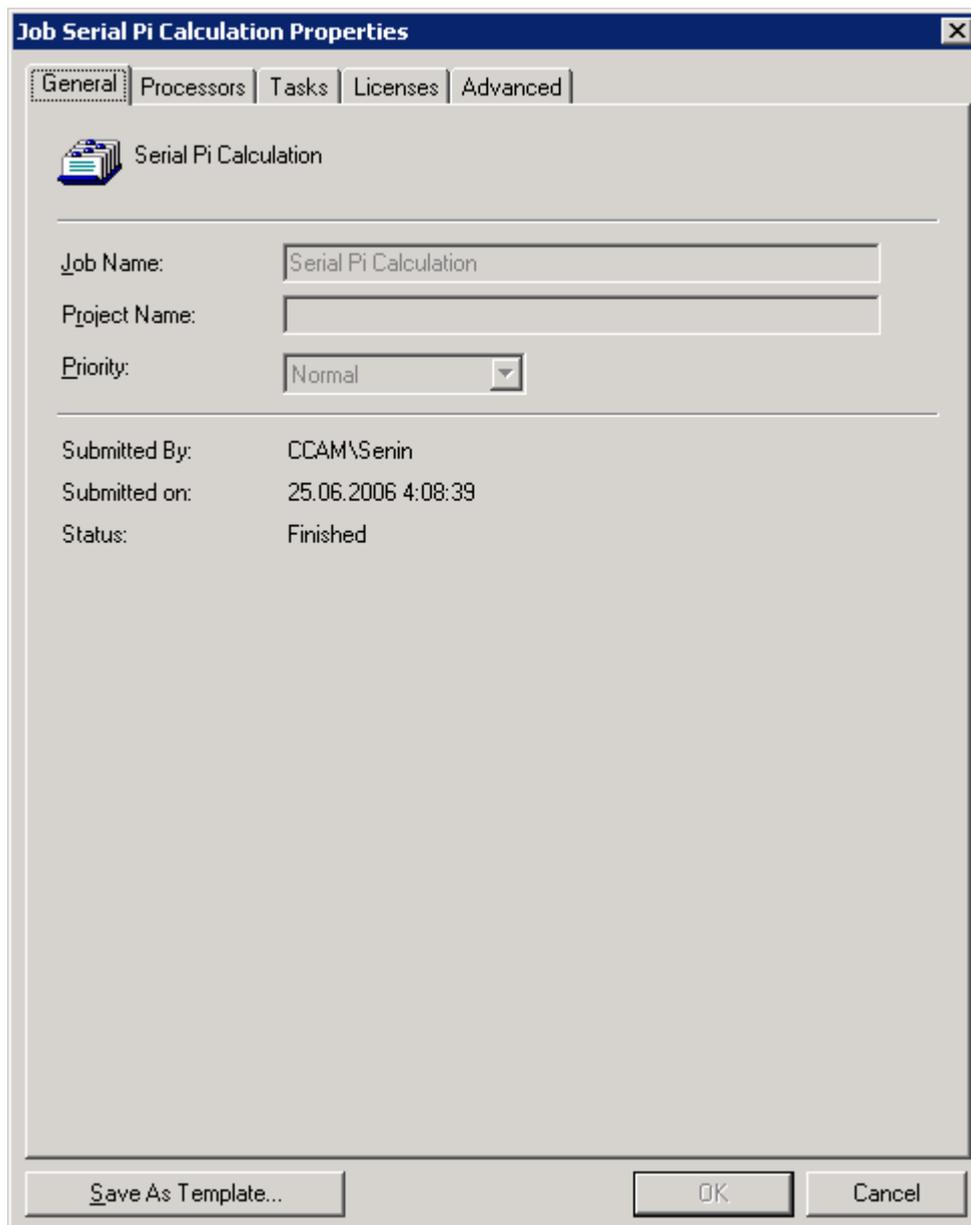
ID	Name	Priority	Submitted By	Status	Submit Time	Pending Reason
1	hostname	Normal	CCAM\Senin	Finished	21.06.2006 4:10:14	
2	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:19:50	
3	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:24:24	
4	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:25:23	
5	imb	Normal	CCAM\Senin	Failed	21.06.2006 5:13:20	
6	Serial Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 1:59:22	
7	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 2:00:46	
8	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 3:50:22	
9	Serial Pi Calculation	Normal	CCAM\Senin	Finished	25.06.2006 4:08:39	

Tasks for SerialPi Calculation

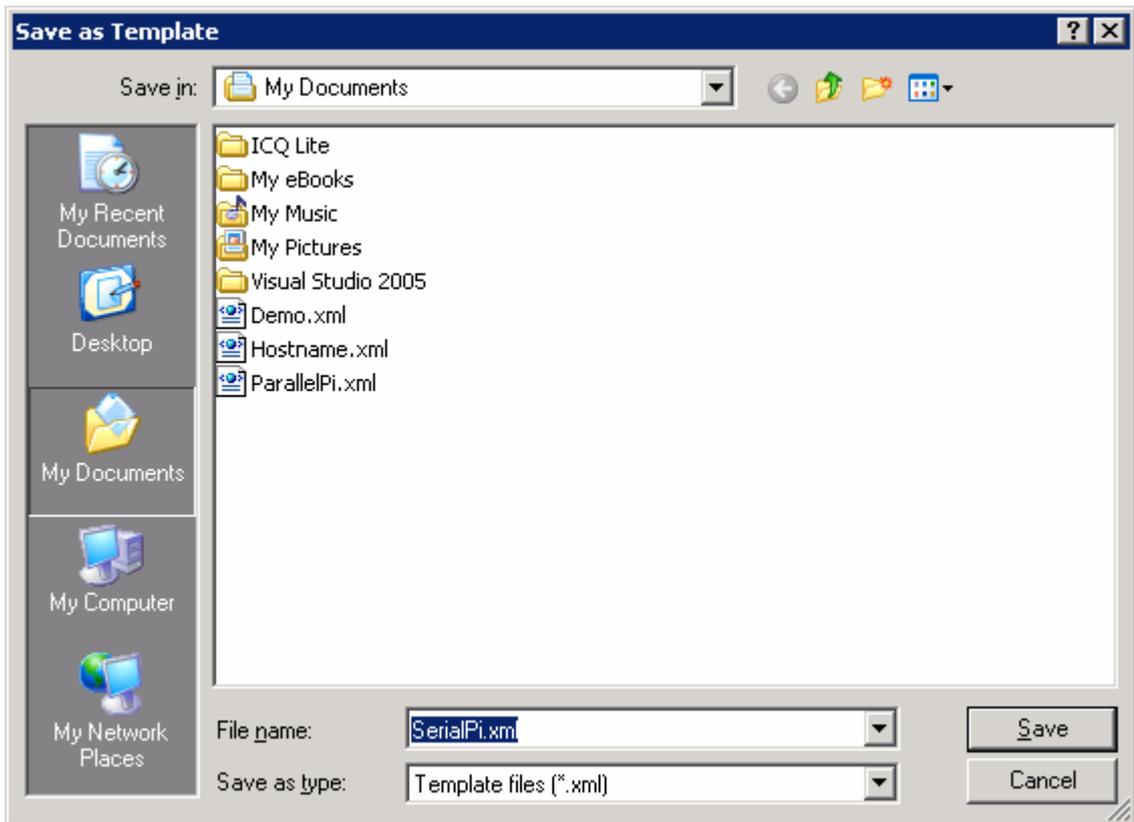
Name	Status	Task Id	Command Line	Processors	End Time	Failure Message
Serial Pi	Finished	1	\\s-cw-head\temp\serialpi...	1	25.06.2006 ...	

Ready

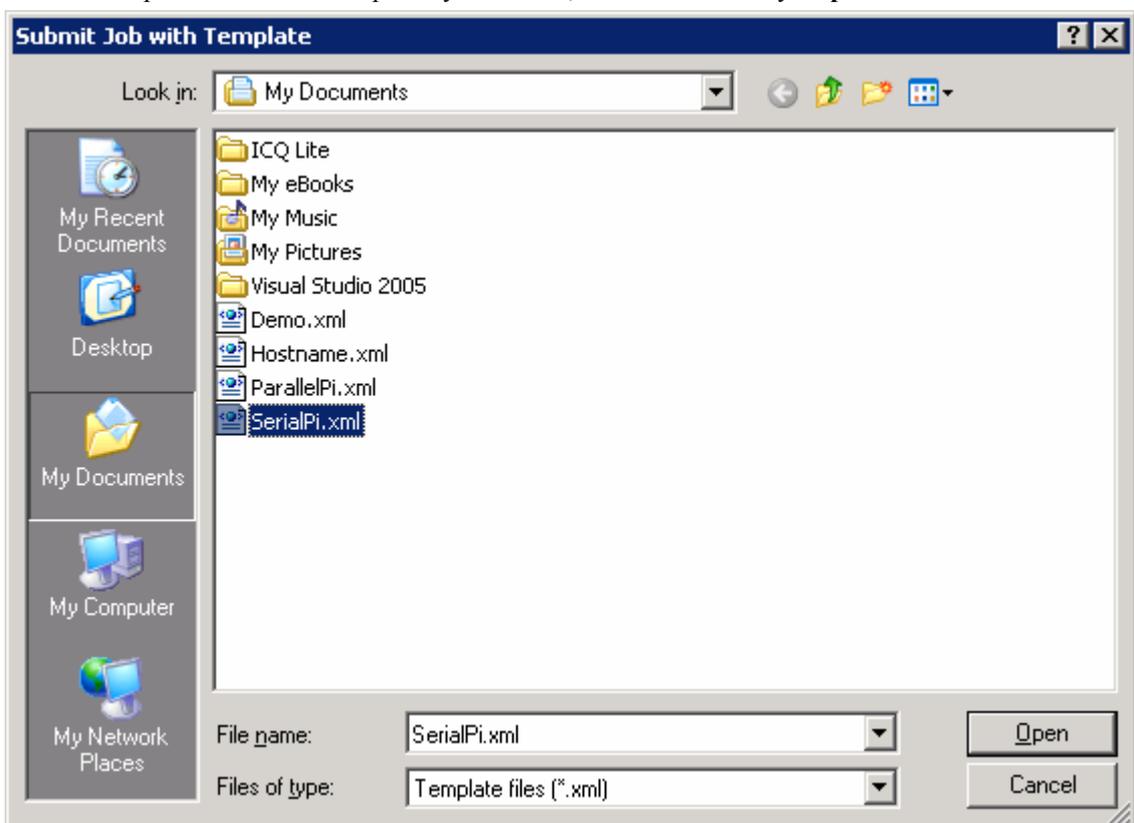
- В открывшемся окне нажмите кнопку “**Save As Template**” для сохранения параметров задания в файл



- В открывшемся окне выберите директорию, в которую следует сохранить файл и введите его имя. Нажмите кнопку “**Save**” для сохранения задания в файл

