Compute Cluster Server Lab 2: Carrying out Jobs under Microsoft Compute Cluster Server 2003

Compute Cluster Server Lab 2: Carrying out Jobs under Microsoft Compute Cluster Server 2003	1
Lab Objective	1
General Scheme of Carrying out the Jobs under Microsoft Compute Cluster Server 2003	1
Exercise 1 – Compiling a Program for Running under CCS 2003	3
Task 1 – Installation of Microsoft Compute Cluster Pack SDK	3
Task 2 – Setting the Development Integration Environment of Microsoft Visual Studio 2005	6
Task 3 – Compiling a Parallel Program in Microsoft Visual Studio 2005	9
Exercise 2 – Running a Serial Task 14	4
Launching the Program via Graphic User Interface	14
Launching the Program by Means of Template	23
Launching the Program from the Command Line	
Exercise 3 – Launching a Parallel Job 2	9
Exercise 4 – Launching a Parametric Sweep	6
Exercise 5 – Launching a Work Flow 4	3
Optional Exercise. Evaluating the Network Performance Parameters	2
General Network Performance Parameters	52
Methods for Evaluating the Network Performance Parameters	53
Compiling the Benchmark Program	53
Running the Benchmarks	53
Discussions	8

In order to use the high performance cluster effectively, it is necessary to use a family of quite complicated software systems. For a long time the users of Windows clusters have to use simultaneously the software from several vendors. With the release of Compute Cluster Server 2003 (CCS) Microsoft company provided a full spectrum of software, which is necessary for the efficient use of clusters and the development of the parallel programs, which fully use the available computational power.

Lab Objective

The lab objective is to learn how to compile and start programs under the management of Microsoft Compute Cluster Server 2003. The lab assignments include:

- Exercise 1 Compile a program to be run under CCS 2003.
- Exercise 2 Launch a serial job
- Exercise 3 Launch a parallel job
- Exercise 4 Launch a parametric sweep
- Exercise 5 Launch a work flow

Estimated time to complete this lab: 90 minutes.

General Scheme of Carrying out the Jobs under Microsoft Compute Cluster Server 2003

To use the computational cluster resources efficiently, it is necessary to provide not only the immediate mechanisms of starting the jobs to be executed, but also to provide the management environment, which should manage the course of executing the jobs and solve the problem of efficient resource distribution. These tasks are efficiently solved with the help of the means embedded to CCS 2003.

Let us define the most important concepts used in CCS 2003:

- A job is a request for the allocation of the cluster computational resources for carrying out a task. Each job may consist of one or more tasks,
- A task is a command or a program (including the parallel ones), which has to be executed on the cluster. A task cannot exist outside a certain job. A job may contain one or more tasks,

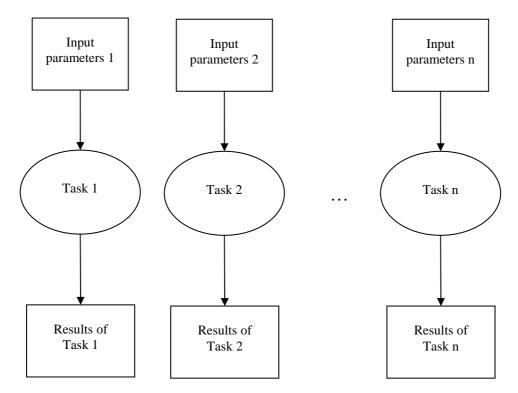
- A job scheduler is a service providing the job queue, allocating system resources, queuing the tasks and controlling the state of the executed tasks,
- A node is a computer, which is included into a cluster controlled by CCS 2003,
- A processor is one of the node computational devices,
- A **queue** is the list of tasks to be executed on the cluster, which are sent to the job scheduler. The order, in which the tasks are executed on the cluster, is determined by the planning policy adopted on the cluster,
- A task list is an equivalent of the job queue for the tasks of each concrete job. The order of executing the tasks will be determined by the FCFS strategy (first come, first served), if the user has not purposefully set some other order.

A job scheduler CCS 2003 operates both the serial and parallel tasks. The task is referred to as serial if it uses the resources of only one processor. The task is regarded to be parallel if it consists of several processes (or threads), which interact with each other in order to solve the task. As a rule, parallel MPI tasks require several processors for efficient execution. If MPI is used to provide message transmissions between parallel processes, then the parallel program may be executed on different cluster nodes. CCS 2003 includes its own realization of the standard MPI2: the library Microsoft MPI (MS MPI). If MS MPI is used, it is necessary to run parallel tasks using the special utility **mpiexec.exe**, which provides a simultaneous start of several parallel program copies on the selected cluster nodes. It should be noted that the immediate task start is the responsibility of the job scheduler, and the user can only add a program to the queue as its starting time is chosen automatically by the system depending on the availability of the computational resources and the tasks waiting in the queue for allocating the resources. Thus, it is necessary to perform the following operations to execute a program under CCS 2003:

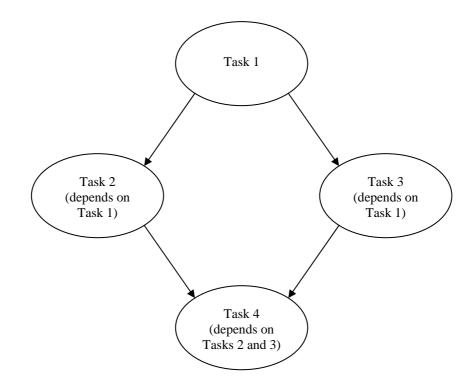
- To create a job describing the computational resources necessary for its execution,
- To create a task. The task is defined by means of a command. The execution of the command leads to running serial or parallel programs on the cluster. For instance, a parallel task is described by means of the command **mpiexec.exe** with the corresponding parameters (the list of the nodes for its executing, the name of the parallel program, the arguments of the command program line etc.),
- To add the task to the job created previously.

There are two special types of jobs:

• A parametric sweep is one and the same program (serial or parallel), several copies of the program are executed (possibly simultaneously) with various input parameters and various output files,



• A work flow is the case when several tasks (possibly the same program with different input parameters) are executed in a certain sequence. The sequence of execution is determined, for instance, by the task dependence against the computational results of the other tasks.



It will be demonstrated further in the lab, how to compile and run serial and parallel tasks under CCS 2003. Besides, there will be given examples of the parametric sweep and the work flow.

Exercise 1 – Compiling a Program for Running under CCS 2003

As it has been stated preciously, Microsoft Compute Cluster Server 2003 makes possible to control the execution of serial and parallel tasks. The parallel MPI tasks can be built with any MPI implementation (though the MS MPI implementation is preferable). Besides, it is possible to use other technologies for supporting the parallel programming (for instance, programming with the use of OpenMP).

This part describes only compiling parallel programs for MS MPI in Microsoft Visual Studio 2005.

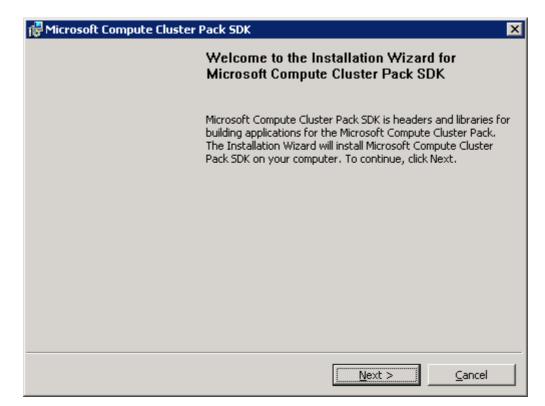
Task 1 – Installation of Microsoft Compute Cluster Pack SDK

In order to compile parallel programs operating in the environment MS MPI, it is necessary to install **SDK** (**Software Development Kit**), which is the set of interfaces and libraries for calling MPI functions:

• Open the directory containing the download version of SDK (the description of the download procedure may be found in the Compute Cluster Server Lab 1 "Installation of Microsoft Compute Cluster Server 2003") and run the installation program, which corresponds to your processor (32 bit or 64 bit version),

🗁 Compute Cluster Server SDK F	C1		
<u>File Edit View Favorites To</u>	ols <u>H</u> elp		A.
🕞 Back 🝷 🕥 🚽 🏂 🔎	Search 🔀 Folders 🛄 🕶		
Address 🛅 \\s-file\Installs\Program	ning\Microsoft\Compute Cluster Pack 2003, RC1\Comp	pute Cluster Server SDK Release Candidate 1\Compute Cluster Server SDK RC1	💌 🄁 Go
File and Folder Tasks 🛛 🗧	sdk_x64.msi Windows Installer Package 1 608 KB	sdk_x86.msi Windows Installer Package 1 018 KB	
Other Places 🛠			
Compute Cluster Server SDK Release Candidate 1 My Documents My Computer My Network Places			
Details 🏾 🛠			
sdk_x86.msi Windows Installer Package			

In the opened window press the button Next to start the installation process,



• Read the license agreement carefully. Choose the option "I accept the terms in the license agreement" in case you agree to the license agreement terms of using the system CCS 2003 and press the button Next,

1	Microsoft Compute Cluster Pack SDK License Agreement	×
	End-User License Agreement	
	Please read the following license agreement carefully	
	IMPORT ANT-READ CAREFULLY: This License Agreement for Pre-release	
	Software ("Agreement") is a legal agreement between you (either an	
	individual or a single entity, referred to in this Agreement as "Recipient")	
	and Microsoft Corporation ("Microsoft") for the Microsoft software that	
	accompanies this Agreement (as defined below). An amendment or	
	a ddendum to this Agreement may accompany the Software. YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT BY	
	INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE. IF	
	YOU DO NOT AGREE, DO NOT INSTALL, COPY, OR USE THE	T
		_
	• I accept the terms in the license agreement	
	\bigcirc I <u>d</u> o not accept the terms in the license agreement	
_		
	Print Back Next Cancel	

• Choose the directory where you are going to install SDK. To change the standard directory, press the button **Change**. Press the button **Next**,

Microsoft Compute Cluster Pack SD	к		
nstallation Folder			
Click Next to install to this folder, or click	< Change to inst	all to a different f:	older.
If you would like to change the default i	nstallation folde	er, use the section	below to do so.
Install Microsoft Compute Cluster Pack S	5DK to:		
C:\Program Files\Microsoft Compute Clu	ister Pack\		<u>C</u> hange
	< <u>B</u> ack	Next >	Cancel

• In the new window press the button Install to start the installation of SDK,

Hicrosoft Compute Cluster Pack SDK
The wizard is ready to begin installation.
Click Install to begin the installation.
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.
< <u>B</u> ack <u>Install</u> <u>C</u> ancel

• Wait till the program of SDK installation copies the required files,

🙀 Microsof	't Compute Cluster Pack SD	ж		_ 🗆 🗙
Installin	ig Microsoft Compute C	luster Pack SDI	ĸ	
	gram features you selected are			
	Please wait while the installat Pack SDK. This may take sev		icrosoft Compute Cluster	
	Status:			
				_
-				
		< <u>B</u> ack	Next >	Cancel

• After copying the necessary files, press the button Finish,

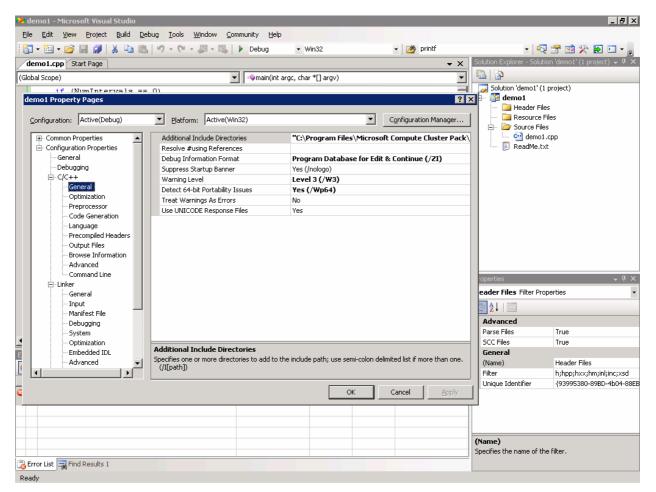
👹 Microsoft Compute Cluster	Pack SDK	X
	Microsoft Compute Cluster Pack SDK Installation Complete Setup has finished installing Microsoft Compute Cluster Pack SDK.	
	< Back Finish Cancel	

• Congratulations! The installation of Microsoft Compute Cluster Server 2003 SDK is completed.

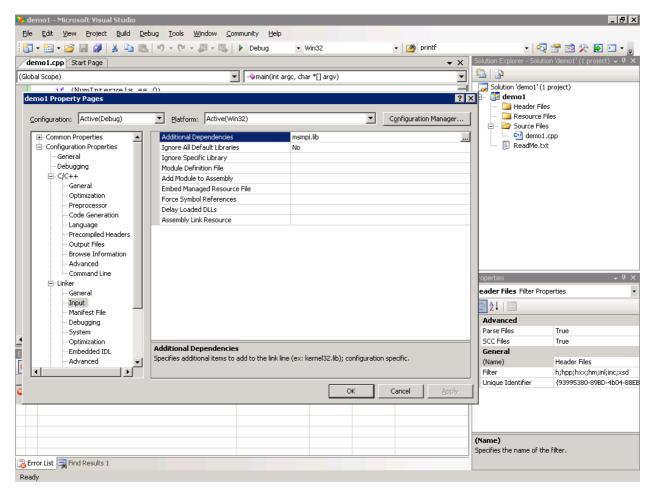
Task 2 – Setting the Development Integration Environment of Microsoft Visual Studio 2005

In order to compile the program using MS MPI, it is necessary to change the following project settings on default in Microsoft Visual Studio 2005:

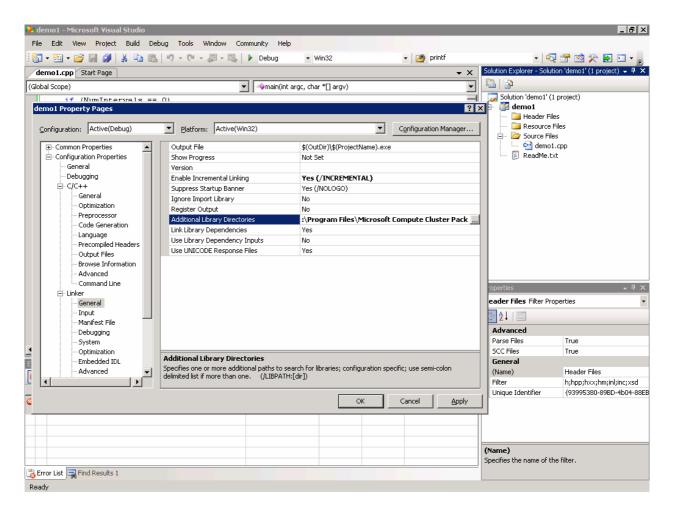
• The path to the header files of MPI declaration. Choose the menu option Project->Project Properties. In the option Configuration Properties->C++->General->Additional Include Directories enter the path to the header files of MS MPI: <Installation Directory CCS SDK>\Include,



