Switching and Routing project

How to get started with S&R project

Diego Tuzi

November 2013

Starting with S&R project development

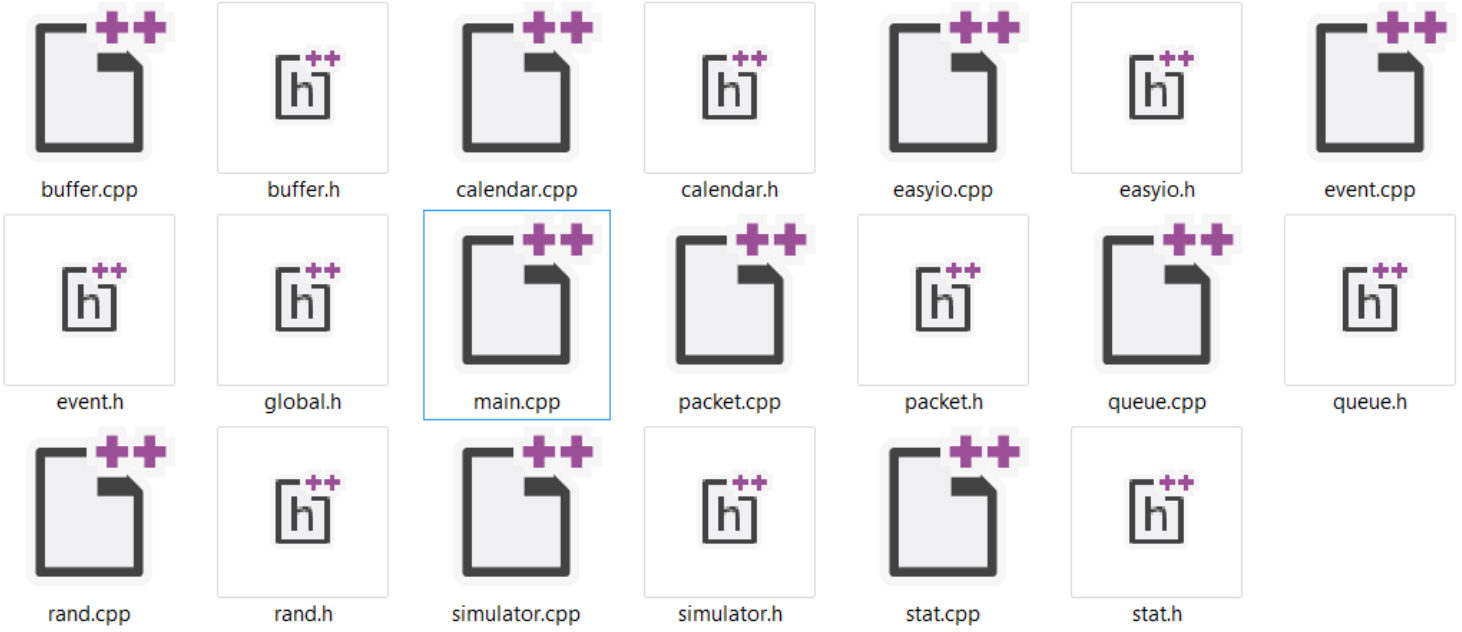
Running simqueue with Visual Studio

Get Visual Studio

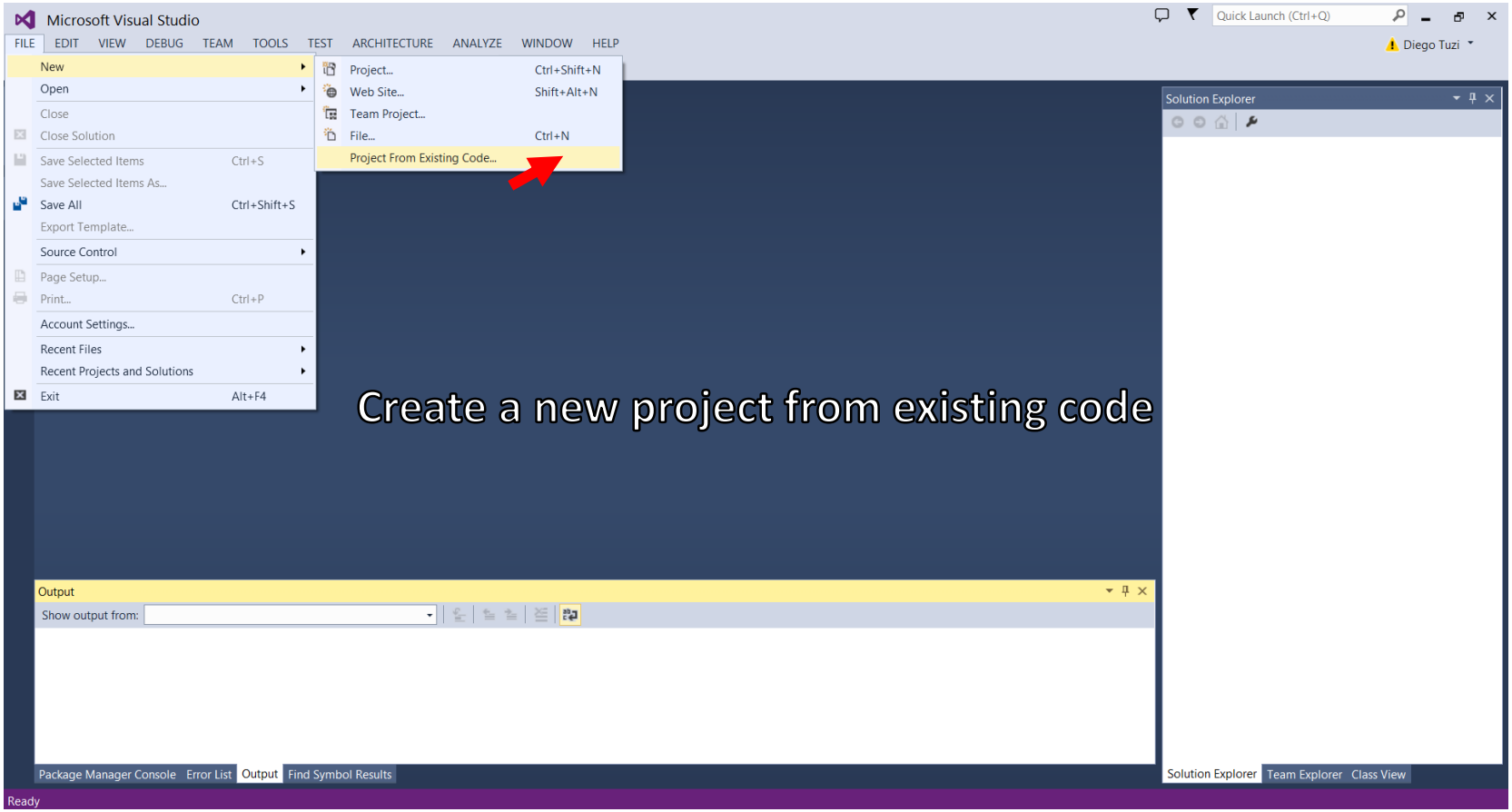
- Full versions from Polimi (Developer Academic Alliance - MSDN AA)
  - *Visual Studio 2013 Professional or Ultimate*
  - http://www.smartpc.polimi.it/en/software-download/students/msdnaa/
- Express version from Microsoft
  - *Visual Studio Express 2013 for Windows Desktop*
  - http://www.visualstudio.com/en-us/downloads/download-visual-studio-vs