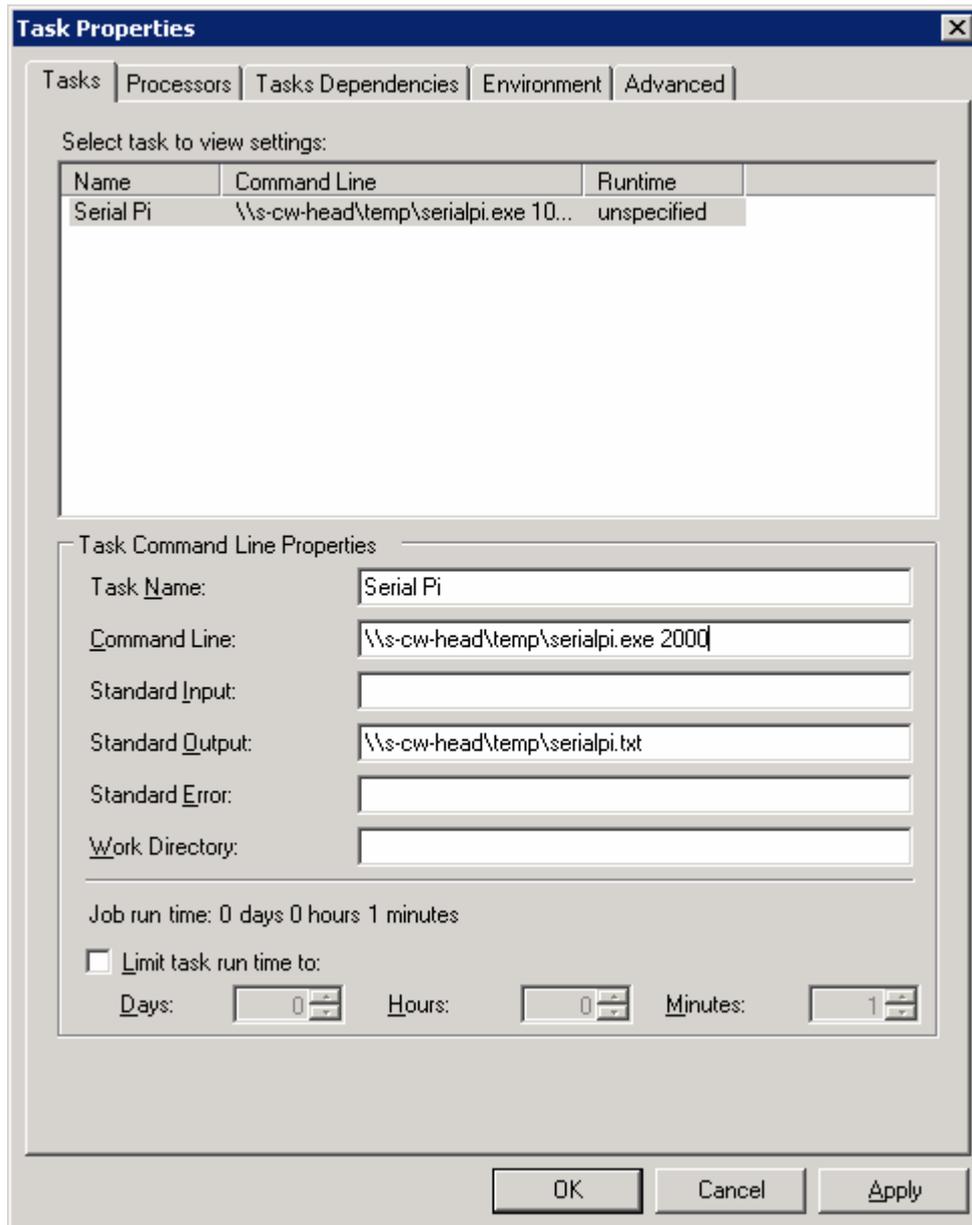


- Для создания задания с использованием шаблона в окне **Compute Cluster Job Manager** выберите пункт меню **File->Submit Job with Template...** В окне выбора шаблона выделите файл, в который было сохранено задание на предыдущем шаге, и нажмите кнопку **“Open”**



- Откроется окно добавления задания в очередь. При этом параметры задания и всех входящих в него задач будут теми же, что и у задания, на основе которого был создан шаблон. Вы можете изменять интересующие параметры задания, оставляя другие неизменными, и экономя, таким образом, время на редактировании. Например, Вы можете увеличить число разбиений отрезка интегрирования при вычислении числа Пи из прошлого примера (используемый алгоритм вычисления числа Пи

сводится к численному вычислению определенного интеграла), оставив остальные параметры теми же.



Запуск программы из командной строки

Часто бывает удобней управлять ходом выполнения заданий из командной строки. Microsoft Compute Cluster Server 2003 имеет в своем составе текстовые утилиты, предоставляющие полный контроль за ходом выполнения заданий на кластере.

В данной лабораторной работе мы покажем, как из командной строки запускать последовательные задачи. Запуск параллельных задач, а также создание параметрического множества и потока задач может быть также осуществлен из командной строки. Дополнительная информация о командах и их параметрах содержится в документации, поставляемой с Microsoft Compute Cluster Pack.

Для запуска последовательного алгоритма вычисления числа Пи выполните следующие действия:

- Откройте командный интерпретатор (**Start->Run**, введите команду **"cmd"** и нажмите **"ввод"**),
- Для создания нового задания введите команду **"job new /jobname:SerialPiCL /scheduler:s-cw-head"** (не забудьте заменить параметры команды на соответствующие Вашему случаю), где параметр **"jobname"** – имя добавляемого задания, **"scheduler"** – имя головного узла кластера. Команда напечатает идентификатор созданного задания, с которым Вы будете работать далее

```
C:\ Command Prompt
D:\Projects\senin\mpi_test\IMB_K\release>job new /jobname:SerialPiCL /scheduler:
s-cw-head
Job queued, ID: 26
D:\Projects\senin\mpi_test\IMB_K\release>_
```

- Для добавления новой задачи в задание введите команду “**job add 26 /numprocessors:1 /scheduler:s-cw-head /stdout://s-cw-head/temp/serialpi.txt /workdir://s-cw-head/temp/ serialpi.exe 1000**” (не забудьте заменить параметры команды на соответствующие Вашему случаю). Здесь число “26” – идентификатор задания, возвращенный на предыдущем шаге. Параметр “**numprocessors**” задает число процессоров, необходимое данной задаче (для задания минимального и максимального числа процессоров необходимо использовать формат “**/numprocessors:x-y**”, где **x** – минимальное число процессоров, **y** – максимальное число процессоров). Параметр “**stdout**” задает файл, в который будет перенаправлен стандартный поток вывода. Параметр “**workdir**” задает директорию по умолчанию для запускаемого приложения. После параметров указывается, собственно, команда и аргументы командной строки

```
C:\ Command Prompt
D:\Projects\senin\mpi_test\IMB_K\release>job new /jobname:SerialPiCL /scheduler:
s-cw-head
Job queued, ID: 26
D:\Projects\senin\mpi_test\IMB_K\release>job add 26 /numprocessors:1 /scheduler:
s-cw-head /stdout://s-cw-head/temp/serialpi.txt /workdir://s-cw-head/temp/ seria
lpi.exe 1000
Task 26.1 added.
D:\Projects\senin\mpi_test\IMB_K\release>_
```

- Для начала планирования задания введите команду “**job submit /id:26 /scheduler:s-cw-head**” (не забудьте заменить параметры команды на соответствующие Вашему случаю). Введите пароль пользователя, под которым Вы зарегистрированы в системе. На вопрос о том, следует ли запомнить Ваш пароль, чтобы не приходилось вводить его далее, введите “**n**” для отказа

```

C:\ Command Prompt
D:\Projects\senin\mpi_test\IMB_K\release>job new /jobname:SerialPiCL /scheduler:
s-cw-head
Job queued, ID: 26
D:\Projects\senin\mpi_test\IMB_K\release>job add 26 /numprocessors:1 /scheduler:
s-cw-head /stdout://s-cw-head/temp/serialpi.txt /workdir://s-cw-head/temp/ seria
lpi.exe 1000
Task 26.1 added.
D:\Projects\senin\mpi_test\IMB_K\release>job submit /id:26 /scheduler:s-cw-head
Enter the password for 'CCAM\senin' to connect to 'S-CW-HEAD':
Remember this password? (Y/N)n
Job 26 has been submitted.
D:\Projects\senin\mpi_test\IMB_K\release>_

```

- Ваша задача была добавлена в очередь, и планировщик начал осуществлять планирование ее запуска. Вы можете посмотреть ее состояние в программе **Job Manager** или введя команду “**job list /scheduler:s-cw-head /all**” (не забудьте заменить параметры команды на соответствующие Вашему случаю).

Упражнение 3 – Запуск параллельного задания

В задании 3 упражнения 1 была скомпилирована параллельная программа вычисления числа Пи для MS MPI. Запустим теперь ее на кластере под управлением Microsoft Compute Cluster Server 2003:

- Откройте **Computer Cluster Job Manager (Start->All Programs->Microsoft Compute Cluster Pack->Compute Cluster Job Manager)** для запуска программы на кластере,

ID	Name	Priority	Submitted By	Status	Submit Time	Pending Reason
1	hostname	Normal	CCAM\Senin	Finished	21.06.2006 4:10:14	
2	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:19:50	
3	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:24:24	
4	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:25:23	
5	imb	Normal	CCAM\Senin	Finished	21.06.2006 5:13:20	
6	Serial Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 1:59:22	
7	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 2:00:46	
8	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 3:50:22	
9	Serial Pi Calculation	Normal	CCAM\Senin	Finished	25.06.2006 4:08:39	

Select a job to view its tasks						
Name	Status	Task Id	Command Line	Processors	End Time	Failure Message

- В открывшемся окне менеджера заданий выберите пункт меню **File->Submit Job** для постановки нового задания в очередь,
- В окне постановки задания в очередь введите имя задания (поле “**Job Name**”). Перейдите на вкладку “**Processors**”

Submit Job Parallel Pi computing

General | Processors | Tasks | Licenses | Advanced

Parallel Pi computing

Job Name: Parallel Pi computing

Project Name:

Priority: Normal

Submitted By: N/A

Submitted on: N/A

Status: Not Submitted

Save As Template... Submit Cancel

- На вкладке “**Processors**” введите минимальное и максимальное числа процессоров, необходимых для выполнения задания (например, 10 и 20 соответственно). Перейдите на вкладку “**Tasks**” для добавления в задание новых задач

Submit Job Parallel Pi computing

General **Processors** Tasks Licenses Advanced

Processors required for this job

Processors available in this cluster: 60

Minimum required: 10

Maximum required: 20

Estimate run time for this job

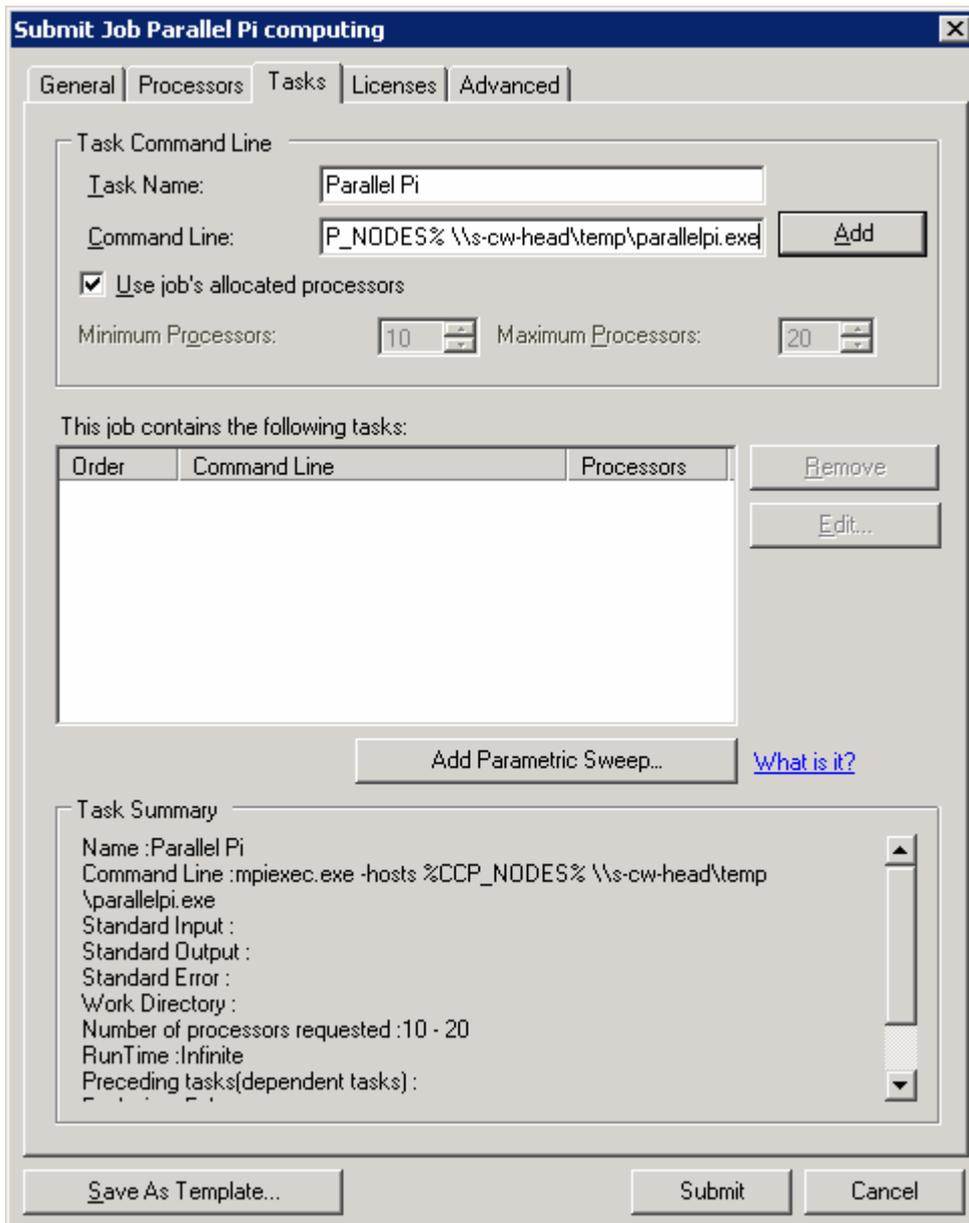
Days: 0 Hours: 1 Minutes: 0

Run job until end of run time or until canceled.