• The library file with the realization of MPI functions. Choose the menu option Project->Project Properties. In the option Configuration Properties->C++->Linker->Input->Additional Dependencies enter the name of the library file msmpi.lib,



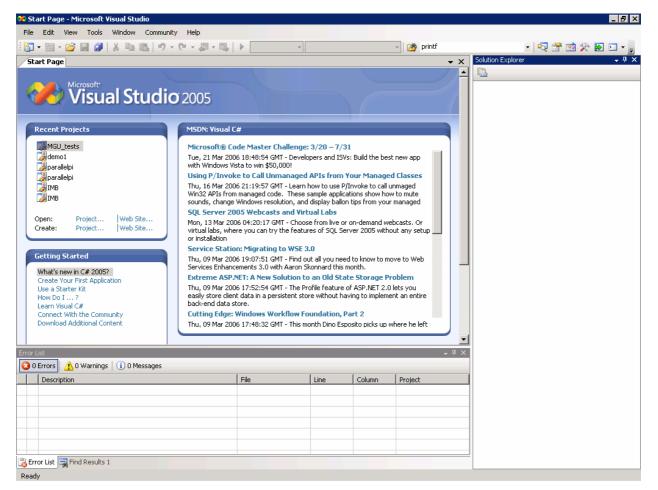
• The path to the library file msmpi.lib. Choose the menu option Project->Project Properties. In the option Configuration Properties->C++->Linker->General->Additional Library Directories enter the path to the library file msmpi.lib: <Installation Directory CCS SDK>\Lib\AMD64 depending on the processor architecture you are using,



Task 3 – Compiling a Parallel Program in Microsoft Visual Studio 2005

As an example of the parallel program for this task, we will use the parallel algorithm of computing the Pi. In this work we describe only the technical aspects of using Microsoft Compute Cluster Server 2003; the description of the algorithm and the aspects of its implementation are described in Lab "Parallel Programming using MPI". In this task we will consider only the aspects of using Visual Studio 2005 for compiling a parallel MPI program to be used in the environment MS MPI:

• Run Microsoft Visual Studio 2005,



• Create a new project: choose the menu option File->New->Project. In the window, where you choose the new project, choose the console Win32 application (Other Languages->Visual C++->Win32->Win32 Console Application), enter the project name in the field Name (for instance, "parallelpi") and make sure that the path to the project if chosen correctly (the field Location). Press the button OK to choose the rest settings of the project being created,

🚧 Start Pa	age - Microsoft Visu	al Studio	_ 8 >
File Edit	V <mark>iew To</mark> ols Wi	indow Commu <mark>n</mark> ity Help	
i 🎦 🗸 🖮	New Project		? ×
Start Pi	Project types:	Templates:	<u>₽</u> ₽ <u>₽</u> ₽ <u>₽</u> ₽ <u>₽</u> ₽ <u>₽</u> ₽ <u>₽</u> <u>₽</u> <u>₽</u> ~ 7 ×
Rece P P P P P P P P P P P P P P P P P P	 → Visual C# → Windows → Smart Devi → Database → Other Languag ⊕ Visual Basic ⊕ Visual J# ⊕ Visual C++ ↓ Visual C++ ↓ Wisual C++ ↓ Wisu	es Search Online Templates	
Error List	A project for creati	ng a Win32 console application	
CO Error	Name:	parallelpi	
Des	Location:	D:\Projects\senin\mpi_test\Release	▼ Browse
	Solution Name:	parallelpi 🔽 Create directory for solution	
		Add to Source Control	
		ОК	Cancel
Error List	Find Results 1		
🏄 Start	🗁 i386	🧭 Start Page - Microsof	🛛 🖪 < 🏂 🗐 📕 0:52

• Press the button **Next** in the new window,

Win32 Application Wizard - PI_Ca	alculation	? ×
Welcome	to the Win32 Application Wizard	
Overview Application Settings	These are the current project settings: • Console application Click Finish from any window to accept the current settings. After you create the project, see the project's readme.txt file for information about the project features and files that are generated. Previous Next > Finish Cancel	

• Choose the project settings in the new window (you can accept all the default settings). Press the button **Finish**,

Overview Application type: Add common header files for: Application Settings	Win32 Application Wizard - PI_C	alculation		? ×
Application Settings Windows application Console application MFC DLL Static library Additional options: Empty project Empty project Export symbols	Applicatio	on Settings		
< Previous Next > Finish Cancel		 Windows application Console application DLL Static library Additional options: Empty project Export symbols Precompiled header 	□ <u>A</u> TL □ <u>M</u> FC	

• In the window **Solution Explorer** double click on the file **parallelpi.cpp** (the file name coincides with the project name you have entered)

👀 parallelpi - Microsoft ¥isual Studi	o					_ 8 ×
File Edit V <mark>iew Pr</mark> oject Build	Debug <mark>T</mark> ools Windov	v Community	Help			
🛅 • 🛅 • 💕 📕 🥥 👗 🗈 I	11. ち・ロ・目・	🖳 🕨 Deb	ug 👻 Win32		• 🖄	
💷 🗞 🖕 💀 🛊 🛊 🗏 😫		æ 🔒 🔍	-			
parallelpi.cpp Start Page			<i></i>	• X	Solution Explorer - Solutio $~-$	+ × 🙀
(Global Scope)	•			•		Prop
<pre> // parallelpi.cpp : D // #include "stdafx.h" int _tmain(int argc,</pre>	_TCH&R* argv[])	point for	r the console	applice	Solution 'parallelpi' (1 projec Parallelpi Parallelpi Parallelpi Parallelpi Parallelpi Parallelpi Parallelpi Parallelpi.cpp Parallelpi.cpp Parallelpi.cpp Parallelpi.cpp Parallelpi.cpp Parallelpi.cpp Parallelpi.cpp Parallelpi.cpp Parallelpi.cpp Parallelpi.cpp	t)
Description		Line C	olumn Project			
Error List Results 1				n 3 Co		TAIC
Ready		1				INS
🏄 Start 🛛 🗁 i386	C Release		💋 parallelpi - Micr	osoft	🛛 🖪 🛛 🛃 🔤 K	0:57

• Delete the file content and replace it with the following code (see Lab "Parallel Programming using MPI"):

```
#include "stdafx.h"
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <mpi.h>
void main(int argc, char *argv[]) {
       NumIntervals = 0; // num intervals in the domain [0,1]
  int
  double IntervalWidth = 0.0; // width of intervals
  double IntervalLength = 0.0; // length of intervals
  double IntrvlMidPoint = 0.0; // x mid point of interval
                         = 0; // loop counter
  int
         Interval
                         = 0; // flag
  int
        done
  double MvPI
                         = 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;
       MyID;
  int
  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)</pre>
      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){</pre>
     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();
}
```

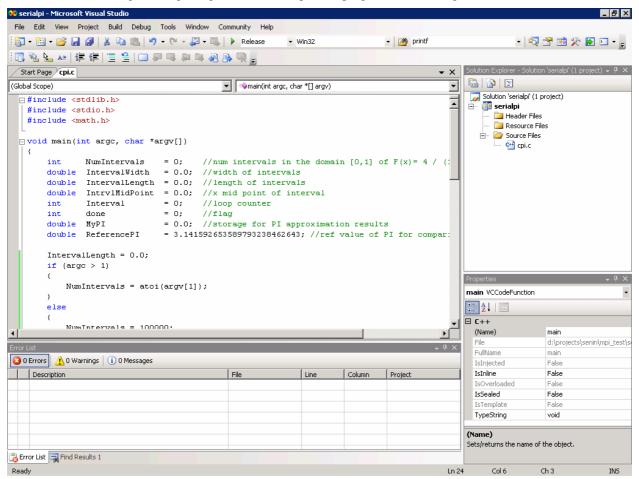
- Make settings of the project Visual Studio 2005 for compiling the MPI part of the project in accordance with the instructions given in "Setting the development integration environment of Microsoft Visual Studio 2005",
- Execute the command Build->Rebuild Solution for compiling and linking the project,
- Congratulations! The compilation of the program MS MPI is successfully completed.

Exercise 2 – Running a Serial Task

A task is considered to be serial if it uses the resources of only one processor for its execution. Compiling a serial program (and also compiling a parallel program with the use of the technology OpenMP) for using it on the cluster under Compute Cluster Server 2003 does not differ from the standard one and does not require using additional libraries. In the given exercise we will consider all required steps to run a serial task on the cluster.

Launching the Program via Graphic User Interface

• Open the project of the serial program for calculating the Pi (**serialpi**), which appears together with Lab "Parallel Programming using MPI", and compile the program in the configuration **Release**,



• Open Computer Cluster Job Manager (Start->All Programs->Microsoft Compute Cluster Pack->Compute Cluster Job Manager) to run the program on the cluster. If you have installed the client part Compute Cluster Pack on your PC, you can queue the tasks immediately from your computer, otherwise you should go to the head cluster node or any other node, where the client part is installed, via Remote Desktop Connection,

		at s-cw		1						_
		Help	Show:	All Jobs	-					
	Nar				Priority	Submitted By	Status	Submit Time	Pending Reason	
		tname			Normal	CCAM\Senin	Finished	21.06.2006 4:10:14		
	imb				Normal	CCAM\Senin	Failed	21.06.2006 4:19:50		
}	imb				Normal	CCAM\Senin	Failed	21.06.2006 4:24:24		
2 } 	imb				Normal	CCAM\Senin	Failed	21.06.2006 4:25:23		
	imb				Normal	CCAM\Senin	Finished	21.06.2006 5:13:20		
cta	jobto	viewits	stasks							
	job to	view its	s tasks Status	Task Id	Command Line	Processors	End Time	Failure Message		
	i job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
	job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
	i job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
	job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
	job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
	i job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
	i job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		_
	i job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
	job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
	i job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
ecta ne	job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		_
	job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
	job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		
	job to	view its		Task Id	Command Line	Processors	End Time	Failure Message		

- In the open window of the job manager choose the menu option **File->Submit Job** in order to enqueue the job,
- In the window of job enqueuing enter the name of the job (field **Job Name**), if it is necessary, change the priority of the job (the user's jobs of high priority will be executed prior to the jobs of lower priority). Go to the tab **Processors**,

ubmit Job Serial Pi	computing			×
General Processors	Tasks Licenses	Advanced		
🟐 Serial Pi con	mputing			
Job Name:	Serial Pi compu	ting		
Project Name:				
<u>P</u> riority:	Normal	•		
Submitted By:	N/A			
Submitted on:	N/A			
Status:	Not Submitted			
<u>S</u> ave As Templa	ite		Submit	Cancel

• In the tab **Processors** enter the maximum and the minimum number of the processors required for the jobs (in our case the maximum required number of processors is one, as the task is serial). We assume the number of processors to be maximum if it is optimum for this job (this number of processes will be allocated in case of low cluster load). It is guaranteed that the job will not be started, if the number of available processors on the cluster is less than the minimum number. Additionally you can enter the estimated run time of the job (it will help the job scheduler to distribute the system resources more efficiently) – the window panel **Estimate run time for this job**. If you want the computational resources to be reserved for the job during some specified period of time even after all the tasks have finished their execution, then tick the option **Run job until end of run time or until canceled**. Thus, you will be able to start new tasks of the job even after all the originally given tasks have been completed. Go to the window tab **Tasks** in order to add new tasks,

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 →	
Bun 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.	
Submit Cancel	

• Enter the task name (the field **Task Name**) and the command for executing (the field **Command Line**) – the program name and the command line parameters. The program should be located on the shared network location, which is accessible from all cluster nodes. Press the button **Add** to add a new task to the job,

Submit Job Serial Pi Calculatio	חכ		×
General Processors Tasks	Licenses Advanced		
⊤ Task Command Line			
	erial Pi		
	s-cw-head\temp\serialpi.exe	1000	Add
		TUUQ	
Minimum Processors:			
Mininum Pi <u>o</u> cessors.	1 🗧 Maximum <u>P</u> ro	icessors.	
This job contains the following	tasks:		
Order Command Line		essors	<u>R</u> emove
			Edit
		_	
	Add Parametric Swe	ер <u>w</u>	'hat is it?
Task Summary			
<u>S</u> ave As Template		Submit	Cancel

• The task, which was added, will appear in the task list of the current job (the list **This job contains the following tasks**). Select it in the list and press the button **Edit** in order to edit the additional task parameters,

ubmit Job Serial Pi Calculatio	n		×
General Processors Tasks I	icenses Advanced		
Task Command Line	Task	Add	
This job contains the following t Order Command Line 1 \\s-cw-head\tem	asks: Proces p\serialpi.exe 1000 1	sors <u>R</u> emove	
		<u>E</u> dit	
	Add Parametric Sweep	<u>What is it?</u>	
Task Summary Name :Serial Pi Command Line :\\s-cw-head Standard Input : Standard Output : Standard Error : Work Directory : Number of processors reques RunTime :Infinite Preceding tasks(dependent to Exclusive :False	ted :1		•
<u>S</u> ave As Template		Submit Can	cel

• In the new window enter the file, where you are going to redirect the standard output stream of the console application (the field **Standard Output**). Besides, you may specify the file of the standard input stream (the field **Standard Input**), the file of the standard error stream (the field **Standard Error**), the work directory of the program being executed (the field **Work Directory**) and the time limits on the duration of the task run time (the total run time must not exceed the estimations of the task run time) – the field **Limit task run time to**. Choose the window tab **Processors**,

Task Properties						×
Tasks Processor	rs Tasks Dep	endencies	Environme	nt Adv	/anced	
Select task to vi	ew settings:					
Name	Command Lir			Runtin		
Serial Pi	\\s-cw-head\	\temp\serialp	i.exe 10	unspe	cified	
, Task Comman	d Line Propertie	es ———				
Task <u>N</u> ame:		Serial Pi				
<u>C</u> ommand Lin	e:	Ns-cw-head	l\temp\seria	alpi.exe	1000	
Standard <u>I</u> npu	ut:					
Standard <u>O</u> ut	put:	Ws-cw-head	l\temp\seria	alpi.txt		
Standard <u>E</u> rro	ir:					
Work Directo	ry:					
Job run time:	unspecified					
<u>L</u> imit task						
<u>D</u> ays:		<u>H</u> ours:	1		<u>M</u> inutes:	0 🚠
				_		
			OK		Cancel	Apply

• In the window tab **Processors** in the upper list (**Select task to view settings**) select the task, where you want to change the settings, and specify the minimum and the maximum number of processors for the selected task (the fields **Min. required** and **Max. required**) if you want the job scheduler to select the nodes automatically (**Use any available processors on any nodes**). If you want to select the nodes manually, choose the option **Select nodes required for this task** and tick the required nodes in the lower list. As the task is serial, it requires only one processor. This is the end of setting the task parameters. Press **OK** to save the changes to be done and return to the job setting,

Task Pro	perties					×
Tasks	Processors	Tasks Depende	encies Environn	nent Advanced	1	
Selec	t task to vie	w settings:				
Nan	ne	Command Line		Processors		
Seri	al Pi	\\s-cw-head\tem	p∖serialpi.exe 10	. 1		
, ⊏Tas	k Command	Line Processors				
		vailable processor:	s on any nodes			
	Available:	1	s on any nodes.			
	<u>M</u> in. require		1 🗧 Ma	x. required:	1 🛨	1
		es required for this			<u> </u>	'
	Jelect N <u>o</u> u	es required for this	Idan.			
	Name		Processors	Speed	RAM	[]
			0	K Ca	ancel <u>A</u> pp	oly

• Go to the window tab **Advanced** and choose the option **Use any available nodes** to choose the nodes for the job automatically. If you want to select the nodes for executing the job manually, choose the option **Use only these nodes**. Remember that if you have selected the nodes manually, these nodes must be also selected for the whole job. Tick **Use the allocated nodes exclusively for this job** in order to prohibit executing several jobs on the same node. Press the button **Submit** to add the job to the queue,

Submit Jo	ob Serial Pi computing				×					
General	Processors Tasks Lic	enses Advanced								
		······/								
	Nodes Specify the compute nodes to use for this job in your cluster									
	 Specify the compute nodes to use for this job in your cluster Use any available nodes. 									
	 Use any available nodes. O Use only these nodes: 									
	Name	Processors	Speed	RAM						
		. 10000010	0,000							
	e the allocated nodes <u>e</u> xclu	usively for this job			1					
	e the allocated hodes <u>e</u> xcli lect this option if your job v		ed due to other iob	s usina the						
	ne nodes.			e aong are						
Sav	ve As Template		Submit	Cancel						
	in the supervision		- O GOTIN							

• Enter the name and the password of the user, who is authorized to run tasks on the cluster, and press OK,

Connect to S-CW-H	IEAD ? 🗙
	GA
Welcome back to S-	CW-HEAD
User name:	🖸 CCAM\senin 💽
Password:	
	Remember my password
	OK Cancel

• The job will appear in the queue. After the execution of the job its status will change to **Finished**.