Get Simqueue source files

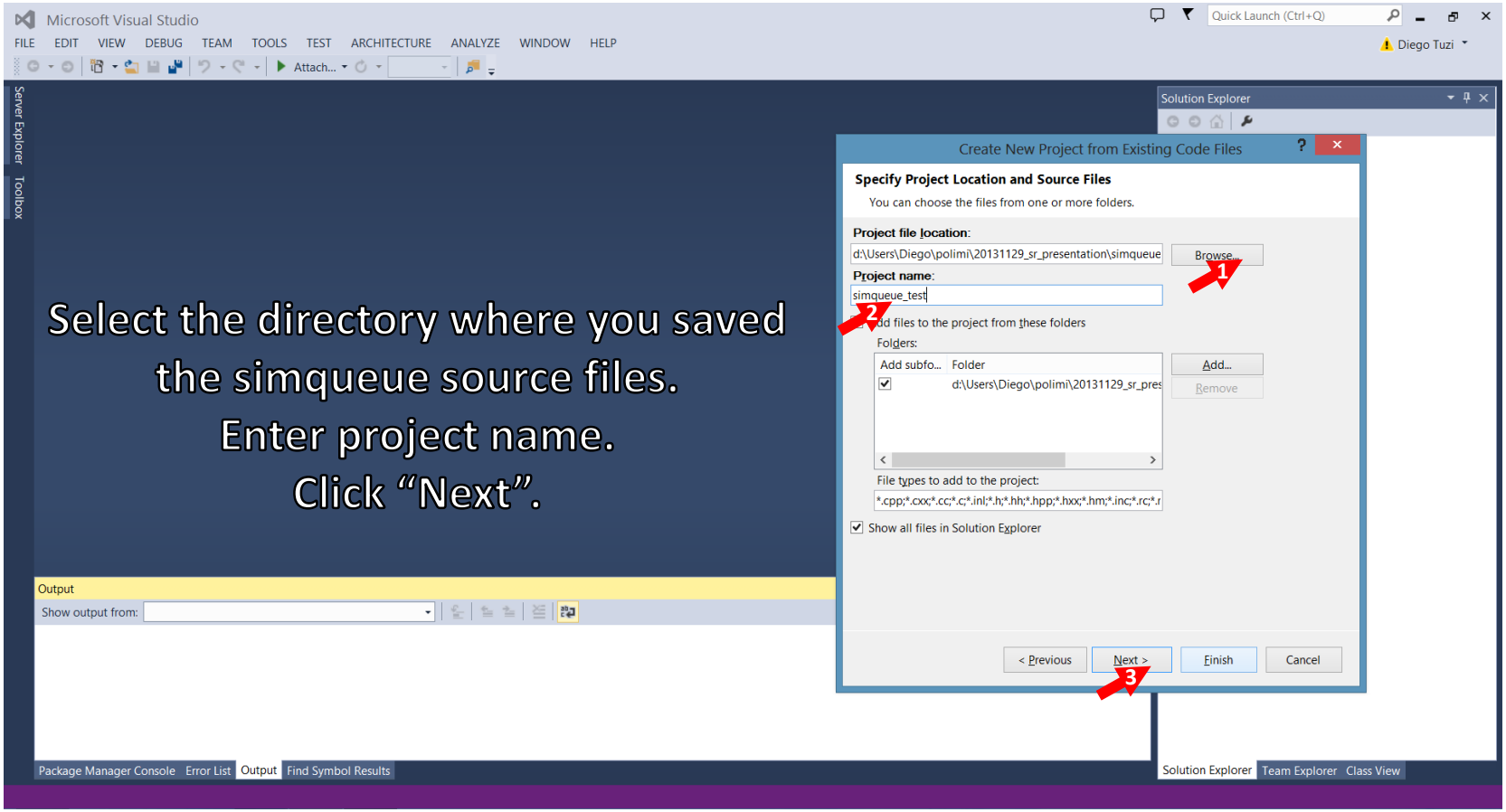
- Download source files from
  - http://home.deib.polimi.it/tornator/Tornatore_files/RCI2010/labo_files/simqueue.zip