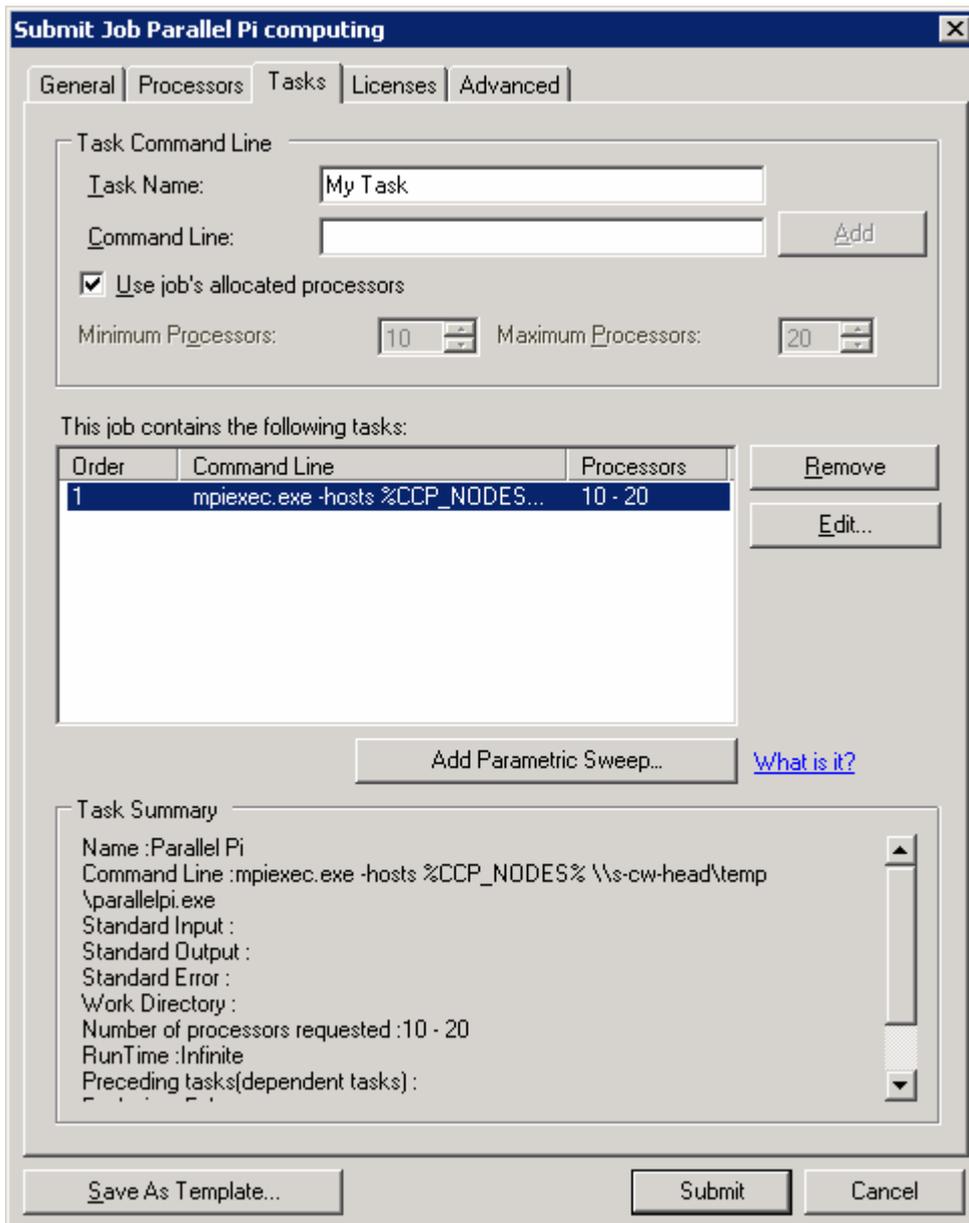
This option lets you run extra tasks after running all tasks already listed in the job if there is time left.

Save As Template... Submit Cancel

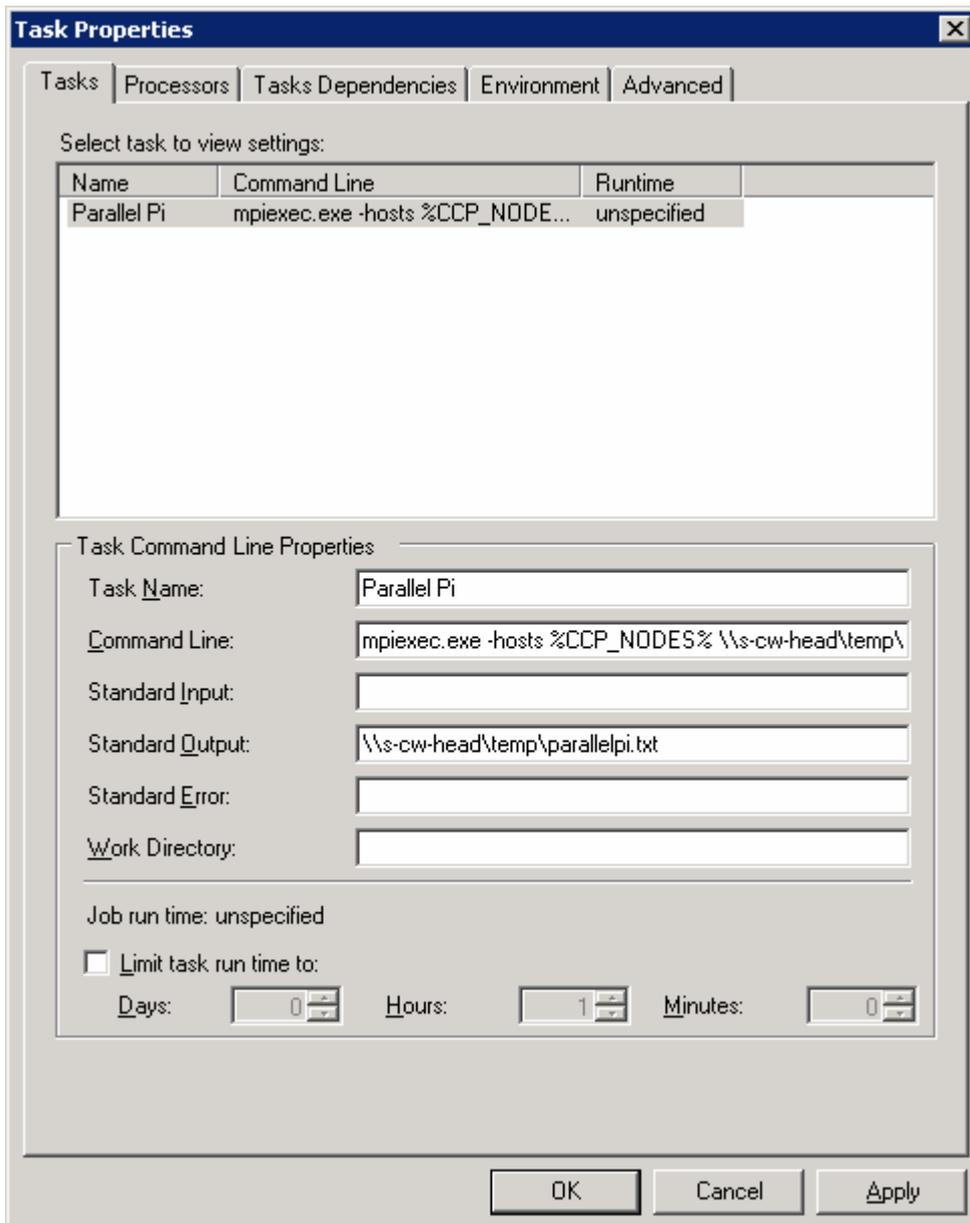
- Введите имя задачи (поле “**Task Name**”) и команду, которую необходимо выполнить (поле “**Command Line**”). Запуск задач, разработанных для MS MPI, необходимо осуществлять с использованием специальной утилиты **mpiexec.exe**, которая принимает в качестве параметров имя параллельной программы, список узлов, на которых произойдет запуск, и параметры запускаемой программы. Список узлов задается параметром “**-hosts**”. При этом в случае, если узлы были выделены планировщиком автоматически, список узлов будет содержать переменная окружения **CCP_NODES**, значение которой и следует передавать в качестве параметра утилите. Пример команды для запуска параллельной программы: “**mpiexec.exe -hosts %CCP_NODES \\s-cw-head\temp\parallempi.exe**”. Нажмите кнопку “**Add**” для добавления задачи в задание



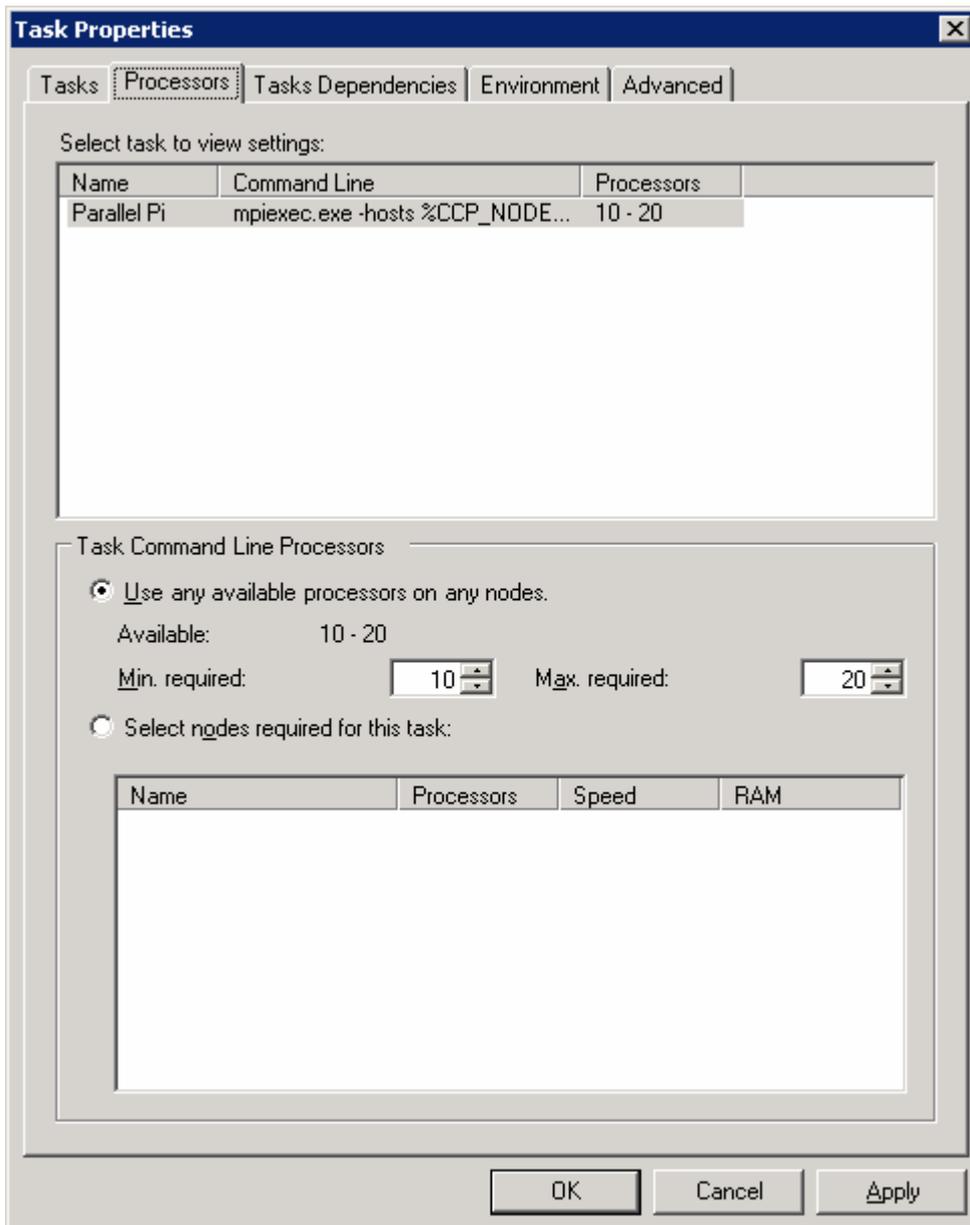
- Добавленная задача появится в списке задач текущего задания (список “**This job contains the following tasks**”). Выделите ее в списке и нажмите кнопку “**Edit**” для редактирования дополнительных параметров задачи