⊻iev) Job	Help	Show:	All Jobs	-					
	Name				Priority	Submitted By	Status	Submit Time	Pending Reason	
	hostname	,			Normal	CCAM\Senin	Finished	21.06.2006 4:10:14		
	imb				Normal	CCAM\Senin	Failed	21.06.2006 4:19:50		
	imb				Normal	CCAM\Senin	Failed	21.06.2006 4:24:24		
	imb				Normal	CCAM\Senin	Failed	21.06.2006 4:25:23		
	imb				Normal	CCAM\Senin	Finished	21.06.2006 5:13:20		
		computing			Normal	CCAM\Senin	Failed	25.06.2006 1:59:22		
		computing			Normal	CCAM\Senin	Finished	25.06.2006 2:00:46		
		computing			Normal	CCAM\Senin	Finished	25.06.2006 3:50:22		
	Serial Pi (Calculation			Normal	CCAM\Senin	Finished	25.06.2006 4:08:39		
	erial Pi C	alculation		Task Id	Compand Line	Prosec	Transfer Time	E silura Manazan		
	erial Pi C	alculation Status Finished		Task Id 1	Command Line	Process	ors End Time 25.06.2006	Failure Message		
s for So errial Pi	erial Pi C	Status								

• In the file specified in the job setting for the redirection of the standard output stream, you can find the results of the program execution.

📕 serialpi.txt - Notepad 📃	
<u>File E</u> dit F <u>o</u> rmat <u>V</u> iew <u>H</u> elp	
NumIntervals = 1000 PI is approximately 3.1415927369231227, Error is 0.0000000833333296	4
<u> </u>	▶ //

Launching the Program by Means of Template

If you want to launch the task of the serial calculation of the Pi once again (for instance, with other parameters), it will be useful for you to use the command storing all the parameters of the previously launched job in **xml**-file with the possibility to quickly create a copy:

• Open Computer Cluster Job Manager (Start->All Programs->Microsoft Compute Cluster Pack->Compute Cluster Job Manager) and double click on the job, the parameters of which you want to save in the xml-file,

	jew <u>J</u>		-head Help	Chour	All Jobs	•					
	_		Help	Show:	All Jobs		1	1.	1	1	
	Name					Priority	Submitted By	Status	Submit Time	Pending Reason	
	hostn	ame				Normal	CCAM\Senin	Finished	21.06.2006 4:10:14		
	imb					Normal	CCAM\Senin	Failed	21.06.2006 4:19:50		
	imb					Normal	CCAM\Senin	Failed	21.06.2006 4:24:24		
	imb					Normal	CCAM\Senin	Failed	21.06.2006 4:25:23		
	imb	-				Normal	CCAM\Senin	Finished	21.06.2006 5:13:20		
			mputing			Normal	CCAM\Senin CCAM\Senin	Failed Finished	25.06.2006 1:59:22		
			mputing			Normal			25.06.2006 2:00:46		
			mputing			Normal	CCAM\Senin	Finished	25.06.2006 3:50:22		
	Serial	FILa	lculation			Normal	CCAM\Senin	Finished	25.06.2006 4:08:39		
for	Serial F	PiCalo	culation								
			Status		Task Id	Command Line	Processors	End Time	Failure Message		
					Task Id 1	Command Line		End Time 25.06.2006 .			
rial F			Status								

• Press the button Save As Template in the window to save the job parameters in the file,

Job Serial Pi Calculatio	on Properties		×							
General Processors	Tasks Licenses Advanced									
Serial Pi Calc	Serial Pi Calculation									
Job Name:	Serial Pi Calculation									
P <u>r</u> oject Name:										
<u>P</u> riority:	Normal									
Submitted By:	CCAM\Senin									
Submitted on:	25.06.2006 4:08:39									
Status:	Finished									
<u>S</u> ave As Template	÷	OK. Cano	:el							

• In the open window choose the directory, where you should save the file, and enter its name. Press the button **Save** to save the job in the file,

Save as Templat	e					? ×
Save jn:	🕒 My Document	\$	•	6	D 📂 🖪	-
My Recent Documents Desktop My Documents My Computer	ICQ Lite My eBooks My Music My Pictures Visual Studio 20 Demo.xml Hostname.xml)05				
My Network Places	File <u>n</u> ame:	SerialPi.xml			•	<u>S</u> ave
	Save as <u>type</u> :	Template files (*.xml)			•	Cancel

• In order to create the job using the template in the window **Compute Cluster Job Manager**, choose the menu option **File->Submit Job with Template**... In the window for choosing the template, select the file, where you saved the job at the previous step, and press the button **Open**,

Submit Job with	Template				? ×
Look jn:	🕒 My Document	8	• 0	🦻 📂 🛄•	
My Recent Documents Desktop My Documents My Computer	☐ ICQ Lite ☐ My eBooks My Music ☐ My Pictures ☐ Visual Studio 20 20 20 ParallelPi.xml 20 20 20 20 20 20 20 20 20 20)05			
My Network	File <u>n</u> ame:	SerialPi.xml		•	<u>O</u> pen
Places	Files of type:	Template files (*.xml)		•	Cancel

• You will see the window for adding the job to the queue. The job parameters and the parameters of all its tasks will be identical to those of the job, which was the basis for creating the template. You can change any required job parameters leaving the rest of them unchanged. Thus, you will save time for editing. For instance, you can increase the number of partitioning the integration interval for calculating the Pi

described in the previous example (the algorithm, which was used for calculating the Pi, is reduced to the numerical calculation of a definite integral), you can leave the rest of the parameters unchanged.

k Properties							I
asks Processor	s Tasks Dep	pendencies	Environme	nt Ad	vanced		
Select task to vie	ew settings:						
Name	Command Li			Runti			
Serial Pi	 \\s-cw-head 	l\temp\serial;	oi.exe 10	unspe	cified		
- Task Command	d Line Properti	ies ——					
Task <u>N</u> ame:	·	Serial Pi					
<u>C</u> ommand Line	e:	Ns-cw-hea	d\temp\seri	alpi.exe	2000		
Standard <u>I</u> npu	ut:						
Standard <u>O</u> utp	put:	Ns-cw-hea	d\temp\seri	alpi.txt			_
Standard <u>E</u> rro	r:						_
Work Director	(y:						_
Job run time: I	-	s 1 minutes					
<u>L</u> imit task r <u>D</u> ays:	run time to:	Hours:			Minutes:		1 -
<u>D</u> ays.		<u>H</u> ours.			<u>Minutes.</u>	1	

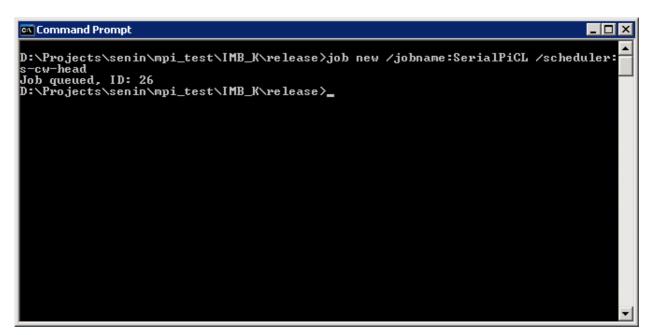
Launching the Program from the Command Line

It is often more convenient to control the course of executing jobs from the command line. Microsoft Compute Cluster Server 2003 includes the utilities, which provide full control over the course of executing jobs on the cluster.

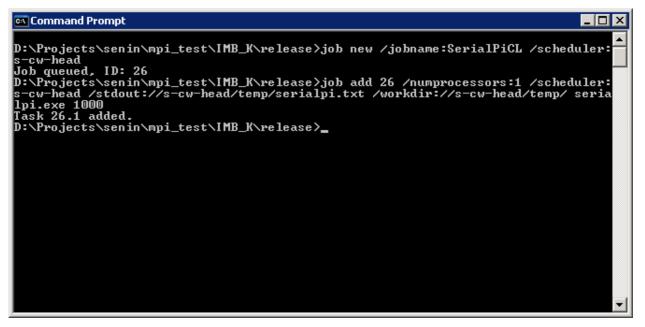
This Lab will illustrate the launch of a serial task from the command line. The launch of a parallel program and the creation of the parametric sweep and a work flow may be also executed from the command line. Additional information on the commands and their parameters is available in the documentation supplied with Microsoft Compute Cluster Pack.

In order to start the serial program of calculating the Pi, you should do the following:

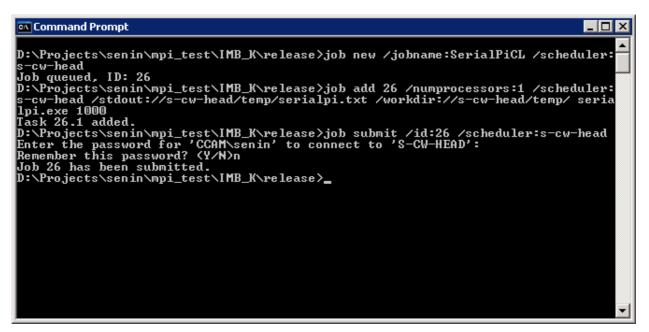
- Open the command window (Start->Run, enter the command cmd and press Enter),
- To create a new job enter the command "job new /jobname:SerialPiCL /scheduler:s-cw-head" (do not forget to change the command parameters for the ones, which correspond to your case), where the parameter "jobname" is the name of the job being added, "scheduler" is the name of the head cluster node. The command will print the identifier (id) of the created job. Further on you will work with this id,



• In order to add a new task to the job enter the command "job add 26 /numprocessors:1 /scheduler:s-cw-head /stdout://s-cw-head/temp/serialpi.txt /workdir://s-cw-head/temp/ serialpi.exe 1000" (do not forget to change the command parameters for the ones, which correspond to your case). Here the number "26" is the job id printed at the previous step. The parameter "numprocessors" sets the number of the processors required for the task (to set the minimum and the maximum number of processors you should use the format "/numprocessors:x-y", where x is the minimum number of processors, and y is the maximum number of processors). The parameter "stdout" sets the file, where the standard output stream will be redirected. The parameter "workdir" sets the directory on default for the application to be launched. After the parameters you will specify the command to launch the application and the arguments of the command line,



• To start scheduling the job enter the command "**job submit** /**id:26** /**scheduler:s-cw-head**" (do not forget to change the command parameters for the ones, which correspond to your case). Enter the user's password, which you used to login in the system. If requested whether to store your password in order not to enter it further, enter "n" to refuse,



• Your task has been added to the queue, and the scheduler has begun to plan its launch. You can track the status in the program **Job Manager** or by entering the command "**job list /scheduler:s-cw-head /all**" (do not forget to change the command parameters for the ones, which correspond to your case).