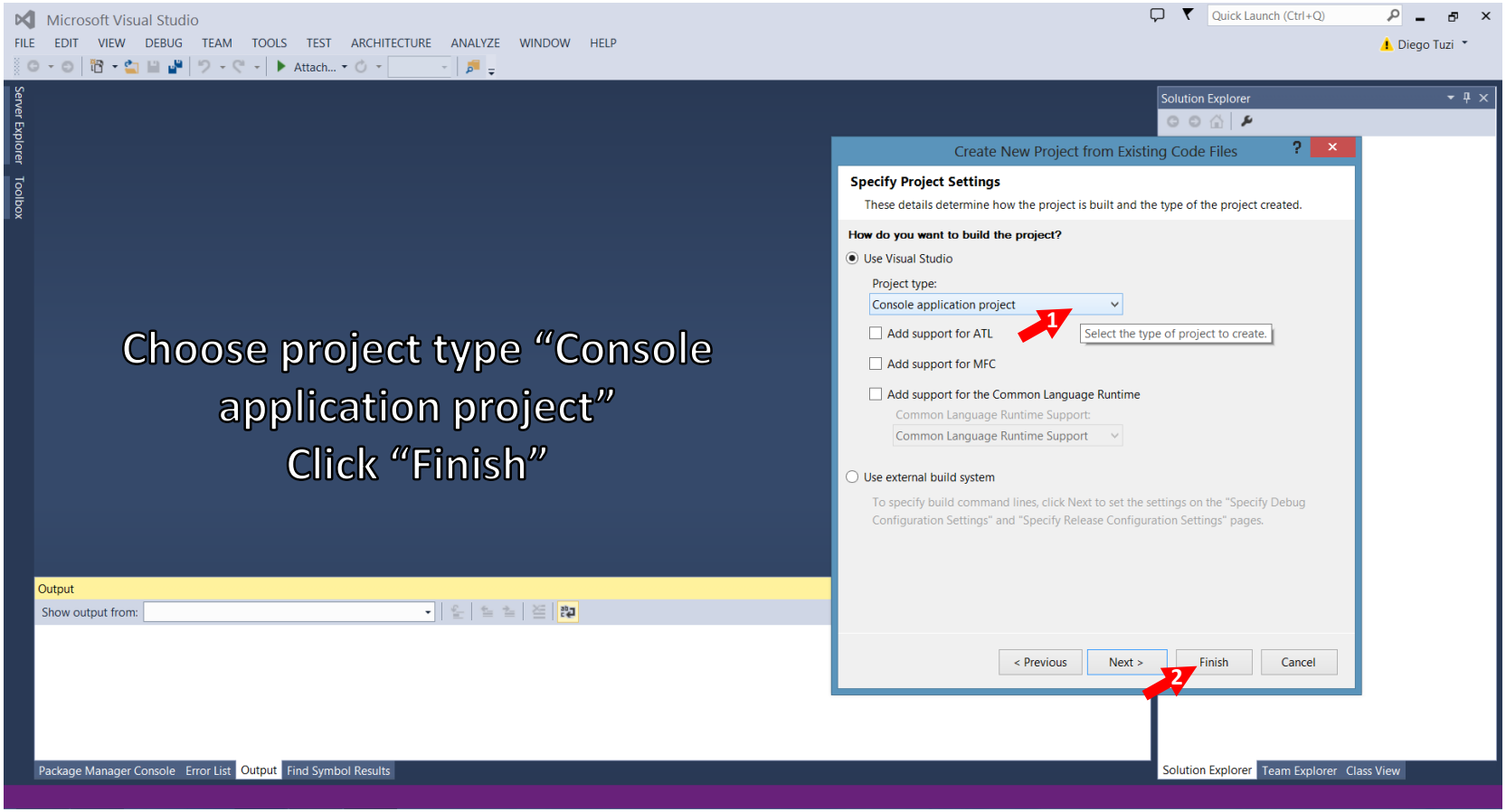
Get started


Microsoft Visual Studio
Quick Launch (Ctrl+Q)
Diego Tuzi
FILE EDIT VIEW DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Source Control
Page Setup...
Print... Ctrl+P
Account Settings...
Recent Files
Recent Projects and Solutions
Exit Alt+F4
Project... Ctrl+Shift+N
Web Site... Shift+Alt+N
Team Project...
File... Ctrl+N
Project From Existing Code...
Create a new project from existing code
Solution Explorer
Output
Show output from:
Package Manager Console Error List Output Find Symbol Results
Solution Explorer Team Explorer Class View
Ready