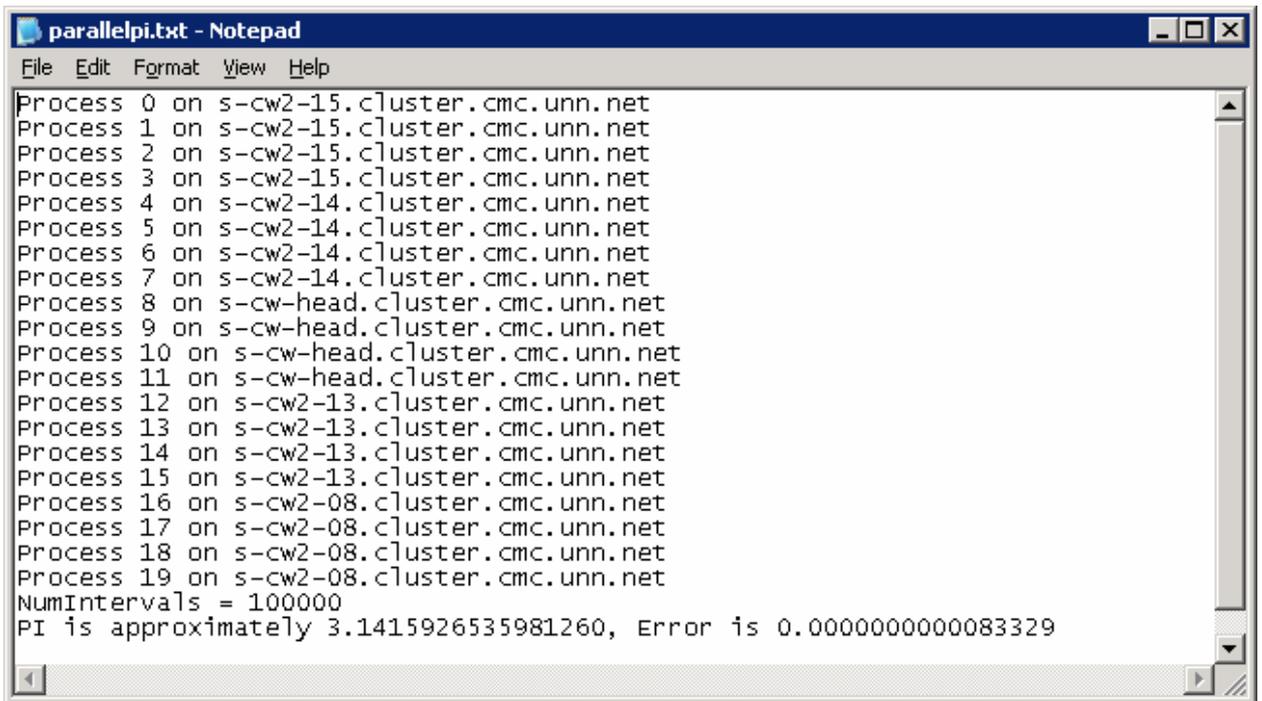
- В открывшемся окне введите путь до файла, в который будет перенаправлен стандартный поток вывода консольного приложения (поле "**Standard Output**"). Выберите вкладку "**Processors**"



- На вкладке “**Processors**” в верхнем списке (“**Select task to view settings**”) выделите задачу, настройки которой Вы хотите изменить, и укажите минимальное и максимальное числа процессоров для выбранной задачи (поля “**Min. required**” и “**Max. required**”). Нажмите “**OK**” для сохранения внесенных изменений и возвращения к настройкам задания



- Нажмите кнопку “**Submit**” для добавления задания в очередь. В окне запроса пароля введите имя и пароль пользователя, имеющего право запуска задач на кластере, и нажмите “**OK**”. Задание появится в очереди **Job Manager**. По окончании работы его состояние изменится на “**Finished**”. В файле, указанном в настройках задачи для перенаправления стандартного потока вывода, содержится результат работы программы



```
parallelpi.txt - Notepad
File Edit Format View Help
Process 0 on s-cw2-15.cluster.cmc.unn.net
Process 1 on s-cw2-15.cluster.cmc.unn.net
Process 2 on s-cw2-15.cluster.cmc.unn.net
Process 3 on s-cw2-15.cluster.cmc.unn.net
Process 4 on s-cw2-14.cluster.cmc.unn.net
Process 5 on s-cw2-14.cluster.cmc.unn.net
Process 6 on s-cw2-14.cluster.cmc.unn.net
Process 7 on s-cw2-14.cluster.cmc.unn.net
Process 8 on s-cw-head.cluster.cmc.unn.net
Process 9 on s-cw-head.cluster.cmc.unn.net
Process 10 on s-cw-head.cluster.cmc.unn.net
Process 11 on s-cw-head.cluster.cmc.unn.net
Process 12 on s-cw2-13.cluster.cmc.unn.net
Process 13 on s-cw2-13.cluster.cmc.unn.net
Process 14 on s-cw2-13.cluster.cmc.unn.net
Process 15 on s-cw2-13.cluster.cmc.unn.net
Process 16 on s-cw2-08.cluster.cmc.unn.net
Process 17 on s-cw2-08.cluster.cmc.unn.net
Process 18 on s-cw2-08.cluster.cmc.unn.net
Process 19 on s-cw2-08.cluster.cmc.unn.net
NumIntervals = 100000
PI is approximately 3.1415926535981260, Error is 0.0000000000083329
```

Упражнение 4 – Запуск множества задач

В данном упражнении идет речь о запуске параметрического множества задач (parametric sweep) внутри одного задания. Под параметрическим множеством задач понимается серия запусков одной и той же программы с разными параметрами. В качестве примера можно привести запуск серии из нескольких сотен экспериментов по вычислению числа Пи для исследования скорости сходимости метода к точному решению. В качестве примера программы для данного упражнения мы будем использовать программу параллельного вычисления числа Пи:

- Откройте **Computer Cluster Job Manager (Start->All Programs->Microsoft Compute Cluster Pack->Compute Cluster Job Manager)** для создания параметрического множества задач

Job Queue at s-cw-head

File View Help Show: All Jobs

ID	Name	Priority	Submitted By	Status	Submit Time	Pending Reason
1	hostname	Normal	CCAM\Senin	Finished	21.06.2006 4:10:14	
2	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:19:50	
3	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:24:24	
4	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:25:23	
5	imb	Normal	CCAM\Senin	Finished	21.06.2006 5:13:20	
6	Serial Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 1:59:22	
7	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 2:00:46	
8	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 3:50:22	
9	Serial Pi Calculation	Normal	CCAM\Senin	Finished	25.06.2006 4:08:39	
10	Parallel Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 22:13:00	
11	Parallel Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 22:15:36	
12	Parallel Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 22:16:49	
13	Parallel Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 22:17:33	
14	Parallel Pi computing	Normal	CCAM\Senin	Finished	26.06.2006 0:29:12	
15	Parallel Pi computing	Normal	CCAM\Senin	Finished	26.06.2006 0:31:30	

Select a job to view its tasks

Name	Status	Task Id	Command Line	Processors	End Time	Failure Message

Ready

- В открывшемся окне менеджера заданий выберите пункт меню **File->Submit Job** для постановки нового задания в очередь,
- В окне постановки задания в очередь введите имя задания (поле “**Job Name**”). Перейдите на вкладку “**Processors**”

Submit Job Parallel Pi parametric sweep

General Processors Tasks Licenses Advanced

Parallel Pi parametric sweep

Job Name: Parallel Pi parametric sweep

Project Name:

Priority: Normal

Submitted By: N/A

Submitted on: N/A

Status: Not Submitted

Save As Template... Submit Cancel

- На вкладке “**Processors**” введите минимальное и максимальное числа процессоров, необходимых для выполнения задания (например, 10 и 20 соответственно). Перейдите на вкладку “**Tasks**” и нажмите кнопку “**Add parametric Sweep**” на вкладке для добавления в задание множества новых задач

Submit Job Parallel Pi parametric sweep

General Processors Tasks Licenses Advanced

Processors required for this job

Processors available in this cluster: 60

Minimum required: 10

Maximum required: 20

Estimate run time for this job

Days: 0 Hours: 1 Minutes: 0

Run job until end of run time or until canceled.

This option lets you run extra tasks after running all tasks already listed in the job if there is time left.

Save As Template... Submit Cancel

- В окне добавления параметрического множества задач введите имя, которое будет присвоено каждой новой задаче (поле “Name”). Введите команду для задач, используя звездочку (“*”) как параметр аргументов командной строки. Символ “*” для каждой конкретной команды будет заменен целым числом, пределы изменения которого указываются в полях “Index Start” и “Index End”. Для нашей задачи индекс (число отрезков численного интегрирования) может изменяться, например, от 50 до 100. Таким образом, команда может быть следующей: “**mpiexec.exe -hosts %CCP_NODES% \\s-cw-head\temp\parallempi.exe ***”. Укажите файлы, в которые будет перенаправлен стандартный поток вывода, используя “*” в качестве параметра. Например: “**\\s-cw-head\temp\parallempi*.txt**”. Нажмите “OK” для добавления множества задач в задание

Add Parametric Sweep

Create Task

Name: Parallel Pi

Command Line (Use * to represent index if desired):
NODES% \\s-cw-head\temp\parallelpi.exe *

Index Start: 50 Index End: 100 Index Skip: 0

Use Standard Input (Use * to represent index if desired)
Input File Location

Collect Standard Output (Use * to represent index if desired)
Output File Location: \\s-cw-head\temp\parallelpi*.txt

Collect Standard Error (Use * to represent index if desired)
Error File Location

Assign Work Directory
Work Directory

Preview Task

Command Line	Standard Output
Parallel Pi	
mpirun -hosts %CCP_NODES% \\s-cw-head\temp\parallelpi.exe 50	\\s-cw-head\ter
mpirun -hosts %CCP_NODES% \\s-cw-head\temp\parallelpi.exe 51	\\s-cw-head\ter
mpirun -hosts %CCP_NODES% \\s-cw-head\temp\parallelpi.exe 52	\\s-cw-head\ter

Extension Length: 1

Add Cancel

- В окне настройки задания выделите все входящие в него задачи (для выделения нескольких задач используйте клавишу “Shift”), нажмите кнопку “Edit” для указания числа процессоров, требуемых для задач. В открывшемся окне перейдите на вкладку “Processors”, выберите пункт “Use any available processors on any nodes” и укажите, например, в качестве минимального числа процессоров 10, в качестве максимального – 20. Нажмите “OK”. В открывшемся окне нажмите кнопку “Submit” для добавления задания в очередь

- В открывшемся окне **Job Manager** появится новое задание. Выделив его, Вы получите возможность наблюдать за ходом выполнения его задач в нижнем списке. Когда задание выполнится, его состояние изменится на **“Finished”**

The screenshot shows the 'Job Queue at s-cw-head' window. The main table lists jobs with columns: ID, Name, Priority, Submitted By, Status, Submit Time, and Pending Reason. Job 16, 'Parallel Pi parametric sweep', is highlighted and has a status of 'Running'.