Exercise 3 – Launching a Parallel Job

In Task 3 of Exercise 1 we compiled the parallel program for calculating the value of Pi for MS MPI. Let us launch it on the cluster under Microsoft Compute Cluster Server 2003:

• Open Computer Cluster Job Manager (Start->All Programs->Microsoft Compute Cluster Pack->Compute Cluster Job Manager) to start the program on the cluster,

e View		Show: All J	obs	-						
	Name			Priority	Submitter		Status	Submit Time	Pending Reason	
	hostname			Normal	CCAM\S		Finished	21.06.2006 4:10:14		
2	imb			Normal	CCAM\S		Failed	21.06.2006 4:19:50		
}	imb			Normal	CCAM\S		Failed	21.06.2006 4:24:24		
	imb			Normal	CCAM\S		Failed	21.06.2006 4:25:23		
i	imb			Normal	CCAM\S		Finished	21.06.2006 5:13:20		
5	Serial Pi compu	uting		Normal	CCAM\S	enin	Failed	25.06.2006 1:59:22		
,	Serial Pi compu	uting		Normal	CCAM\S	enin	Finished	25.06.2006 2:00:46		
}	Serial Pi compu	uting		Normal	CCAM\S	enin	Finished	25.06.2006 3:50:22		
1	Serial Pi Calcul	ation		Normal	CCAM\S	enin	Finished	25.06.2006 4:08:39		
ect a job	b to v ie w its ta	sks								
-		sks atus	Task Id	Command Line		Processors	End Time	Failure Message		
-			Task Id	Command Line	[Processors	End Time	Failure Message		
-			Task Id	Command Line	(Processors	End Time	Failure Message		
-			Task Id	Command Line	[Processors	End Time	Failure Message		
-			Task Id	Command Line		Processors	End Time	Failure Message		
-			Task Id	Command Line		Processors	End Time	Failure Message		
-			Task Id	Command Line		Processors	End Time	Failure Message		
-			Task Id	Command Line		Processors	End Time	Failure Message		
-			Task Id	Command Line		Processors	End Time	Failure Message		
-			Task Id	Command Line		Processors	End Time	Failure Message		
-			Task Id	Command Line		Processors	End Time	Failure Message		
-			Task Id	Command Line		Processors	End Time	Failure Message		
-			Task Id	Command Line	1	Processors	End Time	Failure Message		
-			Task Id	Command Line		Processors	End Time	Failure Message		
ect a job ne			Task Id	Command Line		Processors	End Time	Failure Message		
-			Task Id	Command Line		Processors	End Time	Failure Message		

- In the window of the job manager choose the menu option **File->Submit Job** in order to add a new job to the queue,
- In the window of adding the job to the queue enter the job name (the field **Job Name**). Go to the window tab **Processors**,

omit Job Parallel	Pi computing						
ieneral Processors Tasks Licenses Advanced							
Parallel Pi o	computing						
Job Name:	Parallel Pi computing						
Project Name:							
<u>P</u> riority:	Normal						
Submitted By:	N/A						
Submitted on:	N/A						
Status:	Not Submitted						
<u>S</u> ave As Templ	ate Submit Cancel						

• In the window tab **Processors** enter the minimum and the maximum number of processors required for executing the job (for instance, 10 and 20 correspondingly). Go to the window tab **Tasks** to add new tasks to the job,

Submit Job Parallel Pi comput	ing		×
General Processors Tasks	Licenses Advanced		
Processors required for this jo Processors available in this <u>M</u> inimum required: M <u>a</u> ximum required:			
<u>Estimate run time for this</u>	s job		
Days: 0	Hours: 1 🛒	Mi <u>n</u> utes:	0.#
Bun job until end of run tim This option lets you run extra t there is time left.		already listed in th	ie job if
<u>S</u> ave As Template		Submit	Cancel

• Add the task name (the field **Task Name**) and the command to be executed (the field **Command Line**). Launching the tasks developed for MS MPI must be executed with the use of the special utility **mpiexec.exe**, which accepts the name of the parallel program, the list of nodes, where the launch will be executed, and the parameters of the program being launched, as its parameters. The list of nodes is set by the parameter "-hosts". If the nodes have been allocated automatically by the scheduler, the list of nodes will be contained in the environment variable **CCP_NODES**. The value of the variable should be given to the utility as a parameter. The example of the command for launching the parallel program is "**mpiexec.exe** -hosts %CCP_NODES% \\s-cw-head\temp\parallelpi.exe". Press the button Add to add the task to the job,

iubmit Job Parallel Pi computing 🛛 🛛 🗙
General Processors Tasks Licenses Advanced
Task Command Line Iask Name: Parallel Pi Command Line: P_NODES% \\s-cw-head\temp\parallelpi.exe Image: Use job's allocated processors Minimum Processors: 10 Maximum Processors: 20
This job contains the following tasks:
Order Command Line Processors <u>R</u> emove
Add Parametric Sweep What is it?
Task Summary Name :Parallel Pi Command Line :mpiexec.exe -hosts %CCP_NODES% \\s-cw-head\temp \parallelpi.exe Standard Input : Standard Output : Standard Error : Work Directory : Number of processors requested :10 - 20 BunTime :Infinite Preceding tasks(dependent tasks):
Submit Cancel

• The added task will appear in the task list of the current job (the list **This job contains the following tasks**). Select it in the list and press the button **Edit** to edit the additional task parameters,

ubmit Job Parallel Pi comput	ing	×
General Processors Tasks	Licenses Advanced	
		[
Task Command Line	T 1	_
	• Task	
Command Line:		Add
✓ Use job's allocated proce	essors	
Minimum Pr <u>o</u> cessors:	10 Maximum Processors:	20 ≑
This job contains the following	tasks:	
Order Command Line	Processors	<u>R</u> emove
1 mpiexec.exe -ho:	sts %CCP_NODES 10 - 20	Edit
1	Add Deservable Course	1
	Add Parametric Sweep	<u>What is it?</u>
Task Summary		
Name :Parallel Pi Command Line :mpiexec.exe	+hosts %CCP_NODES% \\s-cw-head\t	emp 🔺 📗
\parallelpi.exe Standard Input :		
Standard Output :		
Standard Error : Work Directory :		
Number of processors reque RunTime :Infinite	sted :10 - 20	
Preceding tasks(dependent	tasks) :	- I
<u>S</u> ave As Template	Subr	nit Cancel

• In the new window enter the path to the file, where the standard output stream of the console application will be redirected (the field **Standard Output**). Choose the window tab **Processors**,

Task Properti	es						×
Tasks Proc	essors Tasks De	pendencies	Environmer	nt Adva	anced		
Select task	to view settings:			·			
Name	Command L	ine		Buntime	e		
Parallel Pi	mpiexec.exe	e -hosts %CCF	P_NODE	unspec	ified		
Taak Com	mand Line Proper	liaa					
Task <u>N</u> a		Parallel Pi					_
<u>C</u> omman			e -hosts %C(EC% Marca	u bood\te	
		Implexec.ex		LF_NOD	23% \\\$*0	w-neau ((e	
Standard							_
Standard		\\s-cw-hea	d\temp\para	allelpi.txt			_
Standard	l <u>E</u> rror:						
<u>W</u> ork Dir	rectory:						
Job run t	ime: unspecified						
🔲 Limit	task run time to:						
<u>D</u> ays:		<u>H</u> ours:	1	<u>-</u>	<u>M</u> inutes:		0 🕀
			OK		Cancel		Apply

• In the window tab **Processors** in the upper list (**Select task to view settings**) select the task, where you want to change the settings, and give the maximum and the minimum number of processors for the selected task (the fields **Min. required** and **Max. required**). Press **OK** to save the changes to be done and return to the job settings,

ask Properties 🔀
Tasks Processors Tasks Dependencies Environment Advanced
Select task to view settings:
Name Command Line Processors
Parallel Pi mpiexec.exe -hosts %CCP_NODE 10 - 20
Task Command Line Processors
<u>U</u> se any available processors on any nodes. Available: 10 - 20
C Select nodes required for this task:
Name Processors Speed RAM
OK Cancel <u>A</u> pply

• Press the button **Submit** to add the job to the queue. In the window requesting for the password, enter the name and the password of the user, who is authorized to launch tasks on the cluster, and press **OK**. The job will appear in the queue **Job Manager**. After the job execution is completed, its status will change for **Finished**. The file, given in the task settings for redirection of the standard output stream, contains the results of the program execution,

📮 parallelpi.txt - Notepad	
<u>File E</u> dit F <u>o</u> rmat <u>V</u> iew <u>H</u> elp	
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 0.000000000083329	-

Exercise 4 – Launching a Parametric Sweep

Here we will consider the launch of parametric sweep within a job. The parametric sweep is a series of launches of the same program with different parameters. As an example you can run a series of several hundreds experiments on computing the Pi in order to study the rate of the method convergence to the solution. As an example of the program for this Exercise we will use the program of parallel computation of the Pi:

• Open Computer Cluster Job Manager (Start->All Programs->Microsoft Compute Cluster Pack->Compute Cluster Job Manager) to create the parametric sweep,

		-	1		
Name	Priority	Submitted By	Status	Submit Time	Pending Reason
hostname	Normal	CCAM\Senin	Finished	21.06.2006 4:10:14	
imb	Normal	CCAM\Senin	Failed	21.06.2006 4:19:50	
imb	Normal	CCAM\Senin	Failed	21.06.2006 4:24:24	
imb	Normal	CCAM\Senin	Failed	21.06.2006 4:25:23	
imb	Normal	CCAM\Senin	Finished	21.06.2006 5:13:20	
Serial Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 1:59:22	
Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 2:00:46	
Serial Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 3:50:22	
Serial Pi Calculation	Normal	CCAM\Senin	Finished	25.06.2006 4:08:39	
Parallel Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 22:13:00	
Parallel Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 22:15:36	
Parallel Pi computing	Normal	CCAM\Senin	Failed	25.06.2006 22:16:49	
Parallel Pi computing	Normal	CCAM\Senin	Finished	25.06.2006 22:17:33	
Parallel Pi computing	Normal	CCAM\Senin	Finished	26.06.2006 0:29:12	
Parallel Pi computing	Normal	CCAM\Senin	Finished	26.06.2006 0:31:30	
t a job to view its tasks					
Status Task Id	Command Line	Processors	End Time	Failure Message	

- In the window of the job manager choose the menu option File->Submit Job to add a new job to the queue,
- In the window of adding the job to the queue enter the job name (the field **Job Name**). Go to the window tab **Processors**,

ubmit Job Parallel Pi	i parametric swee	≥p		×
General Processors	Tasks Licenses	Advanced		
🞒 Parallel Pi pa	rametric sweep			
Job Name:	Parallel Pi param	etric sweep		
P <u>r</u> oject Name:				
<u>P</u> riority:	Normal	•		
Submitted By:	N/A			
Submitted on:	N/A			
Status:	Not Submitted			
<u>S</u> ave As Templati	e		Submit	Cancel

• In the window tab **Processors** enter the minimum and the maximum number of processors required for executing the job (for instance, 10 and 20 correspondingly). Go to the window tab **Tasks** and press the button **Add parametric Sweep** in the window tab to add new tasks to the job,

Submit Job Parallel Pi param	etric swee	р			×
General Processors Tasks	Licenses	Advanced			
Processors required for this Processors available in thi <u>M</u> inimum required: M <u>a</u> ximum required:		60 10 ÷ 20 ÷			
<u>E</u> stimate run time for th	nis job —				
Days: 0 👘	<u>H</u> ours:	1 📩	Mi <u>n</u> utes:	0	3
This option lets you run extra there is time left.	a tasks after	running all task	s already liste	d in the jol	b if
<u>S</u> ave As Template			Submit		Cancel

In the window of adding the parametric sweep enter the name assigned to each new task (the field Name). Enter the command for the task using the asterisk (the symbol "*") as the argument parameter of the command line. The symbol "*" for each particular command will be replaced by an integer number, the range of change for the number will be specified in the fields Index Start and Index End. The index for our task (the number of intervals of the numerical integration) may change, for instance, from 50 to 100. Thus, the command may be the following: "mpiexec.exe –hosts %CCP_NODES% \\s-cw-head\temp\parallelpi.exe *". Specify the files, where the standard output stream will be redirected, using "*" as a parameter. For instance: "\\s-cw-head\temp\parallelpi*.txt". Press OK to add a task sweep to the job,

d Parametric Sweep
Create Task
Name:
Parallel Pi
<u>C</u> ommand Line (Use * to represent index if desired): NODES% \\s-cw-head\temp\parallelpi.exe *
Index <u>S</u> tart: 50 🛨 Index <u>E</u> nd: 100 🐳 Index S <u>kip</u> : 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
Assian Work Directory
Work Directory
Preview Task
Command Line Standard Outpul 🔺
Parallel Pi
mpiexec -hosts %CCP_NODES% \\s-cw-head\temp\parallelpi.exe 50 \\s-cw-head\ter mpiexec -hosts %CCP_NODES% \\s-cw-head\temp\parallelpi.exe 51 \\s-cw-head\ter mpiexec -hosts %CCP_NODES% \\s-cw-head\temp\parallelpi.exe 52 \\s-cw-head\ter
Extension Length: 1 🐳
Add Cancel

• In the window of the job setting select all the tasks contained in the job (in order to select several tasks, use the key **Shift**), press the button **Edit** to specify the number of the processors required for the tasks. In the new window go to the window tab **Processors**, select the option **Use any available processors on any nodes** and specify, for instance, 10 as the minimum number of processors, and 20 as the maximum one. Press **OK**. In the open window press the button **Submit** to add the job to the queue,

ask Properties			×
Tasks Processor	s Tasks Dependencies Environme	nt Advanced	
Select task to vie	ew settings:		
Name	Command Line	Processors	
Parallel Pi	mpiexec -hosts %CCP_NODES%	10 - 20	
Parallel Pi Parallel Pi	mpiexec -hosts %CCP_NODES% mpiexec -hosts %CCP_NODES%	10-20 10-20	
Parallel Pi	mpiexec -hosts %CCP_NODES%	10-20	
Parallel Pi	mpiexec -hosts %CCP_NODES%	10-20	
Parallel Pi	mpiexec -hosts %CCP_NODES%	10 - 20	
Parallel Pi	mpiexec -hosts %CCP_NODES%	10-20	
Parallel Pi Parallel Pi	mpiexec -hosts %CCP_NODES% mpiexec -hosts %CCP_NODES%	10-20 10-20	
Parallel Pi	mpiexec -hosts %CCP_NODES%	10-20	
Parallel Pi	mpiexec -hosts %CCP_NODES%	10-20	_
		40.00	
Available: <u>M</u> in. require	available processors on any nodes. 10 - 20 ed: <u>10 - Max</u> . des required for this task:	required:	20 🛨
Name	Processors 9	Speed RAM	
	ОК	Cancel	Apply

• Enter the name and the password of the user authorized to run tasks on the cluster and press OK,

Connect to S-CW-	HEAD ? 🗙
R	G
Welcome back to S	-CW-HEAD
<u>U</u> ser name:	😴 CCAM\senin 💌
Password:	
	Remember my password
	OK Cancel

• A new job will appear in the window **Job Manager.** If you select it, you will be able to track the execution of its tasks in the lower list. When the job is completed, its status will change for **Finished**,