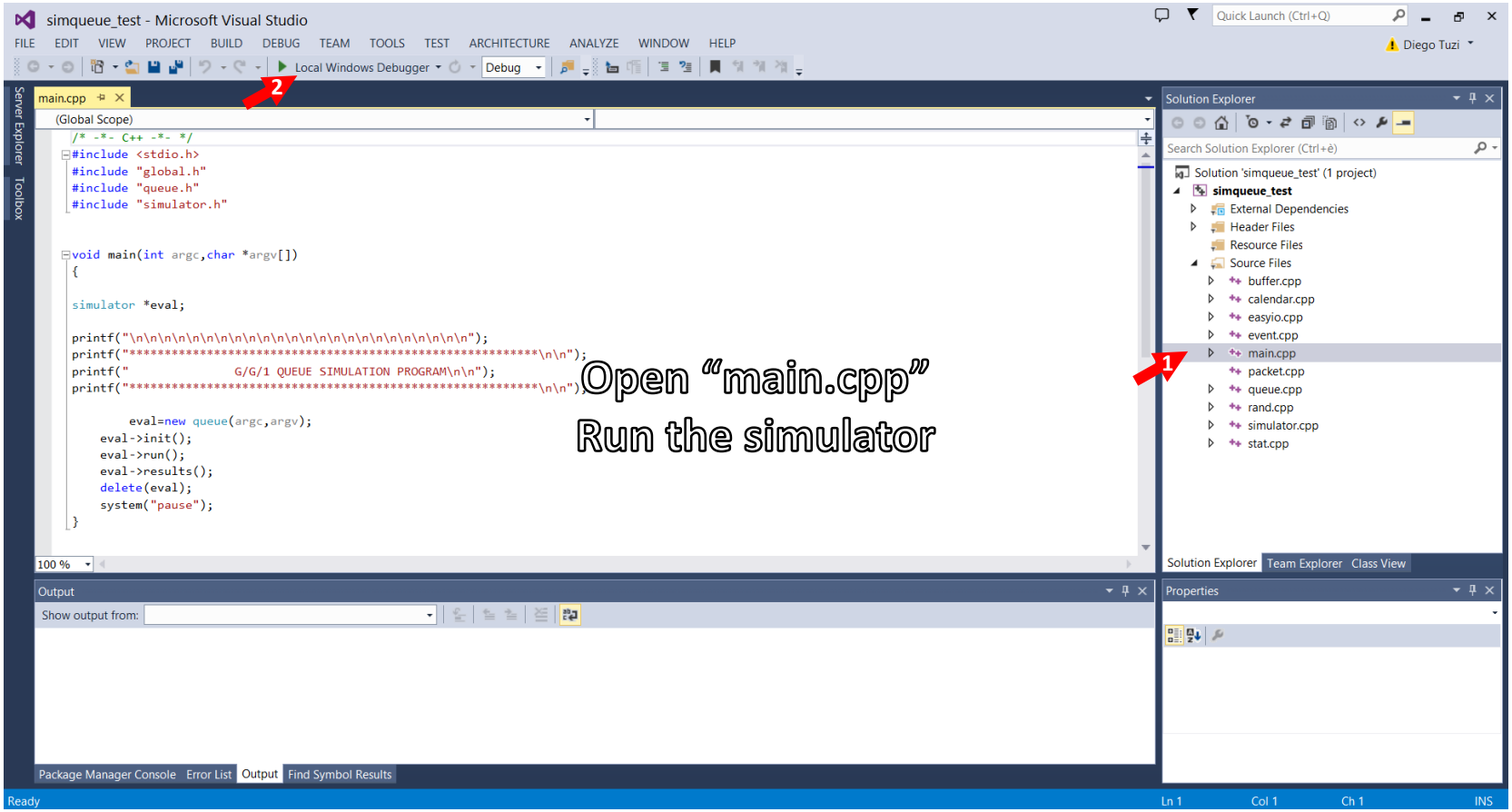
Get started (2)