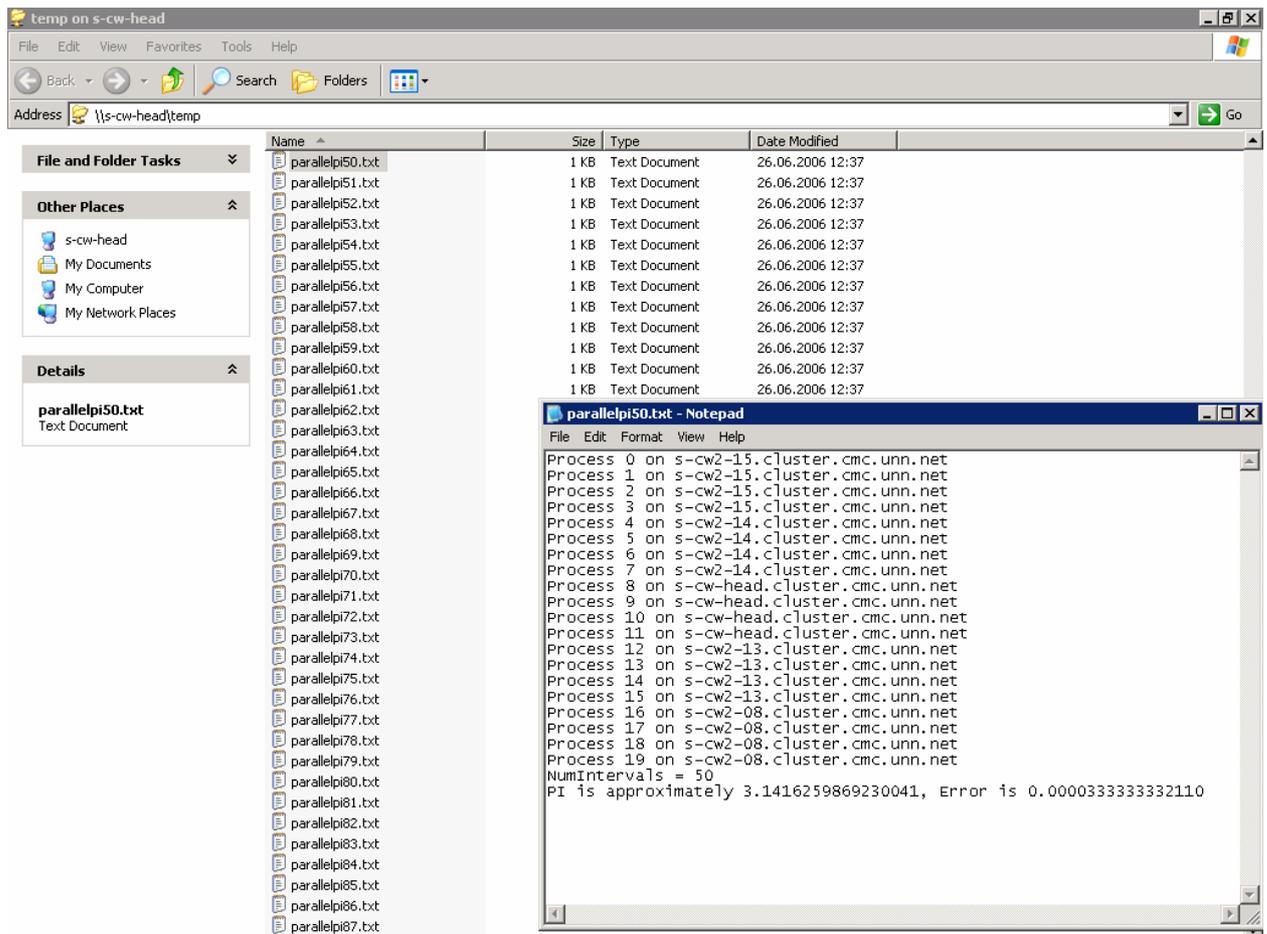
ID	Name	Priority	Submitted By	Status	Submit Time	Pending Reason
1	hostname	Normal	CCAM\Senin	Finished	21.06.2006 4:10:14	
2	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:19:50	
3	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:24:24	
4	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:25:23	
5	imb	Normal	CCAM\Senin	Finished	21.06.2006 5:13:20	
6	Serial Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 1:59:22	
7	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 2:00:46	
8	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 3:50:22	
9	Serial Pi Calculation	Normal	CCAM\Senin	Finished	25.06.2006 4:08:39	
10	Parallel Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 22:13:00	
11	Parallel Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 22:15:36	
12	Parallel Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 22:16:49	
13	Parallel Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 22:17:33	
14	Parallel Pi computing	Normal	CCAM\Senin	Finished	26.06.2006 0:29:12	
15	Parallel Pi computing	Normal	CCAM\Senin	Finished	26.06.2006 0:31:30	
16	Parallel Pi parametric sweep	Normal	CCAM\Senin	Running	26.06.2006 1:37:01	

Below the main table, there is a section titled 'Tasks for ParallelPi parametric sweep' with a detailed table:

Name	Status	Task Id	Command Line	Processors	End Time	Failure Message
Parallel Pi	Finished	1	mpiexec -hosts %CCP_NO...	10 - 20	26.06.2006 ...	
Parallel Pi	Running	2	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	3	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	4	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	5	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	6	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	7	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	8	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	9	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	10	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	11	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	12	mpiexec -hosts %CCP_NO...	10 - 20	N/A	
Parallel Pi	Queued	13	mpiexec -hosts %CCP_NO...	10 - 20	N/A	

A tooltip from 'Compute Cluster Job Manager' is visible in the bottom right corner, stating: 'A new job 16 is submitted. Please click here for detailed information.'

- Вы можете просмотреть результаты выполнения заданий, в файлах, которые Вы указали для сохранения перехваченного потока вывода



Упражнение 5 – Запуск потока задач

Поток задач используется тогда, когда для выполнения некоторой задачи в составе задания необходимы результаты работы других задач, что выдвигает требования к порядку их выполнения. Эти требования удобно задавать в виде ациклического ориентированного графа, в котором каждая вершина представляет собой задачу, а стрелка выражает зависимость вершины-потомка от вершины-родителя. В этом случае порядок выполнения задач определяется следующим простым правилом: ни одна задача не начнет выполняться до тех пор, пока не будут выполнены все задачи, соответствующие на графе зависимостей ее родителям.

Примером может служить следующий граф зависимостей задач.



Реализуем граф зависимостей в CCS 2003:

- Откройте **Computer Cluster Job Manager (Start->All Programs->Microsoft Compute Cluster Pack->Compute Cluster Job Manager)**

ID	Name	Priority	Submitted By	Status	Submit Time	Pending Reason
1	hostname	Normal	CCAM\Senin	Finished	21.06.2006 4:10:14	
2	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:19:50	
3	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:24:24	
4	imb	Normal	CCAM\Senin	Failed	21.06.2006 4:25:23	
5	imb	Normal	CCAM\Senin	Finished	21.06.2006 5:13:20	
6	Serial Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 1:59:22	
7	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 2:00:46	
8	Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 3:50:22	
9	Serial Pi Calculation	Normal	CCAM\Senin	Finished	25.06.2006 4:08:39	
10	Parallel Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 22:13:00	
11	Parallel Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 22:15:36	
12	Parallel Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 22:16:49	
13	Parallel Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 22:17:33	
14	Parallel Pi computing	Normal	CCAM\Senin	Finished	26.06.2006 0:29:12	
15	Parallel Pi computing	Normal	CCAM\Senin	Finished	26.06.2006 0:31:30	
16	Parallel Pi parametric sweep	Normal	CCAM\Senin	Finished	26.06.2006 1:37:01	

Name	Status	Task Id	Command Line	Processors	End Time	Failure Message

- В открывшемся окне менеджера заданий выберите пункт меню **File->Submit Job**,
- В окне постановки задания в очередь введите имя задания (поле **“Job Name”**). Перейдите на вкладку **“Processors”**

Submit Job Example of task flow [X]

General | Processors | Tasks | Licenses | Advanced

 Example of task flow

Job Name:

Project Name:

Priority: ▼

Submitted By: N/A

Submitted on: N/A

Status: Not Submitted

- На вкладке “**Processors**” введите минимальное и максимальное числа процессоров, необходимых для выполнения задания (например, 5 и 10 соответственно). Перейдите на вкладку “**Tasks**” для добавления в задание новых задач

Submit Job Example of task flow

General Processors Tasks Licenses Advanced

Processors required for this job

Processors available in this cluster: 60

Minimum required: 5

Maximum required: 10

Estimate run time for this job

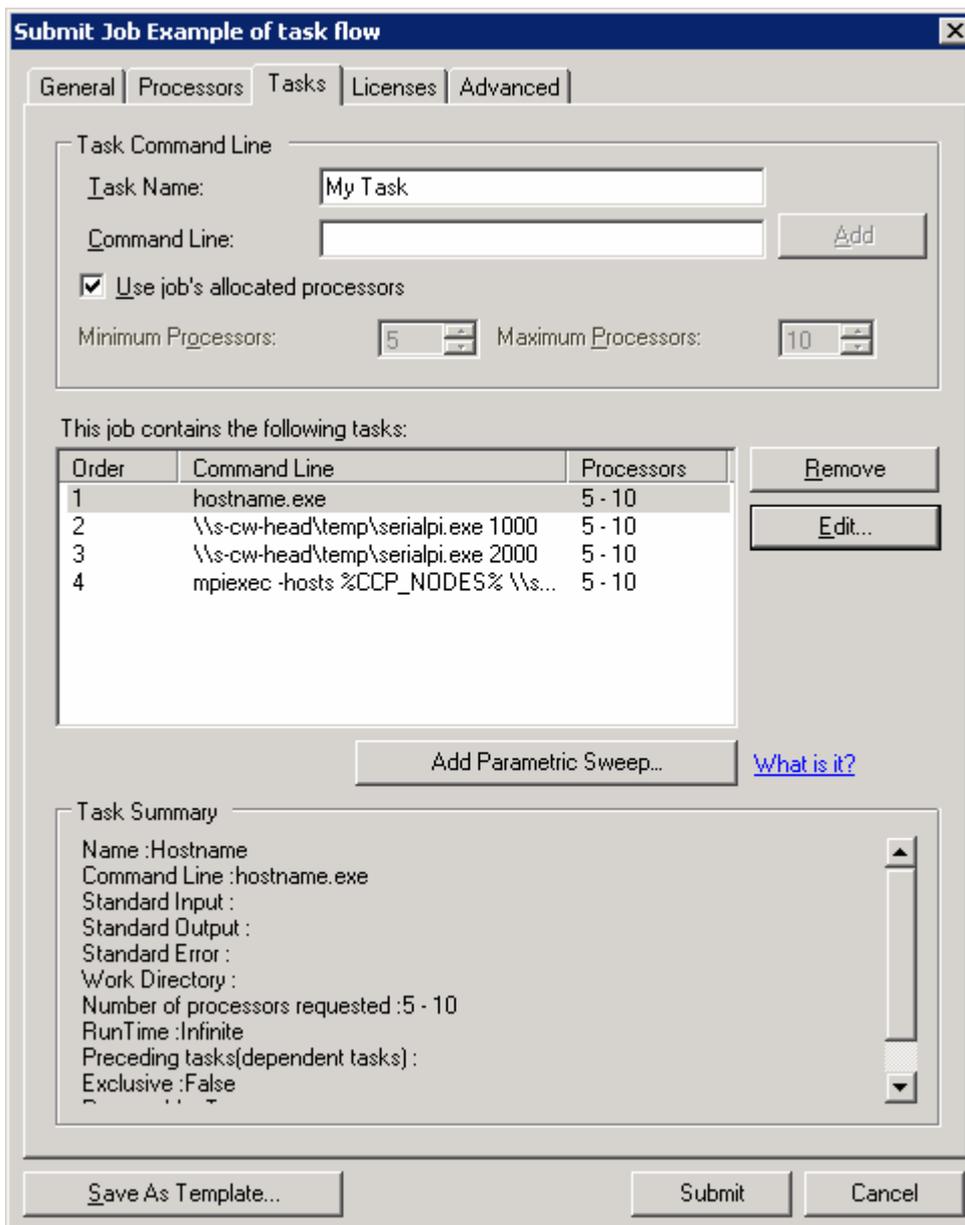
Days: 0 Hours: 1 Minutes: 0

Run job until end of run time or until canceled.