Bite Year Show Al Jobs P 0 Name Prinibel Submit Term Pending Reason 1 hothame Normal CCAM Serin Faled 21.06.2006 4.10.14 3 inb Normal CCAM Serin Faled 21.06.2006 4.24.34 4 inb Normal CCAM Serin Faled 21.06.2006 4.25.23 5 sint Pi computing Normal CCAM Serin Faled 21.06.2006 5.13.20 5 Serial Pi computing Normal CCAM Serin Frinithed 21.06.2006 1.93.20 6 Serial Pi computing Normal CCAM Serin Frinithed 25.06.2006 3.00.2 3 Serial Pi computing Normal CCAM Serin Frinithed 25.06.2006 2.01.30 10 Padel Pi computing Normal CCAM Serin Frinithed 25.06.2006 2.01.30 11 Padel Pi computing Normal CCAM Serin Frinithed 25.06.2006 2.11.30 12 Padel Pi computing Normal CCAM Serin	Job Queue at s	-cw-head							-	
Imp Normal CCAM/Servin Finished 21.06.2006 410:14 12 imb Normal CCAM/Servin Faled 21.06.2006 419:50 21 imb Normal CCAM/Servin Faled 21.06.2006 42:24 24 imb Normal CCAM/Servin Faled 21.06.2006 42:23 25 imb Normal CCAM/Servin Faled 21.06.2006 51:320 26 Serial Pr computing Normal CCAM/Servin Fraihed 25.06.2006 2:00.46 36 Serial Pr computing Normal CCAM/Servin Fraihed 25.06.2006 2:01.50.22 37 Sreial Pr Computing Normal CCAM/Servin Fraihed 25.06.2006 2:13.00 310 Pacalel Pr computing Normal CCAM/Servin Fraihed 25.06.2006 2:13.00 311 Pacalel Pr computing Normal CCAM/Servin Fraihed 25.06.2006 2:11.22 312 Pacalel Pr computing Normal CCAM/Servin Fraihed 25.06.2006 2:11.23 313 Pacalel Pr comput	jie ⊻iew Jo	ob <u>H</u> elp Sha	w: All Jobs	-						
2 mb Nomal CCAM/Serin Falled 21.06.2006 41:950 3 inb Nomal CCAM/Serin Falled 21.06.2006 42:323 5 inb Nomal CCAM/Serin Failed 21.06.2006 42:323 5 inb Nomal CCAM/Serin Failed 21.06.2006 42:323 7 Serial P computing Nomal CCAM/Serin Failed 25.06.2006 1:93.22 8 Serial P computing Nomal CCAM/Serin Friabed 25.06.2006 4:08.33 10 Parallel F computing Nomal CCAM/Serin Friabed 25.06.2006 4:08.33 11 Parallel F computing Nomal CCAM/Serin Failed 25.06.2006 2:13.00 12 Parallel F computing Nomal CCAM/Serin Failed 25.06.2006 2:11.30 13 Parallel F computing Nomal CCAM/Serin Failed 25.06.2006 0:21.2 14 Parallel F computing Normal CCAM/Serin Failed 26.06.2006 0:31.30 15 Parallel F parametric sweep Normal CCAM/Serin Failed 26.06.2006 1:37.01	Name			Priority	Submitt	ed By	Status	Submit Time	Pending Reason	
2 mb Nomal CCAM/Serin Faled 21.06.2006 4.19.50 3 inb Nomal CCAM/Serin Faled 21.06.2006 4.23.23 4 inb Nomal CCAM/Serin Faled 21.06.2006 4.23.23 5 inb Nomal CCAM/Serin Faled 21.06.2006 4.23.23 6 Seial Picomputing Nomal CCAM/Serin Finished 21.06.2006 4.23.23 7 Seial Picomputing Nomal CCAM/Serin Finished 25.06.2006 1.09.22 8 Seial Picomputing Nomal CCAM/Serin Finished 25.06.2006 4.08.39 10 Paeallel Ficomputing Nomal CCAM/Serin Finished 25.06.2006 2.13.00 11 Paeallel Ficomputing Nomal CCAM/Serin Finished 25.06.2006 2.21.5.36 12 Paeallel Fi computing Nomal CCAM/Serin Finished 25.06.2006 0.231.30 13 Paeallel Fi computing Nomal CCAM/Serin Finished 26.06.2006 0.33.30 14 Paeallel Fi computing Nomal CCAM/Serin Finizhed 26.06.2006 0.33.30	1 hostna	me		Normal	CCAM\	Senin	Finished	21.06.2006 4:10:14		
3 imb Normal CCAM/Serin Faled 21.0620642424 4 imb Normal CCAM/Serin Faled 21.06200451320 5 imb Normal CCAM/Serin Faled 21.06200451320 6 Serial Pi computing Normal CCAM/Serin Fraibed 25.062003 20.045 8 Serial Pi computing Normal CCAM/Serin Fraibed 25.062003 20.045 9 Serial Pi computing Normal CCAM/Serin Fraibed 25.062003 20.045 10 Pasallel Pi computing Normal CCAM/Serin Fraibed 25.062003 25.06202 21.00 11 Pasallel Pi computing Normal CCAM/Serin Fraibed 25.062006 22.15.36 12 Pasallel Pi computing Normal CCAM/Serin Fraibed 25.062006 22.15.36 13 Pasallel Pi computing Normal CCAM/Serin Fraibed 25.062006 25.062006 21.2 15 Pasallel Pi computing Normal CCAM/Serin Fraibed 26.062006 25.062006	2 imb			Normal	CCAM\	Senin	Failed	21.06.2006 4:19:50		
5 minb Normal CCAM/Serin Finished 25.06.2006.513.20 6 Serial Pi computing Normal CCAM/Serin Finished 25.06.2006.19.922 7 Serial Pi computing Normal CCAM/Serin Finished 25.06.2006.19.922 8 Serial Pi computing Normal CCAM/Serin Finished 25.06.2006.00.40.8 10 Paralel Pi computing Normal CCAM/Serin Finished 25.06.2006.22.15.36 11 Paralel Pi computing Normal CCAM/Serin Finished 25.06.2006.22.15.36 12 Paralel Pi computing Normal CCAM/Serin Finished 25.06.2006.22.17.33 13 Paralel Pi computing Normal CCAM/Serin Finished 25.06.2006.02.312 14 Paralel Pi computing Normal CCAM/Serin Finished 25.06.2006.02.312 15 Paralel Pi computing Normal CCAM/Serin Finished 25.06.2006.02.312 16 Paralel Pi parametric sweep Normal CCAM/Serin Funished 25.06.2006.1.37/01	3 imb			Normal	CCAM\	Senin	Failed	21.06.2006 4:24:24		
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 3:50 22 9 Serial Pi computing Normal CCAM/Senin Finished 25 06 2006 3:50 22 9 Serial Pi Caculation Normal CCAM/Senin Finished 25 06 2006 22:13:00 10 Parallel Pi computing Normal CCAM/Senin Finished 25 06 2006 22:15:36 12 Parallel Pi computing Normal CCAM/Senin Finished 25 06 2006 22:15:36 13 Parallel Pi computing Normal CCAM/Senin Finished 25 06 2006 22:15:36 14 Parallel Pi computing Normal CCAM/Senin Finished 25 06 2006 02:31:20 15 Parallel Pi computing Normal CCAM/Senin Finished 26 06:2006 1:37:01 ks for Parallel Pi parametric sweep Normal CCAM/Senin Running 26 US: 2006 1:37:01 ks for Parallel Pi parametric sweep Normal CCAM/Senin Finished 26 06:2006 0:31:30 Parallel Pi normating Normal CCAM/Senin Finished 26 06:2006 1:37:01 Parallel Pi normating	4 imb			Normal	CCAM\	Senin	Failed	21.06.2006 4:25:23		
5 Serial Pi computing Normal CCAM/Serin Falled 25.06.2006 1:59.22 7 Serial Pi computing Normal CCAM/Serin Finished 25.06.2006 3:50:22 3 Serial Pi Catulation Normal CCAM/Serin Finished 25.06.2006 2:01:30 3 Serial Pi Catulation Normal CCAM/Serin Finished 25.06.2006 2:01:30 10 Parallel Pi computing Normal CCAM/Serin Finished 25.06.2006 2:01:30 11 Parallel Pi computing Normal CCAM/Serin Finished 25.06.2006 2:01:30 12 Parallel Pi computing Normal CCAM/Serin Finished 25.06.2006 0:21:13 13 Parallel Pi computing Normal CCAM/Serin Finished 25.06.2006 0:21:12 15 Parallel Pi computing Normal CCAM/Serin Finished 25.06.2006 0:31:30 16 Parallel Pi parametric sweep Normal CCAM/Serin Finished 26.06.2006 1:37.01 Parallel Pi parametric sweep Normal CCAM/Serin Finished 26.06.2006 1:37.01 No	5 imb			Normal	CCAM\	Senin	Finished	21.06.2006 5:13:20		
3 Serial Pi computing Normal CCAM/Sernin Finished 25.06.2006 3:00.2 9 Serial Pi Caculation Normal CCAM/Sernin Finished 25.06.2006 2:21:3:00 10 Parallel Pi computing Normal CCAM/Sernin Finished 25.06.2006 2:21:3:00 11 Parallel Pi computing Normal CCAM/Sernin Finished 25.06.2006 2:21:5:36 12 Parallel Pi computing Normal CCAM/Sernin Finished 25.06.2006 2:21:30 12 Parallel Pi computing Normal CCAM/Sernin Finished 25.06.2006 0:21:0 14 Parallel Pi computing Normal CCAM/Sernin Finished 25.06.2006 0:21:0 15 Parallel Pi computing Normal CCAM/Sernin Finished 25.06.2005 0:21:0 16 Parallel Pi parametric sweep Normal CCAM/Sernin Finished 25.06.2005 1:37.01 Parallel Pi parametric sweep Normal CCAM/Sernin Finished 25.06.2005 1:37.01 Parallel Pi computing Normal CCAM/Sernin Finished 25.06.2005 1:37	6 Serial	Pi computing		Normal	CCAM\	Senin	Failed	25.06.2006 1:59:22		
3 Serial PT Calculation Normal CCAM/Servin Finished 25,06,2006 4/08.39 10 Parallel PT computing Normal CCAM/Servin Failed 25,06,2006 2215.36 12 Parallel PT computing Normal CCAM/Servin Finished 25,06,2006 2215.36 13 Parallel PT computing Normal CCAM/Servin Finished 25,06,2006 2215.36 13 Parallel PT computing Normal CCAM/Servin Finished 25,06,2006 2215.36 14 Parallel PT computing Normal CCAM/Servin Finished 25,06,2006 0.23130 15 Parallel PT computing Normal CCAM/Servin Finished 25,06,2006 1.37,01 Status Task ld Command Line Processors End Time Failure Message Parallel PT Finished 1 mpiexec-hosts %CCP_NO 10 - 20 N/A Parallel PT Queued 4 mpiexec-hosts %CCP_NO 10 - 20 N/A Parallel PT Queued 7 mpiexec-hosts %CCP_NO 10 - 20 N/A Parallel PT Queued 7 <	7 Serial I	Pi computing		Normal	CCAM\	Senin	Finished	25.06.2006 2:00:46		
3 Serial PT Calculation Normal CCAM/Servin Finished 25,06,2006 4/08.39 10 Parallel PT computing Normal CCAM/Servin Failed 25,06,2006 2215.36 12 Parallel PT computing Normal CCAM/Servin Finished 25,06,2006 2215.36 13 Parallel PT computing Normal CCAM/Servin Finished 25,06,2006 2215.36 13 Parallel PT computing Normal CCAM/Servin Finished 25,06,2006 2215.36 14 Parallel PT computing Normal CCAM/Servin Finished 25,06,2006 0.23130 15 Parallel PT computing Normal CCAM/Servin Finished 25,06,2006 1.37,01 Status Task ld Command Line Processors End Time Failure Message Parallel PT Finished 1 mpiexec-hosts %CCP_NO 10 - 20 N/A Parallel PT Queued 4 mpiexec-hosts %CCP_NO 10 - 20 N/A Parallel PT Queued 7 mpiexec-hosts %CCP_NO 10 - 20 N/A Parallel PT Queued 7 <		Pi computing		Normal	CCAM\	Senin	Finished	25.06.2006 3:50:22		
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 Finished 25.06.2006 22.15.36 13 Parallel Pi computing Normal CCAM/Senin Finished 25.06.2006 22.17.33 14 Parallel Pi computing Normal CCAM/Senin Finished 25.06.2006 0.29.12 15 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 1.37.01 16 Parallel Pi parametric sweep Normal CCAM/Senin Running 26.06.2006 1.37.01 Status Task Id Command Line Processors End Time Failure Message Parallel Pi parametric sweep Vormal CCAM/Senin Finished 26.06.2006 1.37.01 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 1.37.01 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 1.37.01 <td co<="" td=""><td></td><td>Pi Calculation</td><td></td><td>Normal</td><td>CCAM\</td><td>Senin</td><td>Finished</td><td>25.06.2006 4:08:39</td><td></td></td>	<td></td> <td>Pi Calculation</td> <td></td> <td>Normal</td> <td>CCAM\</td> <td>Senin</td> <td>Finished</td> <td>25.06.2006 4:08:39</td> <td></td>		Pi Calculation		Normal	CCAM\	Senin	Finished	25.06.2006 4:08:39	
11 Parallel Pi computing Normal CCAM/Servin Finished 25.06.2006 22:15:36 12 Parallel Pi computing Normal CCAM/Servin Finished 25.06.2006 22:17:33 14 Parallel Pi computing Normal CCAM/Servin Finished 26.06.2006 22:17:33 14 Parallel Pi computing Normal CCAM/Servin Finished 26.06.2006 0:31:30 15 Parallel Pi parametric sweep Normal CCAM/Servin Finished 26.06.2006 1:37:01	10 Paralle	l Pi computing		Normal	CCAM\	Senin	Failed	25.06.2006 22:13:00		
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 22:17:33 15 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 23:13 16 Parallel Pi parametric sweep Normal CCAM\Senin Running 26:06:2006 1:37:01 ks for Parallel Pi parametric sweep Normal CCAM\Senin Running 26:06:2006 1:37:01		l Pi computing		Normal	CCAM\	Senin	Finished	25.06.2006 22:15:36		
3 Parallel Pi computing Normal CCAM/Senin Finished 25.06.2006.22.17.33 4 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006.0.23.12 5 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 ks for Parallel Pi parametric sweep te Status Task Id Command Line Processors End Time Failure Message arallel Pi parametric sweep Vormal CCAM/Senin To 20 20.66.2006 Parallel Pi Finished 1 mpiexec-hosts 2CCP_NO 10 - 20 N/A Parallel Pi Running 2 projece chosts 2CCP_NO 10 - 20 N/A Parallel Pi Queued 3 mpiexec-hosts 2CCP_NO 10 - 20 N/A Parallel Pi Queued 4 mpiexec-hosts 2CCP_NO 10 - 20 N/A Parallel Pi Queued 7 mpiexec-hosts 2CCP_NO 10 - 20 N		l Pi computing		Normal	CCAM\	Senin	Failed	25.06.2006 22:16:49		
4 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0.2312 5 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0.23130 6 Parallel Pi parametric sweep Normal CCAM\Senin Running 26.06.2006 1.37.01 ks for Parallel Pi parametric sweep result Task.Id Command Line Processors End Time Falure Message Parallel Pi parametric sweep set Status Task.Id Command Line Processors End Time Falure 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 <t< td=""><td>3 Paralle</td><td>l Pi computing</td><td></td><td>Normal</td><td>CCAM\</td><td>Senin</td><td>Finished</td><td>25.06.2006 22:17:33</td><td></td></t<>	3 Paralle	l Pi computing		Normal	CCAM\	Senin	Finished	25.06.2006 22:17:33		
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 ks for Parallel Pi parametric sweep ne Status Task Id Command Line Processors End Time Failure Message Parallel Pi Finished 1 mpiexec -hosts %CCP_NO 10 - 20 X/A 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 7 mpiexec -hosts %CCP_NO 10 - 20 N/A Parallel Pi		l Pi computing		Normal			Finished	26.06.2006 0:29:12		
I6 Parallel Pi parametric sweep Normal CCAM\Senin Running 26.06.2006 1:37:01 ks for Parallel Pi parametric sweep Normal Command Line Processors End Time Failure Message Brailel Pi Status Task Id Command Line Processors End Time Failure Message Brailel Pi Finished 1 mpiexec hosts %CCP_NO 10 · 20 25.06.2006 Brailel Pi Running 2 mpiexec hosts %CCP_NO 10 · 20 N/A Brailel Pi Queued 3 mpiexec hosts %CCP_NO 10 · 20 N/A Brailel Pi Queued 4 mpiexec hosts %CCP_NO 10 · 20 N/A Brailel Pi Queued 5 mpiexec hosts %CCP_NO 10 · 20 N/A Brailel Pi Queued 6 mpiexec hosts %CCP_NO 10 · 20 N/A Brailel Pi Queued 7 mpiexec hosts %CCP_NO 10 · 20 N/A Brailel Pi Queued 9 mpiexec hosts %CCP_NO 10 · 20 N/A Brailel Pi Queued 9 mpiexec hosts %CCP_NO 10 · 20 N/A Brailel Pi Queued 10 mpiexec hosts %CCP_NO 10 · 20 N/A		l Pi computing		Normal	CCAM\	Senin	Finished	26.06.2006 0:31:30		
ks for Parallel Pi parametric sweep ne Status Task Id Command Line Processors End Time Failure Message Parallel Pi Finished 1 mpiexec hosts %CCP_NO 10 - 20 25.06.2006 Parallel Pi Queued 3 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 <td></td> <td></td> <td>en</td> <td>Normal</td> <td>CCAM\</td> <td>Senin</td> <td>Bunning</td> <td>26.06.2006.1:37:01</td> <td></td>			en	Normal	CCAM\	Senin	Bunning	26.06.2006.1:37:01		
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 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 9 mpiexec -hosts %CCP_NO 10 - 20 N/A Parallel Pi Queued 10 mpiexec -hosts %CCP_NO 10 - 20 N/A Parallel Pi Queued </th <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>										
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 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 <td< th=""><th></th><th>-,</th><th></th><th>- (</th><th></th><th></th><th>1</th><th>1</th><th></th></td<>		-,		- (1	1		
Parallel Pi Running 2 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 3 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 4 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 5 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 6 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 7 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 7 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 8 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 9 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 10 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 11 mpiexec -hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued										
Parallel Pi Queued 3 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 4 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 5 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 6 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 7 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 8 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 9 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 10 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 10 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 11 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 11 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 12 mpiexec -hosts %CCP_N0 10 - 20 N/A A new job 16										
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 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 11 mpiexec hosts %CCP_NO 10 · 20 N/A Parallel Pi Queued 12 mpiexec hosts %CCP_NO 10 · 20 N/A		-		•	_					
Parallel Pi Queued 5 mpiexec -hosts %CCP_ND 10 - 20 N/A Parallel Pi Queued 6 mpiexec -hosts %CCP_ND 10 - 20 N/A Parallel Pi Queued 7 mpiexec -hosts %CCP_ND 10 - 20 N/A Parallel Pi Queued 8 mpiexec -hosts %CCP_ND 10 - 20 N/A Parallel Pi Queued 8 mpiexec -hosts %CCP_ND 10 - 20 N/A Parallel Pi Queued 9 mpiexec -hosts %CCP_ND 10 - 20 N/A Parallel Pi Queued 10 mpiexec -hosts %CCP_ND 10 - 20 N/A Parallel Pi Queued 11 mpiexec -hosts %CCP_ND 10 - 20 N/A Parallel Pi Queued 11 mpiexec -hosts %CCP_ND 10 - 20 N/A Parallel Pi Queued 12 mpiexec -hosts %CCP_ND 10 - 20 N/A										
Parallel Pi Queued 6 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 7 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 8 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 9 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 10 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 11 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 11 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 11 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 12 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 12 mpiexec hosts %CCP_N0 10 - 20 N/A										
Parallel Pi Queued 7 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 8 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 9 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 10 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 11 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 11 mpiexec -hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 12 mpiexec -hosts %CCP_N0 10 - 20 N/A										
Parallel Pi Queued 8 mpiexec hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 9 mpiexec hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 10 mpiexec hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 11 mpiexec hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 11 mpiexec hosts %CCP_N0 10 · 20 N/A Parallel Pi Queued 12 mpiexec hosts %CCP_N0 10 · 20 N/A		-								
Parallel Pi Queued 9 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 10 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 11 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 11 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 12 mpiexec hosts %CCP_N0 10 - 20 N/A					_					
Parallel Pi Queued 10 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 11 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 12 mpiexec hosts %CCP_N0 10 - 20 N/A Parallel Pi Queued 12 mpiexec hosts %CCP_N0 10 - 20 N/A										
Parallel Pi Queued 11 mpiexec hosts %CCP_N0 10 - 20 N/A Compute Cluster Job Manager X Parallel Pi Queued 12 mpiexec hosts %CCP_N0 10 - 20 N/A A new job 16 is submitted. A new job 16 is submitted.							N/A			
Parallel Pi Queued 11 mpiexec hosts %CCP_N0 10-20 N/A A new job 16 is submitted. Parallel Pi Queued 12 mpiexec hosts %CCP_N0 10-20 N/A A new job 16 is submitted.						10 - 20				
Falalier Fi Guedeu 12 hiplexed hosts & CCF_NO 10 20 NVA Discourse the factor of the		Queued		mainten la sta 2/0				ť		
Parallel Pi Dueued 13 mnievec boots %CCP NO 10.20 N/A	Parallel Pi	Queued				10 - 20		ĺ		
				mpiexec -hosts %0	CCP_NO	10 - 20	N/A N/A	ĺ	A new job 16 is submitted.	