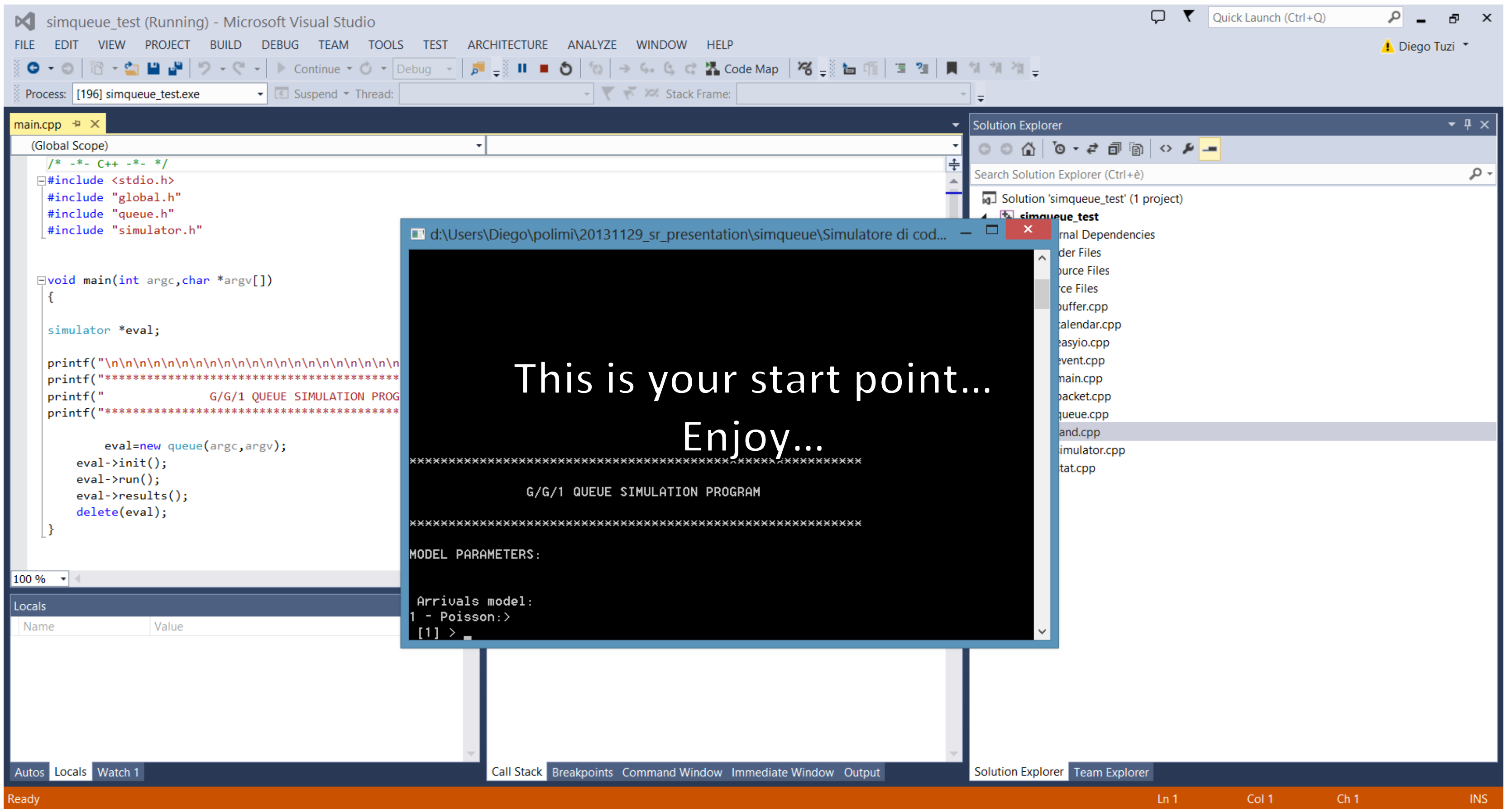
Get started (3)