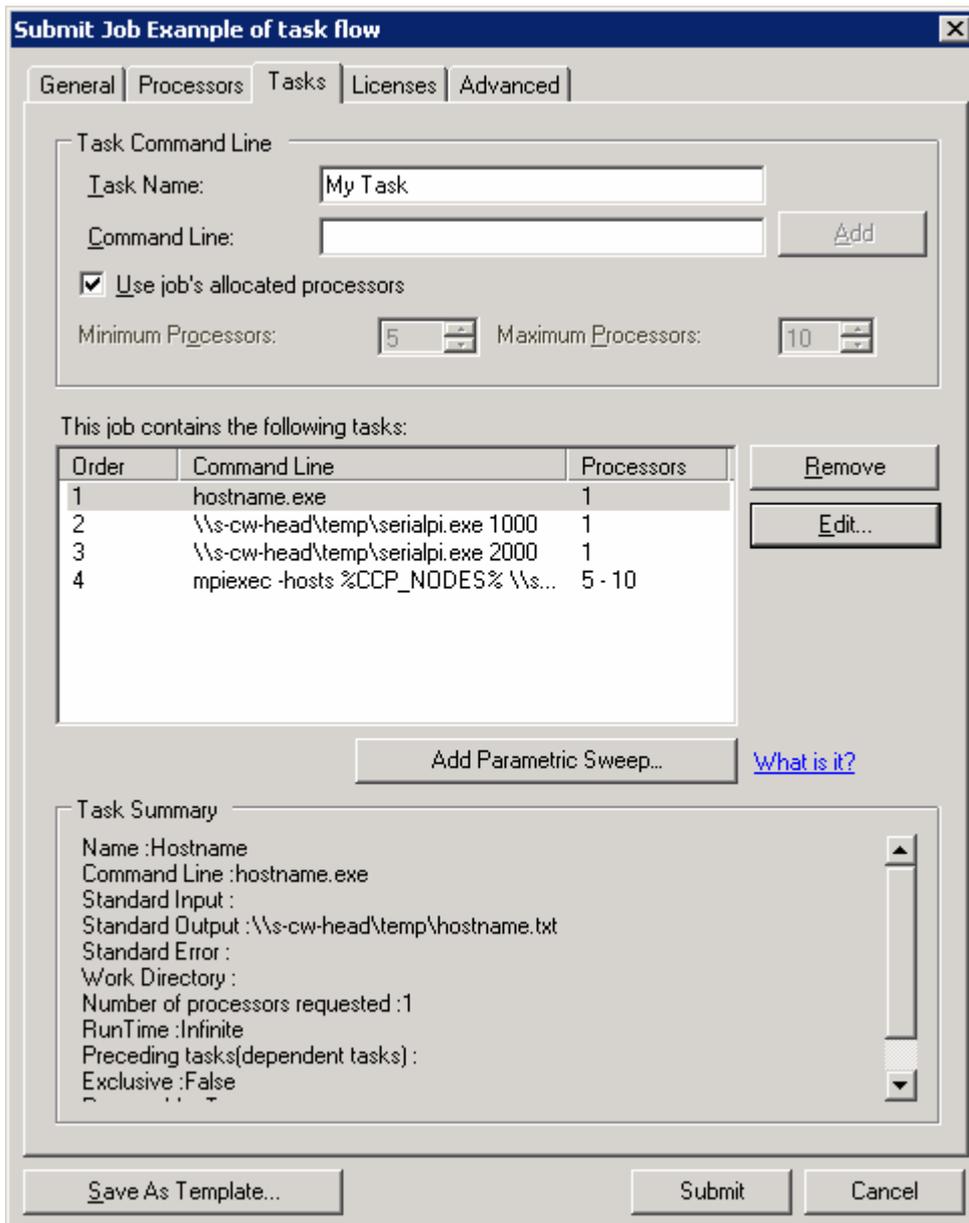
This option lets you run extra tasks after running all tasks already listed in the job if there is time left.

Save As Template... Submit Cancel

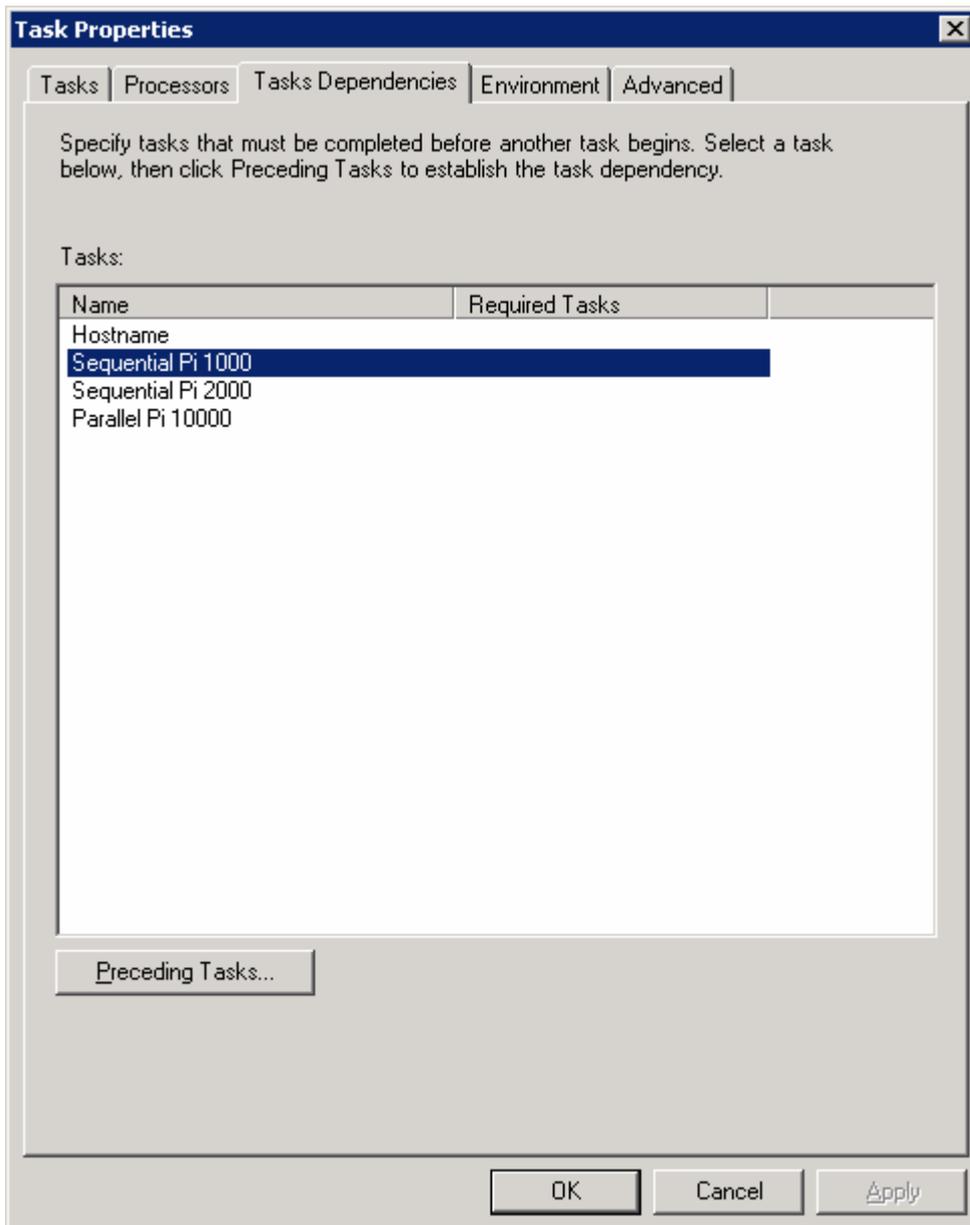
- Последовательно добавьте в задание следующие 4 задачи:
 - Задачу с именем **“Hostname”** и командой **“hostname.exe”**,
 - Задачу с именем **“Sequential Pi 1000”** и командой **“\\s-cw-head\temp\serialpi.exe 1000”** (замените путь до исполняемого файла программы на существующий),
 - Задачу с именем **“Sequential Pi 2000”** и командой **“\\s-cw-head\temp\serialpi.exe 2000”** (замените путь до исполняемого файла программы на существующий),
 - Задачу с именем **“Parallel Pi 10000”** и командой **“mpixec -hosts %CCP_NODES% \\s-cw-head\temp\parallepi.exe 10000”** (замените путь до исполняемого файла программы на существующий)



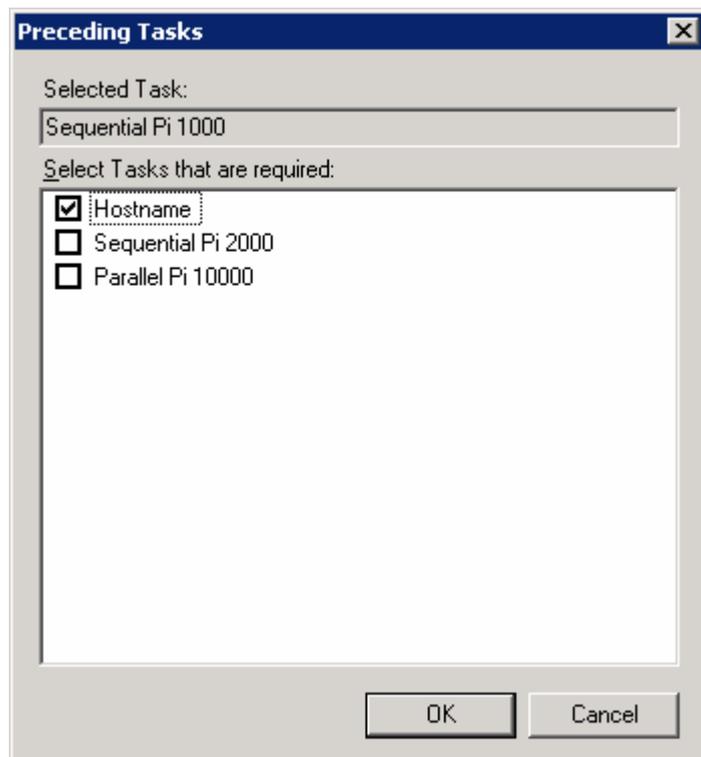
- Установите дополнительные свойства задач:
 - Для задач “**Hostname**”, “**Sequential Pi 1000**” и “**Sequential Pi 2000**” установите максимальное необходимое число процессоров в 1, установите для каждой из 3 задач файл для переправления стандартного потока вывода,
 - Для задачи “**Parallel Pi 10000**” установите минимальное и максимальное числа процессоров в 5 и 10 соответственно, установите файл для переправления стандартного потока вывода,
- Перейдите на вкладку “**Tasks Dependencies**” свойств задач (для перехода к свойствам задач нажмите кнопку “**Edit**” на вкладке “**Tasks**” окна “**Submit Job**”)