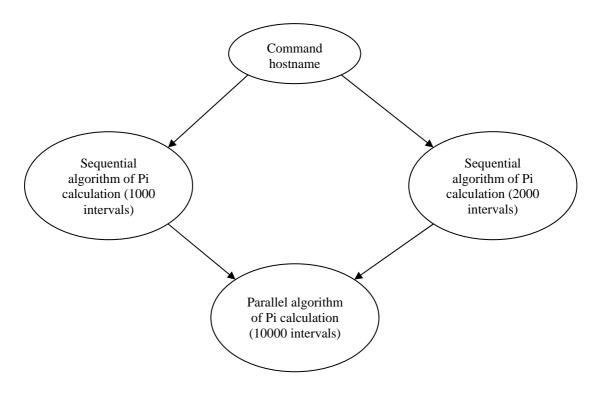
• You can look through the results of the job execution in the files, which you have specified for saving the redirected output stream.

두 temp on s-cw-head		_ 8 ×
File Edit View Favorites Too	ols Help	N
🕞 Back 🔻 🕥 👻 🏂 🔎	Search 😥 Folders 🛄 🕇	
Address 😌 \\s-cw-head\temp		💌 🄁 Go
	Name 🔺	Size Type Date Modified
File and Folder Tasks →	parallopicorexe	1 KB Text Document 26.06.2006 12:37
	parallelpi51.txt	1 KB Text Document 26.06.2006 12:37
Other Places *	parallelpi52.txt parallelpi53.txt	1 KB Text Document 26.06.2006 12:37 1 KB Text Document 26.06.2006 12:37
👮 s-cw-head	parallelpi54.txt	1 KB Text Document 26.06.2006 12:37 1 KB Text Document 26.06.2006 12:37
A My Documents	Darallelpi55.txt	1 KB Text Document 26.06.2006 12:37
My Computer	parallelpi56.txt	1 KB Text Document 26.06.2006 12:37
My Network Places	parallelpi57.txt	1 KB Text Document 26.06.2006 12:37
My Network Places	🗊 parallelpi58.txt	1 KB Text Document 26.06.2006 12:37
	🗊 parallelpi59.txt	1 KB Text Document 26.06.2006 12:37
Details 🕆	📃 parallelpi60.txt	1 KB Text Document 26.06.2006 12:37
	📃 parallelpi61.txt	1 KB Text Document 26.06.2006 12:37
parallelpi50.txt	📃 parallelpi62.txt	📑 parallelpi50.txt - Notepad 📃 🗖 🗙
Text Document	🗐 parallelpi63.txt	File Edit Format View Help
	🕖 parallelpi64.txt	Process 0 on s-cw2-15.cluster.cmc.unn.net
	🗊 parallelpi65.txt	Process 1 on s-cw2-15.cluster.cmc.unn.net
	parallelpi66.txt	Process 2 on s-cw2-15.cluster.cmc.unn.net Process 3 on s-cw2-15.cluster.cmc.unn.net
	parallelpi67.txt parallelpi68.txt	Process 4 on s-cw2-14.cluster.cmc.unn.net
	parallelpi69.txt	Process 5 on s-cw2-14.cluster.cmc.unn.net Process 6 on s-cw2-14.cluster.cmc.unn.net
	Darallelpi70.txt	Process 7 on s-cw2-14.cluster.cmc.unn.net
	parallelpi71.txt	Process & on s-cw-head.cluster.cmc.unn.net
	parallelpi72.txt	Process 9 on s-cw-head.cluster.cmc.unn.net Process 10 on s-cw-head.cluster.cmc.unn.net
	parallelpi73.txt	Process 11 on s-cw-head.cluster.cmc.unn.net
	🗊 parallelpi74.txt	Process 12 on s-cw2-13.cluster.cmc.unn.net Process 13 on s-cw2-13.cluster.cmc.unn.net
	🗊 parallelpi75.txt	Process 14 on s-cw2-13.cluster.cmc.unn.net
	🗐 parallelpi76.txt	Process 15 on s-cw2-13.cluster.cmc.unn.net
	🗐 parallelpi77.txt	Process 16 on s-cw2-08.cluster.cmc.unn.net Process 17 on s-cw2-08.cluster.cmc.unn.net
	📃 parallelpi78.txt	Process 18 on s-cw2-08.cluster.cmc.unn.net
	📃 parallelpi79.txt	Process 19 on s-cw2-08.cluster.cmc.unn.net NumIntervals = 50
	🗐 parallelpi80.txt	PI is approximately 3.1416259869230041, Error is 0.0000333333332110
	🕖 parallelpi81.txt	
	parallelpi82.txt	
	parallelpi83.txt	
	🗐 parallelpi84.txt 🗐 parallelpi85.txt	
	parallelpi86.txt	V
	parallelpi87.txt	
	paralicipio/.cxc	

Exercise 5 – Launching a Work Flow

The work flow is used if the execution of a certain task within a job requires the results of the other task execution, which creates requirements to the task execution sequence. These requirements are convenient to set in the form of the acyclic oriented graph, where each vertex is a task, and the arrow shows the dependence of the vertex - child against the vertex - parent. In this case the task execution sequence is defined by the following simple rule: neither of the tasks can be launched until all the tasks, which correspond to its parents on the dependence graph, are executed.

We can consider the following task dependence graph as an example:



Let us set this dependence graph in CCS 2003:

• Open Computer Cluster Job Manager (Start->All Programs->Microsoft Compute Cluster Pack->Compute Cluster Job Manager),