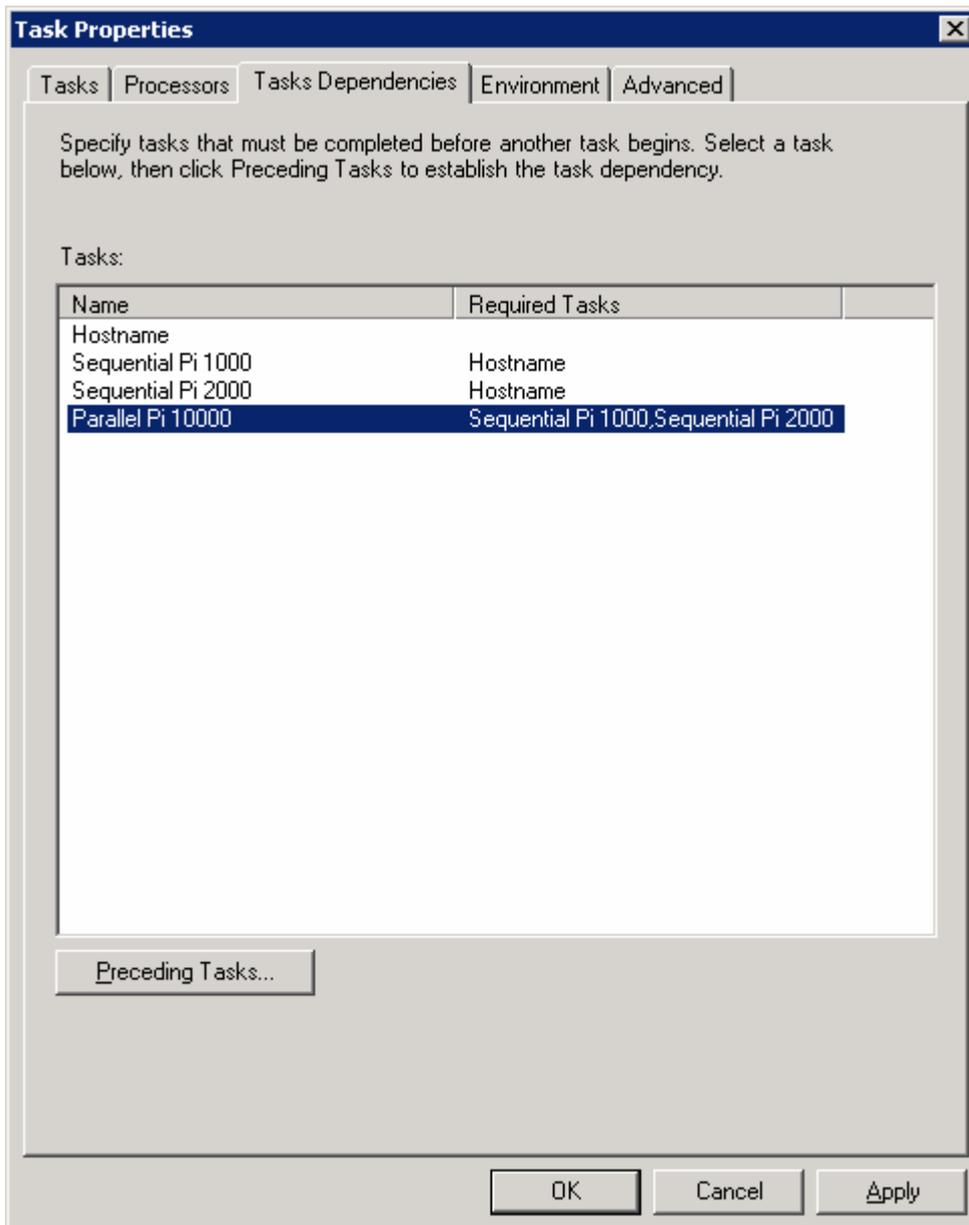
- Выделите задачу “**Sequential Pi 1000**” и нажмите кнопку “**Preceding Tasks**” для указания задач, от которых зависит рассматриваемая задача



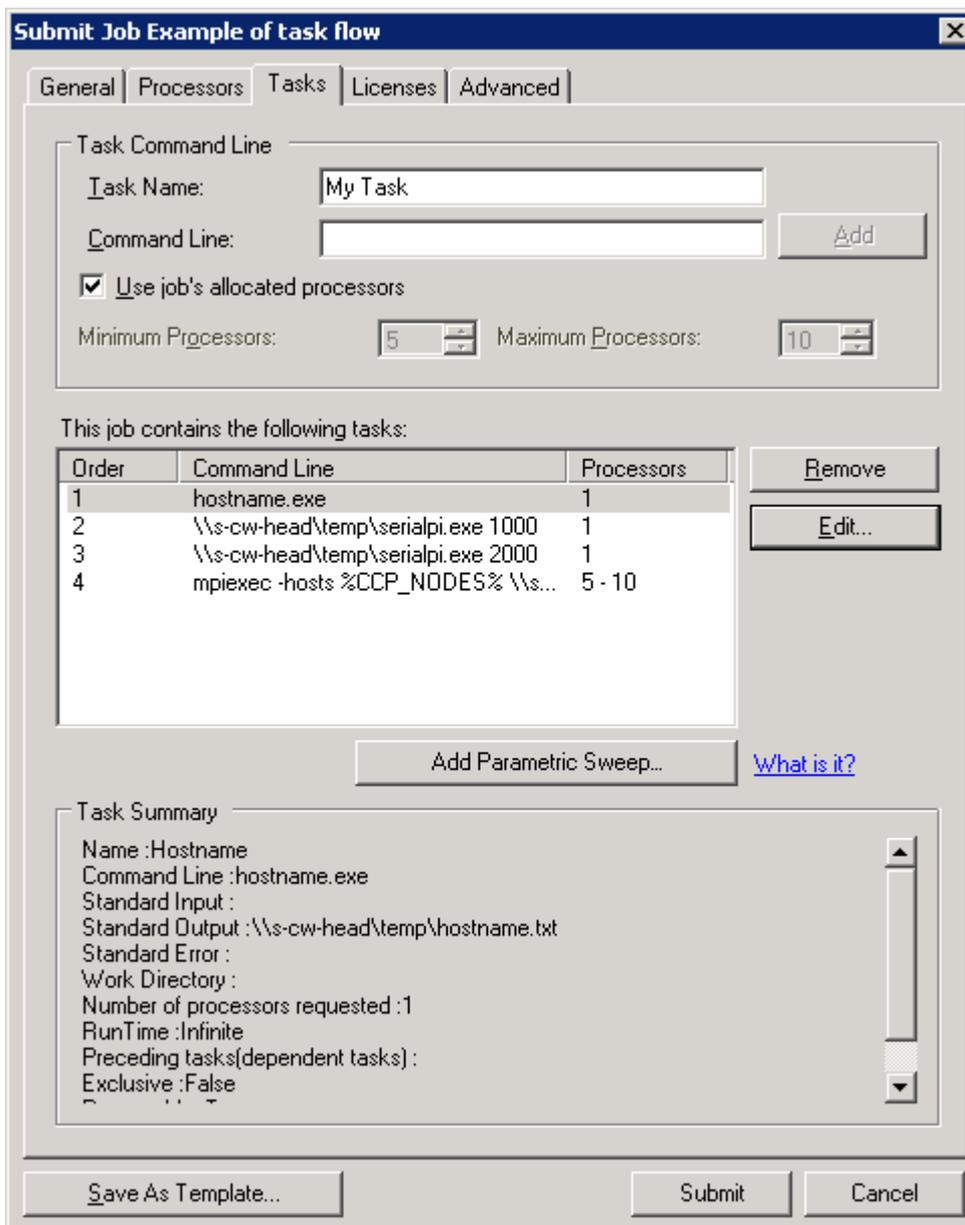
- В открывшемся окне поставьте флажок около задачи “**Hostname**”. Нажмите “**OK**”



- Установить для задачи **“Sequential Pi 2000”** зависимость от **“Hostname”**. Установите для задачи **“Parallel Pi 10000”** зависимость от задач **“Sequential Pi 1000”** и **“Sequential Pi 2000”**. Нажмите **“OK”** для сохранения внесенных изменений



- В окне “**Submit Job**” нажмите кнопку “**Submit**” для добавления задания в очередь



- Введите имя и пароль пользователя с правами запуска заданий на кластере,
- Планировщик CCS 2003 сначала запустит задачу “**Hostname**”, затем параллельно задачи “**Sequential Pi 1000**” и “**Sequential Pi 2000**” и только потом задачу “**Parallel Pi 10000**”.

Дополнительное упражнение. Задача определения характеристик сети передачи данных

Особенностью разработки программ для кластерных систем является необходимость учитывать не только характеристики отдельных компьютеров (прежде всего это производительность процессора и скорость памяти), но и характеристики сети передачи данных между ними. Чаще всего эти характеристики используются для построения теоретических оценок времени работы алгоритмов, что позволяет предсказывать время работы программ в зависимости от размера входных данных. Получение характеристик сети – отдельная задача, решаемая запуском специальных тестовых программ на конкретном имеющемся оборудовании. Необходимость проведения тестов для каждого конкретного кластера объясняется тем, что данные, предоставляемые поставщиком аппаратного обеспечения, могут сильно отличаться в зависимости от используемого программного обеспечения, настроек кластера или особенностей взаимодействия оборудования разных моделей и изготовителей.

Описание характеристик, определяющих производительность сети

Основными характеристиками, определяющими производительность сети, являются **латентность** и **пропускная способность**. Латентностью (задержкой) называется время, затрачиваемое аппаратной и

программной частью на обработку запроса отправки сетевого сообщения. То есть это время с момента поступления команды на пересылку информации до начала ее непосредственной передачи. Обычно латентность указывается в **микросекундах (мкс)**.

Пропускной способностью сети называется количество информации, передаваемое между узлами сети за единицу времени. Обычно пропускная способность сети указывается в **мегабайтах в секунду (Мбайт/сек)** или **мегабитах в секунду (Мбит/сек)**.

Общая характеристика алгоритма

Идея алгоритма определения характеристик сети, используемого в тестах данной работы, состоит в последовательной пересылке между 2 узлами сообщений различной длины, используя средства установленной реализации MPI, и измерения времени, затрачиваемого на пересылку. Имея эти данные, пропускную способность можно определить, поделив длину переданного сообщения на затраченное на передачу время. Для минимизации погрешности передачу повторяют N раз и усредняют результат. При этом оценка пропускной способности обычно увеличивается с ростом длины сообщения, стремясь к некоторому предельному значению. Обычно именно предельное значение (или значение, полученное при пересылке большого сообщения) целесообразно использовать как оценку пропускной способности.

За латентность обычно принимается время, затрачиваемое для пересылки сообщений нулевой длины.

В данной лабораторной работе рассматриваются 2 тестовые программы: **Intel MPI Benchmark (IMB)** и **набор тестов, разработанный в НИВЦ МГУ**.

Компиляция программы

Последнюю версию тестового пакета IMB в составе Intel Cluster Tools Вы можете скачать на сайте Intel по адресу <http://www.intel.com/cd/software/products/asmo-na/eng/cluster/mpi/219848.htm> . Для того, чтобы скомпилировать IMB для ОС Windows, Вам придется самостоятельно создать проект в Microsoft Visual Studio 2005 аналогично проектам, создание которых было описано в данной работе. Либо Вы можете воспользоваться заготовкой проекта, которая входит в состав лабораторной работы (папка **IMB_2_3**).

Скачать тесты НИВЦ МГУ можно по адресу <http://parallel.ru/ftp/tests/mpi-bench-suite.zip> . Для компиляции тестов в ОС Windows Вам также придется самостоятельно создать проект в Microsoft Visual Studio 2005, либо воспользоваться заготовкой проекта, входящей в состав лабораторной работы (папка **MGU_tests**).