Name Priority Submitted By Status Submit Time Pending Reason hostname Normal CCAM/Senin Finished 21.06.2006 4:19:50 imb Normal CCAM/Senin Failed 21.06.2006 4:19:50 imb Normal CCAM/Senin Failed 21.06.2006 4:19:50 imb Normal CCAM/Senin Failed 21.06.2006 4:24:24 imb Normal CCAM/Senin Failed 21.06.2006 4:24:24 imb Normal CCAM/Senin Failed 21.06.2006 4:24:24 imb Normal CCAM/Senin Finished 21.06.2006 4:20:31 Serial Pi computing Normal CCAM/Senin Finished 25.06.2006 1:59:22 Serial Pi computing Normal CCAM/Senin Finished 25.06.2006 5:02.2 Parallel Pi computing Normal CCAM/Senin Finished 25.06.2006 2:15.30 Parallel Pi computing Normal CCAM/Senin Finished	hostnameNormalCCAM\SeninFinished21.06.2006 4:10:14imbNormalCCAM\SeninFailed21.06.2006 4:24:24imbNormalCCAM\SeninFailed21.06.2006 4:24:24imbNormalCCAM\SeninFailed21.06.2006 4:25:23imbNormalCCAM\SeninFinished21.06.2006 5:13:20Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2:00:46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2:00:46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2:00:46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2:00:46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2:00:46Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 2:10:00Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 2:15:36Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 2:17:33Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 0:21:2Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 0:21:30Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 0:21:30Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 0:21:30Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 1:37:01Parallel Pi parametric swe	[N	Show: All Jo		•		Leu	Le La Tr		1
imbNormalCCAM\SeninFailed21.06.2006 4:19:50imbNormalCCAM\SeninFailed21.06.2006 4:24:24imbNormalCCAM\SeninFailed21.06.2006 5:13:20Serial Pi computingNormalCCAM\SeninFailed25.06.2006 5:13:20Serial Pi computingNormalCCAM\SeninFinished25.06.2006 5:09:22Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2:00:46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2:00:46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2:00:46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2:00:46Serial Pi computingNormalCCAM\SeninFailed25.06.2006 2:00:46Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:00:46Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:10:30Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 2:17:33Parallel Pi computingNormalCCAM\SeninFinished26.06.2006 0:31:30Parallel Pi parametric sweepNormalCCAM\SeninFinished26.06.2006 1:37:01Apple to view its tasksNormalCCAM\SeninFinished26.06.2006 1:37:01	imbNormalCCAM/SeninFailed21.06.2006 4:19:50imbNormalCCAM/SeninFailed21.06.2006 4:24:24imbNormalCCAM/SeninFailed21.06.2006 4:25:23imbNormalCCAM/SeninFinished21.06.2006 5:13:20Serial Pi computingNormalCCAM/SeninFinished25.06.2006 5:13:22Serial Pi computingNormalCCAM/SeninFinished25.06.2006 2:00:46Serial Pi computingNormalCCAM/SeninFinished25.06.2006 4:08:33Serial Pi computingNormalCCAM/SeninFinished25.06.2006 4:08:33Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:21:3:00Parallel Pi computingNormalCCAM/SeninFailed25.06.2006 2:21:3:00Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:21:3:36Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:21:3:36Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 0:23:1:2Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 0:31:30Parallel Pi parametric sweepNormalCCAM/SeninFinished26.06.2006 1:37:01						_		Pending Reason	
imbNormalCCAM\SeninFailed21,06,2006 4:24:24imbNormalCCAM\SeninFailed21,06,2006 4:25:23imbNormalCCAM\SeninFailed21,06,2006 5:13:20Serial Pi computingNormalCCAM\SeninFailed25,06,2006 1:59:22Serial Pi computingNormalCCAM\SeninFinished25,06,2006 3:50:22Serial Pi computingNormalCCAM\SeninFinished25,06,2006 4:08:39Parallel Pi computingNormalCCAM\SeninFailed25,06,2006 2:01:46Parallel Pi computingNormalCCAM\SeninFailed25,06,2006 2:13:00Parallel Pi computingNormalCCAM\SeninFailed25,06,2006 2:21:36Parallel Pi computingNormalCCAM\SeninFailed25,06,2006 2:21:36Parallel Pi computingNormalCCAM\SeninFinished25,06,2006 2:21:36Parallel Pi computingNormalCCAM\SeninFinished25,06,2006 0:21:13Parallel Pi computingNormalCCAM\SeninFinished26,06,2006 0:21:12Parallel Pi computingNormalCCAM\SeninFinished26,06,2006 1:37:01Parallel Pi parametric sweepNormalCCAM\SeninFinished26,06,2006 1:37:01	imbNormalCCAM\SeninFailed21.06.2006 4.24:24imbNormalCCAM\SeninFailed21.06.2006 5.13:20imbNormalCCAM\SeninFailed25.06.2006 5.13:20Serial Pi computingNormalCCAM\SeninFailed25.06.2006 1.53:22Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2.00.46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2.00.46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2.00.46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2.13:00Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 2.21:3:00Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 2.21:3:00Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 2.21:3:00Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 0.23:1:2Parallel Pi computingNormalCCAM\SeninFinished26.06.2006 0.23:1:2Parallel Pi computingNormalCCAM\SeninFinished26.06.2006 1:37:01Parallel Pi parametric sweepNormalCCAM\SeninFinished26.06.2006 1:37:01									
imbNormalCCAM\SeninFailed21,06,2006 4:25:23imbNormalCCAM\SeninFinished21,06,2006 5:13:20Serial Pi computingNormalCCAM\SeninFinished25,06,2006 1:59:22Serial Pi computingNormalCCAM\SeninFinished25,06,2006 3:50:22Serial Pi computingNormalCCAM\SeninFinished25,06,2006 4:08:39Parallel Pi computingNormalCCAM\SeninFinished25,06,2006 2:13:00Parallel Pi computingNormalCCAM\SeninFinished25,06,2006 2:13:00Parallel Pi computingNormalCCAM\SeninFinished25,06,2006 2:21:3:06Parallel Pi computingNormalCCAM\SeninFinished25,06,2006 2:21:3:06Parallel Pi computingNormalCCAM\SeninFinished25,06,2006 2:21:3:06Parallel Pi computingNormalCCAM\SeninFinished25,06,2006 2:21:3:06Parallel Pi computingNormalCCAM\SeninFinished26,06,2006 0:23:1:2Parallel Pi computingNormalCCAM\SeninFinished26,06,2006 0:23:1:2Parallel Pi computingNormalCCAM\SeninFinished26,06,2006 1:37:01Parallel Pi parametric sweepNormalCCAM\SeninFinished26,06,2006 1:37:01	imbNormalCCAM/SeninFailed21.06.2006 4:25.23imbNormalCCAM/SeninFinished21.06.2006 5:13.20Serial Pi computingNormalCCAM/SeninFinished25.06.2006 1:59.22Serial Pi computingNormalCCAM/SeninFinished25.06.2006 4:08.39Serial Pi computingNormalCCAM/SeninFinished25.06.2006 4:08.39Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:13:00Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:13:00Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:16:49Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 0:29:12Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 0:29:12Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 0:29:12Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 1:37:01Parallel Pi parametric sweepNormalCCAM/SeninFinished26.06.2006 1:37:01									
imbNormalCCAM\SeninFinished21.06.2006 5:13:20Serial Pi computingNormalCCAM\SeninFailed25.06.2006 1:59:22Serial Pi computingNormalCCAM\SeninFinished25.06.2006 3:022Serial Pi calculationNormalCCAM\SeninFinished25.06.2006 4:08:39Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:21:300Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:21:300Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:21:536Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:15:36Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 2:21:6.49Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 0:23:12Parallel Pi computingNormalCCAM\SeninFinished26.06.2006 0:31:30Parallel Pi parametric sweepNormalCCAM\SeninFinished26.06.2006 1:37:01	imbNormalCCAM/SeninFinished21.06.2006 5:13.20Serial Pi computingNormalCCAM/SeninFaled25.06.2006 1:59:22Serial Pi computingNormalCCAM/SeninFinished25.06.2006 3:02.46Serial Pi CalculationNormalCCAM/SeninFinished25.06.2006 4:08:39Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:13:00Parallel Pi computingNormalCCAM/SeninFailed25.06.2006 2:16:36Parallel Pi computingNormalCCAM/SeninFailed25.06.2006 2:16:49Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:17:33Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 0:31:30Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 0:31:30Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 1:37:01Parallel Pi parametric sweepNormalCCAM/SeninFinished26.06.2006 1:37:01									
Serial Pi computingNormalCCAM\SeninFailed25.06.2006 1:59:22Serial Pi computingNormalCCAM\SeninFinished25.06.2006 2:00:46Serial Pi computingNormalCCAM\SeninFinished25.06.2006 4:00:39Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:13:00Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:21:8:00Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:21:8:00Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:21:8:00Parallel Pi computingNormalCCAM\SeninFailed25.06.2006 2:21:8:00Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 2:21:8:49Parallel Pi computingNormalCCAM\SeninFinished25.06.2006 0:29:12Parallel Pi computingNormalCCAM\SeninFinished26.06.2006 0:31:30Parallel Pi computingNormalCCAM\SeninFinished26.06.2006 1:37:01Parallel Pi parametric sweepNormalCCAM\SeninFinished26.06.2006 1:37:01	Serial Pi computingNormalCCAM/SeninFailed25.06.2006 1:59:22Serial Pi computingNormalCCAM/SeninFinished25.06.2006 3:50:22Serial Pi computingNormalCCAM/SeninFinished25.06.2006 4:08:33Parallel Pi computingNormalCCAM/SeninFailed25.06.2006 2:13:00Parallel Pi computingNormalCCAM/SeninFailed25.06.2006 2:15:36Parallel Pi computingNormalCCAM/SeninFailed25.06.2006 2:16:49Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:17:33Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 0:29:12Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 1:37:01Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 1:37:01Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 1:37:01Parallel Pi parametric sweepNormalCCAM/SeninFinished26.06.2006 1:37:01									
Serial Pi computing Normal CCAM\Senin Finished 25.06.2006 2:00.46 Serial Pi computing Normal CCAM\Senin Finished 25.06.2006 4:08:39 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 2:13:00 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 2:15:36 Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 2:21:300 Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 2:21:300 Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 2:21:5:36 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 2:21:6:49 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 0:29:12 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0:29:12 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 1:37:01 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 1:37:01 Parallel Pi computing Normal CCAM\Senin	Serial Pi computingNormalCCAM/SeninFinished25.06.2006 2:00.46Serial Pi computingNormalCCAM/SeninFinished25.06.2006 4:08:39Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:13:00Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:15:36Parallel Pi computingNormalCCAM/SeninFinished25.06.2006 2:17:33Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 2:17:33Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 0:29:12Parallel Pi computingNormalCCAM/SeninFinished26.06.2006 1:37:01Parallel Pi parametric sweepNormalCCAM/SeninFinished26.06.2006 1:37:01		moutina							
Serial Pi computing Normal CCAM\Senin Finished 25.06.2006 3:50:22 Serial Pi Calculation Normal CCAM\Senin Finished 25.06.2006 4:08:39 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 22:13:00 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 22:15:36 Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 22:16:49 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 22:17:33 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 02:12 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 02:13:0 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 01:3:0 Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01	Serial Pi computing Normal CCAM\Senin Finished 25.06.2006 4:08:39 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 4:08:39 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 2:13:00 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 2:15:36 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 2:16:49 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 2:17:33 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0:29:12 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0:29:12 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0:29:13 Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01									
Serial Pi Calculation Normal CCAM\Senin Finished 25.06.2006 4:08:39 Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 22:18:00 Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 22:18:30 Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 22:18:39 Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 22:17:33 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 0:29:12 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 0:31:30 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0:31:30 Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01	Serial Pi Calculation Normal CCAM/Senin Finished 25.06.2006 4:08:39 Parallel Pi computing Normal CCAM/Senin Failed 25.06.2006 22:13:00 Parallel Pi computing Normal CCAM/Senin Failed 25.06.2006 22:15:36 Parallel Pi computing Normal CCAM/Senin Failed 25.06.2006 22:16:49 Parallel Pi computing Normal CCAM/Senin Finished 25.06.2006 22:17:33 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 0:29:12 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 0:31:30 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 1:37:01 Parallel Pi parametric sweep Normal CCAM/Senin Finished 26.06.2006 1:37:01									
Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 22:13:00 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 22:15:36 Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 22:16:49 Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 22:17:33 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 02:17:33 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 02:12 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0:31:30 Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01	Parallel Pi computing Normal CCAM/Senin Failed 25.06.2006 22:13:00 Parallel Pi computing Normal CCAM/Senin Finished 25.06.2006 22:15:36 Parallel Pi computing Normal CCAM/Senin Failed 25.06.2006 22:16:49 Parallel Pi computing Normal CCAM/Senin Finished 25.06.2006 22:17:33 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 0:29:12 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 0:31:30 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 1:37:01 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 1:37:01 Parallel Pi parametric sweep Normal CCAM/Senin Finished 26.06.2006 1:37:01									
Parallel Pi computing Normal CCAM\Senin Failed 25.06.2006 22:16:49 Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 02:17:33 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 02:12 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 02:12 Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01	Parallel Pi computing Normal CCAM/Senin Failed 25.06.2006 22:16:49 Parallel Pi computing Normal CCAM/Senin Finished 25.06.2006 0:29:12 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 0:29:12 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 0:29:12 Parallel Pi parametric sweep Normal CCAM/Senin Finished 26.06.2006 0:29:130 Parallel Pi parametric sweep Normal CCAM/Senin Finished 26.06.2006 1:37:01									
Parallel Pi computing Normal CCAM\Senin Finished 25.06.2006 02:17:33 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0:29:12 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0:31:30 Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01 ajob to view its tasks	Parallel Pi computing Normal CCAM/Senin Finished 25.06.2006 22:17:33 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 0:29:12 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 0:31:30 Parallel Pi parametric sweep Normal CCAM/Senin Finished 26.06.2006 1:37:01					CCAM\Senin	Finished			
Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0.29:12 Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0.31:30 Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01	Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 0.29:12 Parallel Pi computing Normal CCAM/Senin Finished 26.06.2006 0.31:30 Parallel Pi parametric sweep Normal CCAM/Senin Finished 26.06.2006 1:37:01	Parallel Pi c	omputing		Normal	CCAM\Senin	Failed	25.06.2006 22:16:49		
Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0.31:30 Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01 a job to view its tasks	Parallel Pi computing Normal CCAM\Senin Finished 26.06.2006 0.31:30 Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01	Parallel Pi c	omputing		Normal	CCAM\Senin	Finished	25.06.2006 22:17:33		
Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01	Parallel Pi parametric sweep Normal CCAM\Senin Finished 26.06.2006 1:37:01	Parallel Pi c	omputing		Normal	CCAM\Senin	Finished	26.06.2006 0:29:12		
a job to view its tasks	a job to view its tasks	Parallel Pi c	omputing		Normal	CCAM\Senin	Finished	26.06.2006 0:31:30		
		Parallel Pi p	arametric sweep		Normal	CCAM\Senin	Finished	26.06.2006 1:37:01		
	a job to view its tasks Status Task Id Command Line Processors End Time Failure Message									
Status Task Id Command Line Processors End Time Failure Message	Status Task Id Command Line Processors End Time Failure Message									
			stasks							
			stasks	Task Id	Command Line	Processors	End Time	Failure Message		
			stasks	Task Id	Command Line	Processors	End Time	Failure Message		
			stasks	Task Id	Command Line	Processors	End Time	Failure Message		
			stasks	Task Id	Command Line	Processors	End Time	Failure Message		

- In the new window of job manager choose the menu option File->Submit Job,
- In the window of adding a job to the queue enter the name of the job (the field **Job Name**). Go to the window tab **Processors**,

iubmit Job Example	of task flow			×
General Processors	Tasks Licenses	Advanced		
Example of t	task flow			
Job Name:	Example of task	flow		
P <u>r</u> oject Name:				
<u>P</u> riority:	Normal	•		
Submitted By:	N/A			
Submitted on:	N/A			
Status:	Not Submitted			
<u>S</u> ave As Templa	te		Submit	Cancel

• In the window tab **Processors** enter the minimum and the maximum number of processors required for executing the job (for instance, 5 and 10 correspondingly). Go to the window tab **Tasks** to add new tasks to the job,

5ubmit 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	
☐ <u>E</u> stimate 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.	
Submit Cancel	

- Add the following 4 tasks to the job sequentially:
 - The task named "Hostname" with the command "hostname.exe",
 - The task named "Serial Pi 1000" with the command "\\s-cw-head\temp\serialpi.exe 1000" (change the path to the executed file of the program for the existing one),
 - The task named "Serial Pi 2000" with the command "\\s-cw-head\temp\serialpi.exe 2000" (change the path to the executed file of the program for the existing one),
 - The task named "Parallel Pi 10000" with the command "mpiexec -hosts %CCP_NODES% \\s-cw-head\temp\parallelpi.exe 10000" (change the path to the executed file of the program for the existing one),

Submit Job Example of task flow
General Processors Tasks Licenses Advanced
Task Command Line
This job contains the following tasks:
Order Command Line Processors 1 hostname.exe 5 - 10 2 \\s-cw-head\temp\serialpi.exe 1000 5 - 10 3 \\s-cw-head\temp\serialpi.exe 2000 5 - 10 4 mpiexec -hosts %CCP_NODES% \\s 5 - 10
Add Parametric Sweep What is it?
Task Summary Name :Hostname Command Line :hostname.exe Standard Input : Standard Output : Standard Error : Work Directory : Number of processors requested :5 - 10 RunTime :Infinite Preceding tasks(dependent tasks) : Exclusive :False
Submit Cancel

- Set the additional parameters of the tasks:
 - For the tasks "Hostname", "Serial Pi 1000" and "Serial Pi 2000" set the maximum required number of processors as 1, set the file for redirecting the standard output stream for each of the 3 tasks,
 - For the task "**Parallel Pi 10000**" set the minimum and the maximum required numbers of processors as 5 and 10 correspondingly, set the file for redirecting the standard output stream,
- Go to the window tab **Tasks Dependencies** of the task properties (to go to the task properties, press the button **Edit** in the window tab **Tasks** of the window **Submit Job**),

iubmit Job Example of task flow
General Processors Tasks Licenses Advanced
Task Command Line <u>I</u> ask Name: <u>C</u> ommand Line:
✓ Use job's allocated processors
Minimum Pr <u>o</u> cessors: <u>5</u> Maximum <u>P</u> rocessors: <u>10</u>
This job contains the following tasks:
Order Command Line Processors Remove 1 hostname.exe 1 2 \\s-cw-head\temp\serialpi.exe 1000 1 3 \\s-cw-head\temp\serialpi.exe 2000 1 4 mpiexec -hosts %CCP_NODES% \\s 5 - 10
Add Parametric Sweep What is it?
Task Summary Name :Hostname Command Line :hostname.exe Standard Input : Standard Output :\\s-cw-head\temp\hostname.txt Standard Error : Work Directory : Number of processors requested :1 BunTime :Infinite Preceding tasks(dependent tasks) : Exclusive :False
Submit Cancel