Выполнение программы

Тесты необходимо запускать на 2 узлах сети по одному процессу на каждом узле. Таким образом, для запуска тестов в CCS 2003 необходимо указать в качестве требований для задания суммарное число процессоров на 2 узлах и выбрать эти узлы вручную:

- Откройте **Job Manager**. Выполните команду меню **File->Submit Job...** Укажите имя задачи и перейдите на вкладку **“Processors”**

Submit Job Network test [X]

General | Processors | Tasks | Licenses | Advanced

 Network test

Job Name:

Project Name:

Priority: ▼

Submitted By: N/A

Submitted on: N/A

Status: Not Submitted

- Укажите в качестве требования к заданию суммарное число процессоров на 2 вычислительных узлах, на которых Вы хотите произвести запуск (например, 8 в случае использования 2 4х процессорных узлов). Перейдите на вкладку “**Tasks**”

Submit Job Network test

General Processors Tasks Licenses Advanced

Processors required for this job

Processors available in this cluster: 60

Minimum required: 8

Maximum required: 8

Estimate run time for this job

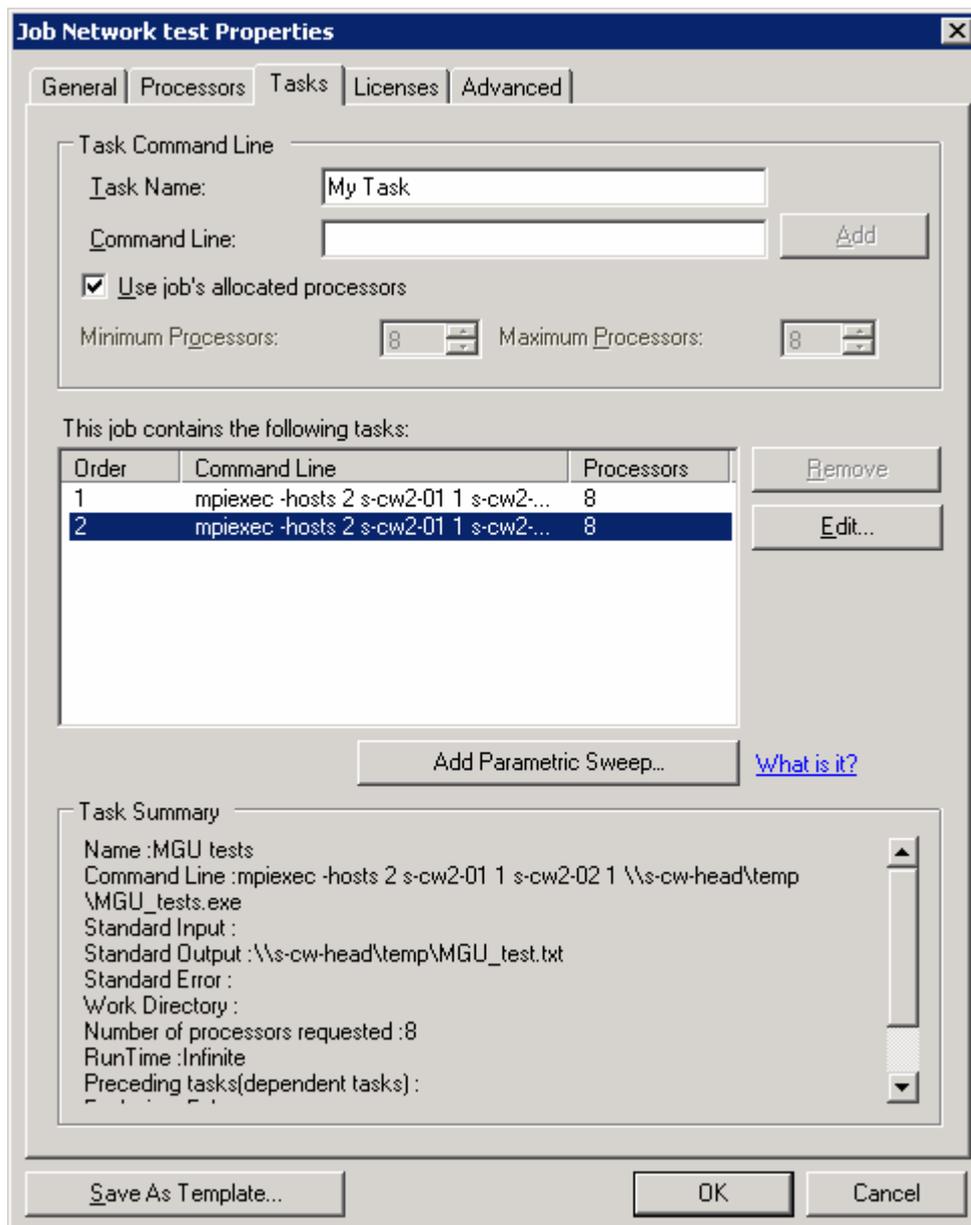
Days: 0 Hours: 1 Minutes: 0

Run job until end of run time or until canceled.

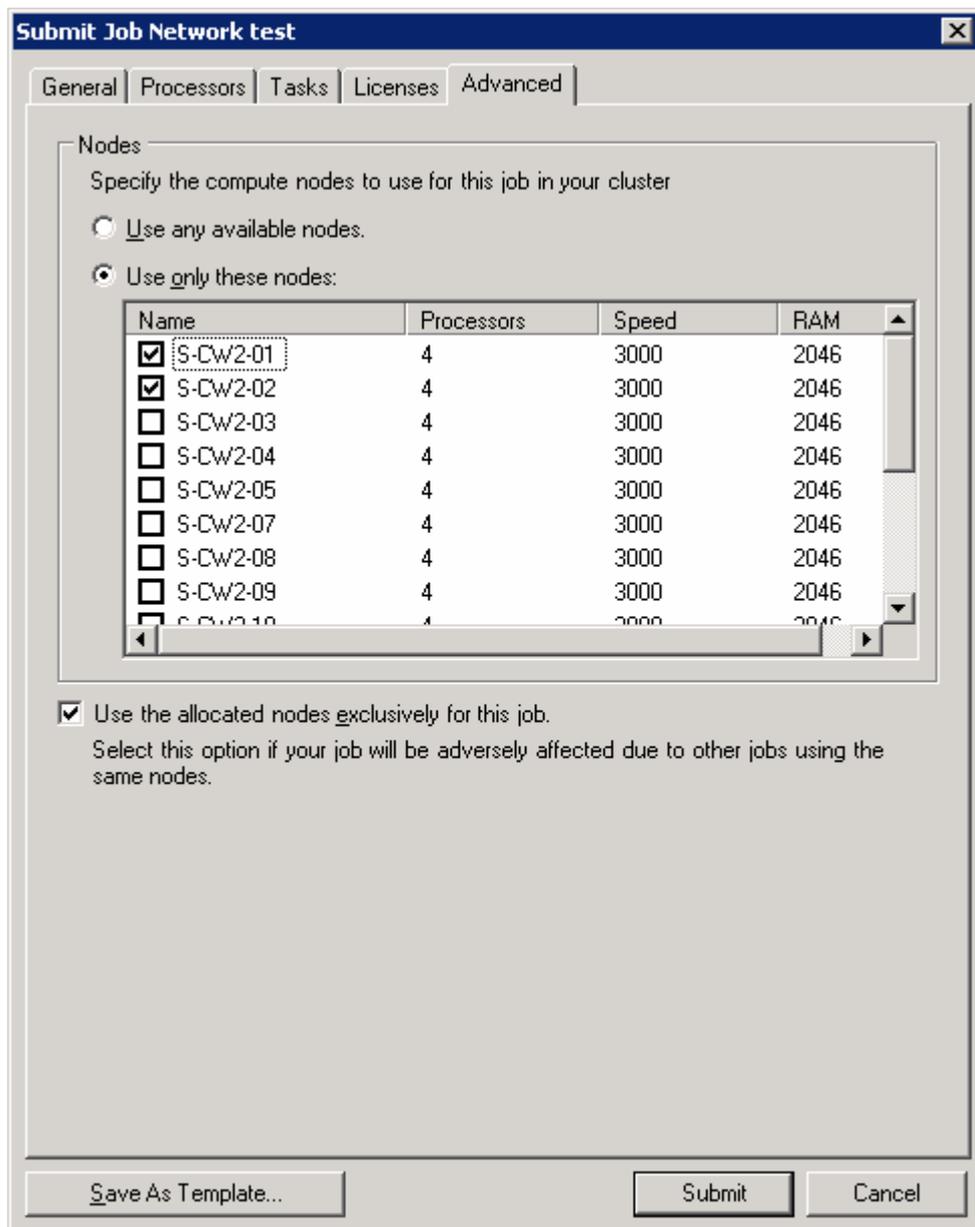
This option lets you run extra tasks after running all tasks already listed in the job if there is time left.

Save As Template... Submit Cancel

- Добавьте в задание две задачи: “`mpixec -hosts 2 s-cw2-01 1 s-cw2-02 1 \\s-cw-head\temp\imb.exe`”, “`mpixec -hosts 2 s-cw2-01 1 s-cw2-02 1 \\s-cw-head\temp\MGU_tests.exe`” (не забудьте заменить параметры команды на соответствующие Вашему случаю). В командах должны быть указаны именно те узлы, на которых планируется запуск. Параметр “hosts” имеет следующий формат: “**n node1 m1 node2 m2 ... noden mn**”. Использовать переменную окружения **CCP_NODES** в данном случае нельзя, так как на каждом узле должен быть запущен только 1 процесс. Укажите для задач файлы для перенаправления стандартного потока вывода. Перейдите на вкладку “Advanced”



- Выберите пункт “Use only these nodes” и поставьте флажки около тех узлов, которые были указаны в командах на предыдущем шаге. Нажмите кнопку “Submit” для добавления задания в очередь и введите имя и пароль пользователя, имеющего права запуска заданий на кластере



- Файлы, в которые был переправлен поток вывода, содержат результаты работы тестов. Важно отметить, что ИМВ проводит много различных тестов, но нас для получения характеристик сети интересует только первый из них, **PingPong**, так как он передает данные между двумя узлами сети используя блокирующие функции MPI_Send и MPI_Recv, что, вероятно, наилучшим образом подходит для оценки характеристик сети.

```
IMB_test.txt - Notepad
File Edit Format View Help
#-----
# Benchmarking PingPong
# #processes = 2
#-----
#
#bytes #repetitions      t[usec]  Mbytes/sec
#-----
0      1000      131.62   0.00
1      1000      130.68   0.01
2      1000      130.47   0.01
4      1000      128.91   0.03
8      1000      127.64   0.06
16     1000      124.63   0.12
32     1000      124.85   0.24
64     1000      123.77   0.49
128    1000      123.78   0.99
256    1000      125.24   1.95
512    1000      128.76   3.79
1024   1000      129.25   7.56
2048   1000      137.03   14.25
4096   1000      254.61   28.34
8192   1000      364.53   42.73
16384  1000      665.65   81.16
32768  1000      1291.61  161.51
65536  640       880.04   108.16
131072 320       1993.63  249.20

MGU_test.txt - Notepad
File Edit Format View Help
Size(b) Transfer (MB/sec)

Iteration 0
[0 -- 1] Latency: 127.272 microseconds (at 20 times)

1024    6.878
2048    8.065
3072   11.79
4096   15.71
5120   20.11
6144   22.67
7168   25.65
8192   26.35
9216   24.59
10240  26.89
11264  29.62
12288  32.4
13312  34.92
14336  37.72
15360  40.29
16384  42.98
17408  44.91
```

Контрольные вопросы

- Дайте определения терминам задание (job) и задача (task). В чем основные отличия?
- Какие основные настройки Microsoft Visual Studio 2005 необходимо произвести при компиляции параллельной программы для использования в среде MS MPI?
- В чем особенность запуска параллельных задач (скомпилированных для MS MPI) на кластере?
- Что такое параметрическое множество задач (parametric sweep)? Что такое поток задач (task flow)?
- Какие характеристики, определяющие производительность сети, Вы знаете? Дайте их определения.