• Select the task "Serial Pi 1000" and press the button Preceding Tasks to set the tasks, from which the considered task depends on,

Task Properties	×
Tasks Processors Tasks Dependencies Environment Advanced	
Specify tasks that must be completed before another task begins. Select a task below, then click Preceding Tasks to establish the task dependency.	
Tasks:	
Name Hostname Sequential Pi 1000 Sequential Pi 10000 Parallel Pi 10000	
Preceding Tasks	
OK Cancel	Apply

• In the new window tick the task "Hostname". Press OK,

Preceding Tasks	×
Selected Task:	
Sequential Pi 1000	
Select Tasks that are required:	
Hostname Sequential Pi 2000 Parallel Pi 10000	
0K	Cancel

• Set the dependence against "Hostname" for the task "Serial Pi 2000". Set the dependence against the tasks "Serial Pi 1000" and "Serial Pi 2000" for the task "Parallel Pi 10000". Press OK to change the changes to be done,

Task Properties				×
Tasks Processors	Tasks Dependencies	Environment A	dvanced	
Specify tasks that below, then click f Tasks:	must be completed bef Preceding Tasks to esta	ore another task b	egins. Select a ta:	sk
Name		Required Tasks		
Hostname Sequential Pi 100 Sequential Pi 10000 Parallel Pi 10000		Hostname Hostname Sequential Pi 1000),Sequential Pi 200	0
Preceding Tas	ks			
		ОК	Cancel	Apply

• In the window **Submit Job** press the button **Submit** to add the job to the queue,

Submit Job Example of task flow
General Processors Tasks Licenses Advanced
Task Command Line
This job contains the following tasks:
Order Command Line Processors <u>Remove</u> 1 hostname.exe 1 2 \\s-cw-head\temp\serialpi.exe 1000 1 3 \\s-cw-head\temp\serialpi.exe 2000 1 4 mpiexec -hosts %CCP_NODES% \\s 5 - 10
Add Parametric Sweep What is it?
Task Summary Name :Hostname Command Line :hostname.exe Standard Input : Standard Output :\\s-cw-head\temp\hostname.txt Standard Error : Work Directory : Number of processors requested :1 RunTime :Infinite Preceding tasks(dependent tasks) : Exclusive :False
Submit Cancel

- Enter the name and the password of the user authorized to run jobs on the cluster,
- The scheduler CCS 2003 will first start the task "Hostname", then the tasks "Serial Pi 1000" and "Serial Pi 2000" in parallel, and only after that it will start the task "Parallel Pi 10000".

Optional Exercise. Evaluating the Network Performance Parameters

The necessity to take into account not only the characteristics of individual computers (first of all the processor performance and the memory rate), but also the performance parameters of the network transmitting the data among them, has to be in the centre of attention in the process of efficient parallel program development for cluster systems. These parameters are often used for formulating the theoretical estimations of the algorithm execution time. It makes possible to predict the program execution time depending on the transmitted data size. Obtaining the network performance parameters is a separate problem, which is solved by means of launching special test programs on the particular available equipment. It is necessary to carry out these tests on each particular cluster because the data provided by the hardware vendor may vary to a great extent depending on the available software and cluster settings.

General Network Performance Parameters

The basic parameters, which are widely applied to characterize the network performance, are the latency and the bandwidth. **The latency** (the delay) is the time, spent by the hardware and the software for processing the request of sending a network message, i.e. this is the interval of time from the moment when the command to transmit data is entered to the beginning of the data transmission. Usually the latency is given in microseconds.

The network bandwidth is the maximum amount of data that can be transmitted by the network channels at a time unit. Usually it is measured in Mbyte/sec or Mbit/sec.

Methods for Evaluating the Network Performance Parameters

The basic idea of the algorithm for determining the network performance parameters, which is used in the tests of given Lab, consists in sequential transmitting messages of various lengths between two nodes, using the functions of the installed MPI implementation, and measuring the time spent on the transmission. If this data is available, the bandwidth may be determined by dividing the length of the transmitted message by the time spent on the transmission. In order to minimize the error, the transmission is repeated several times and the result is averaged. In this case the estimation of the bandwidth usually increases with the increase of the message length going to some maximum value. Usually theis maximum value (or the value obtained when a large message is transmitted) is used as the estimation of the bandwidth.

The time spent on transmitting messages of zero length is usually considered to be the latency.

This Lab describes two test programs: Intel MPI Benchmark (IMB) and tests developed in the Research Computational Center of the Moscow State University (RCC MSU).

Compiling the Benchmark Program

You can download the latest version of the IMB test package included into Intel Cluster Tools from Intel site <u>http://www.intel.com/cd/software/products/asmo-na/eng/cluster/mpi/219848.htm</u>. In order to compile IMB for Microsoft Windows, you will have to create a project in Microsoft Visual Studio 2005 on your own, by the analogy to the projects, which we described in this Lab. You can also use the project framework included into this Lab (the folder IMB_2_3).

Go to <u>http://parallel.ru/ftp/tests/mpi-bench-suite.zip</u> to download the tests developed by the **RCC MSU**. In order to compile the tests for Microsoft Windows, you will have to create a project in Microsoft Visual Studio 2005 on your own. You can also use the project framework included into this Lab (the folder **MGU_tests**).

Running the Benchmarks

The benchmarks should be run on 2 network nodes (one process on each node). Thus, to launch the benchmarks in CCS 2003 it is necessary to specify the total number of processors on 2 network nodes as the job requirements and to choose these nodes manually:

• Open Job Manager. Execute the menu option File->Submit Job... Give the task name and go to the window tab Processors,

ubmit Job Network t	est			×
General Processors	Tasks Licenses	Advanced		
Network test				
Job Name:	Network test			
P <u>r</u> oject Name:				
<u>P</u> riority:	Normal	•		
Submitted By:	N/A			
Submitted on:	N/A			
Status:	Not Submitted			
<u>S</u> ave As Template	e		Submit	Cancel

• Give the total number of processors on the 2 computer nodes, where you are going to execute the benchmarks (for instance, 8 in case you use 2 nodes consisting of 4 processors) as the job requirements. Go to the window tab **Tasks**,

Submit Job Network test
General Processors Tasks Licenses Advanced
Processors required for this job Processors available in this cluster: 60 Minimum required: B Maximum required: 8
Estimate run time for this job Days: Image: Hours: Image: Hours:
Eun 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.
Submit Cancel

• Add the two tasks to the job: "mpiexec -hosts 2 s-cw2-01 1 s-cw2-02 1 \\s-cw-head\temp\imb.exe", "mpiexec -hosts 2 s-cw2-01 1 s-cw2-02 1 \\s-cw-head\temp\MGU_tests.exe" (remember to change the command parameters for those corresponding to your case). The parameter "hosts" has the following format: "n node1 m1 node2 m2 ... noden mn". You cannot use the environment variable CCP_NODES in this case, as only 1 process must be run on each node. Specify the files for redirecting the standard output stream for the tasks. Go to the window tab Advanced,

ob Network test Properties			×
General Processors Tasks L	icenses Advanced		
Task Command Line	Task ssors	m <u>P</u> rocessors:	Add
	asks: s-cw2-01 1 s-cw2 s-cw2-01 1 s-cw2	Processors 8 8	<u>R</u> emove
	Add Parametric	: Sweep	<u>What is it?</u>
Task Summary Name :MGU tests Command Line :mpiexec -hos \MGU_tests.exe Standard Input : Standard Dutput :\\s-cw-hea Standard Error : Work Directory : Number of processors reques RunTime :Infinite Preceding tasks(dependent t	d\temp\MGU_test.txt ted :8	02 1 \\s-cw-head	Ntemp
<u>S</u> ave As Template		OK	Cancel

• Choose the option **Use only these nodes** and tick the nodes, which were given in the command at the previous step. Press the button **Submit** to add the jobs to the queue and enter the name and the password of the user authorized to run jobs on the cluster,

ubmit Jo	b Network test			2
General	Processors Tasks	Licenses Advanced]	
			'	
Node				
Spe	cify the compute nodes	s to use for this job in y	our cluster	
01	<u>U</u> se any available node	s.		
• I	Use <u>o</u> nly these nodes:			
Г	Name	Processors	Speed	RAM 🔺
	S-CW2-01	4	3000	2046
	S-CW2-02	4	3000	2046
	S-CW2-03	4	3000	2046
	S-CW2-04	4	3000	2046
	S-CW2-05	4	3000	2046
	S-CW2-07	4	3000	2046
	S-CW2-08	4	3000	2046
	S-CW2-09	4	3000	2046
	4		2000	
L L	<u> </u>			
Sele	e the allocated nodes <u>e</u> ect this option if your jo e nodes.		ected due to othe	r jobs using the
Save	e As Template		Subr	it Cancel

• The files, where the output stream was redirected to, contain the results of the benchmark execution. It is important to note that IMB carries out a number of various tests, but we are interested only in the first of them, **PingPong**, in order to obtain the network performance parameters. **PingPong** transmits data between two network nodes using the blocking functions *MPI_Send* and *MPI_Recv*, which is optimum for the estimation of the network parameters.

1] Latency: 127.272 microseconds (at 20 times) 4 6.878 8 .065 2 11.79 5 15.71 0 20.11 4 22.67 3 25.65 2 26.35 5 24.59 10 26.89 54 29.62 36 32.4 234.92 36 37.72 50 40.29	IMB_test.txt - Not					
processes = 2 #bytes #repetitions t [Usec] Mbytes/sec 0 1000 130.68 0.01 2 1000 130.68 0.01 4 1000 128.91 0.03 8 1000 127.64 0.06 16 1000 124.83 0.24 32 1000 124.83 0.24 4 4 1000 125.76 0.499 256 1000 125.24 1.95 512 1000 128.76 3.79 1024 1000 129.25 7.56 2048 1000 364.61 15.34 4 5122 1000 364.63 41.45 4096 1000 364.63 41.45 4096 1000 364.63 41.45 153768 1000 364.63 42.70 EULetLt Notepad Edt Format Vew Help atlon 0 1] Latency: 127.272 microseconds (at 20 times) 4 6.878 8 8.065 2 11.79 5 12.102 1000 100 1000 100 1] Latency: 127.272 microseconds (at 20 times) 4 6.878 8 8.065 2 11.79 3 25.61 2 2.67 2 2.67 2 2.67 2 2.67 2 2.68 3 2.4 3 2.72 1 3 2 3.72 1 3 3 3.72 1	ile Edit Format Vie	ew Help				
Fbytes #repetitions t[usec] Mbytes/sec 0 1000 131.62 0.00 1 1000 132.61 0.01 2 1000 132.61 0.01 3 1000 132.61 0.01 2 1000 132.61 0.02 3 1000 127.64 0.06 16 1000 123.77 0.49 32 1000 123.77 0.49 128 1000 127.63 7.95 131 1000 127.78 1.99 236 1000 127.78 1.99 131 1000 127.63 1.425 4096 1000 364.53 11.43 8192 1000 364.53 11.62 131072 320 1993.63 62.70	Benchmarking	PingPong 2				
GU_test.txt - Notepad Edit Format View Help e(b) Transfer (MB/sec) *ration 0 1] Latency: 127.272 microseconds (at 20 times) 4 6.878 8 8.065 2 11.79 5 15.71 0 20.11 4 22.67 3 25.65 2 26.35 3 22.62 38 32.4 10 26.89 4 24.92 36 37.72 50 40.29 4 42.98	0 1 2 4 8 16 64 128 256 512 2048 4096 8192 16384 32768 65536	$\begin{array}{c} 1000\\ 640 \end{array}$	$\begin{array}{c} 131.62\\ 130.68\\ 130.47\\ 128.91\\ 127.64\\ 124.63\\ 124.85\\ 123.77\\ 123.78\\ 125.24\\ 128.76\\ 129.25\\ 137.03\\ 254.61\\ 364.53\\ 365.65\\ 491.61\\ 880.04 \end{array}$	0.00 0.01 0.03 0.06 0.12 0.24 0.49 0.99 1.95 3.79 7.56 14.25 15.34 21.43 42.73 63.57 71.02		
Edit Format View Help e(b) Transfer (MB/sec) ration 0 1] Latency: 127.272 microseconds (at 20 times) 4 6.878 8 8.065 2 11.79 5 15.71 0 20.11 4 22.67 3 25.65 2 26.35 5 24.59 10 26.89 4 29.62 33 22.4 24 42.98		320	1993.63	62.70		
Edit Format View Help e(b) Transfer (MB/sec) ration 0 1] Latency: 127.272 microseconds (at 20 times) 4 6.878 8 8.065 2 11.79 5 15.71 0 20.11 4 22.67 3 25.65 2 26.35 5 24.59 10 26.89 4 29.62 33 22.4 24 42.98						
<pre>ration 0 1] Latency: 127.272 microseconds (at 20 times) 4 6.878 8 8.065 2 11.79 5 15.71 0 20.11 4 22.67 8 25.65 2 26.35 5 24.59 10 26.89 10 26.89 14 22.62 15 34.92 16 32.4 12 34.92 16 37.72 16 40.29 17 42.98 18 42.98 19 42.98 10</pre>						
<pre>ration 0 1] Latency: 127.272 microseconds (at 20 times) 4 6.878 8 8.065 2 11.79 5 15.71 0 20.11 4 22.67 8 25.65 2 26.35 5 24.59 10 26.89 10 26.89 14 22.62 15 34.92 16 32.4 12 34.92 16 37.72 16 40.29 17 42.98 18 42.98 19 42.98 10</pre>	ize(b) Transfe	er (MB/sec)				
1] Latency: 127.272 microseconds (at 20 times) 4 6.878 3 8.065 2 11.79 5 15.71 0 20.11 4 22.67 3 25.65 2 26.35 5 24.59 40 26.89 42 9.62 38 32.4 12 34.92 36 37.72 50 40.29 4 42.98						
8.065 11.79 5 15.71 0 20.11 4 22.67 3 25.65 2 26.35 5 24.59 40 26.89 23 22.62 38 32.4 12 34.92 36 37.72 50 40.29 34 42.98	teration 0 0 1] Latenc	y: 127.272 mic	croseconds (a	at 20 times)		
4 22.67 3 25.65 2 26.35 5 24.59 40 26.89 54 29.62 38 32.4 12 34.92 36 37.72 50 40.29 34 42.98	048 8.065 072 11.79 096 15.71					
JS 44.91	120 20.11 144 22.67 168 25.65 192 26.35 216 24.59 0240 26.89					
	2288 32.4 3312 34.92 4336 37.72 5360 40.29 6384 42.98					

Discussions

- Define the terms "job" and "task". What is the difference between them?
- What basic Microsoft Visual Studio 2005 settings should be specified for compiling a parallel program to be used in the environment MS MPI?
- What are the peculiarities of launching parallel tasks (compiled for MS MPI) on the cluster?
- What is the parametric sweep? What is the workflow?
- What parameters characterizing the network performance do you know? Define them.