Get started (4)



Get started (5)

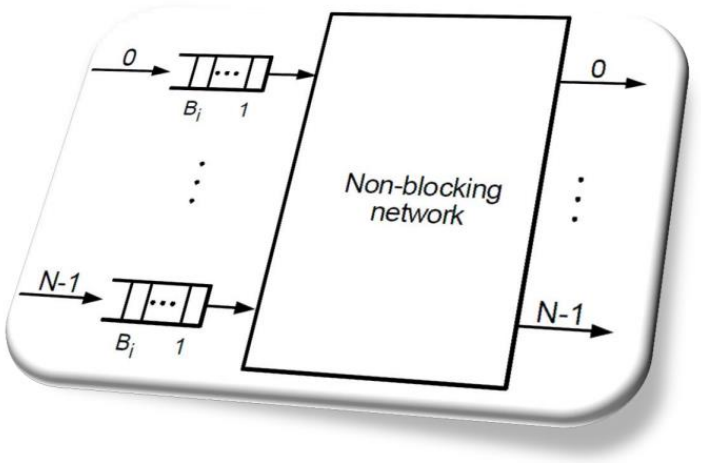


S&R project example

Project developed by Tuzi, Visin, Saliai and Panti in AY 2010/2011

Example of S&R project

- Project name:
  - **Input-buffered switches simulation program**
- Project goal:
  - To reproduce a non-blocking network with input buffer
  - To demonstrate the HOL blocking problem
  - To overcome HOL problem
    - implementing output speed-up
    - implementing different types of advanced scheduling algorithms
  - Performance analysis and comparison



Input parameters

INPUT	DEFAULT	RANGE	DESCRIPTION
N	128	2-256	Number of inputs
M	128	2-256	Number of outputs
Load	1	0.1-1	Traffic load for each input
Input buffer length	8	0-2048	Input buffer length
Output buffer length	8192	0-8192	Output buffer length
Transient accuracy	1000	10-10000	Transient accuracy
Run length	10000	100-20000	Length (in slot) of a single run
Number of runs	10	2-20	Number of runs
Confidence range probability	95	1-100	Confidence range probability in percentage
Output speed-up	1	1-8	Speed-up for each output

Algorithm selection

#	NAME	DESCRIPTION
0	**unfair**	Inputs have different transmission priorities. It suffer from HOL and starvation.
1	**modN**	It has a circular transmission priority. It suffer from HOL but starvation is not possible.
2	**localFifo**	It has a circular transmission priority but contenransmission priority to cell that is in hol position for the longest time. It suffer from HOL.
3	**globalFifo**	There is only one virtual input queue and cells are served in FIFO policy. It is the most fair algorithm. It suffer from HOL.
4	**iRRM**	It implements iRRM algorithm
5	**iSLIP**	It implements iSLIP algorithm
6	**FIRM**	It implements FIRM algorithm

Output values

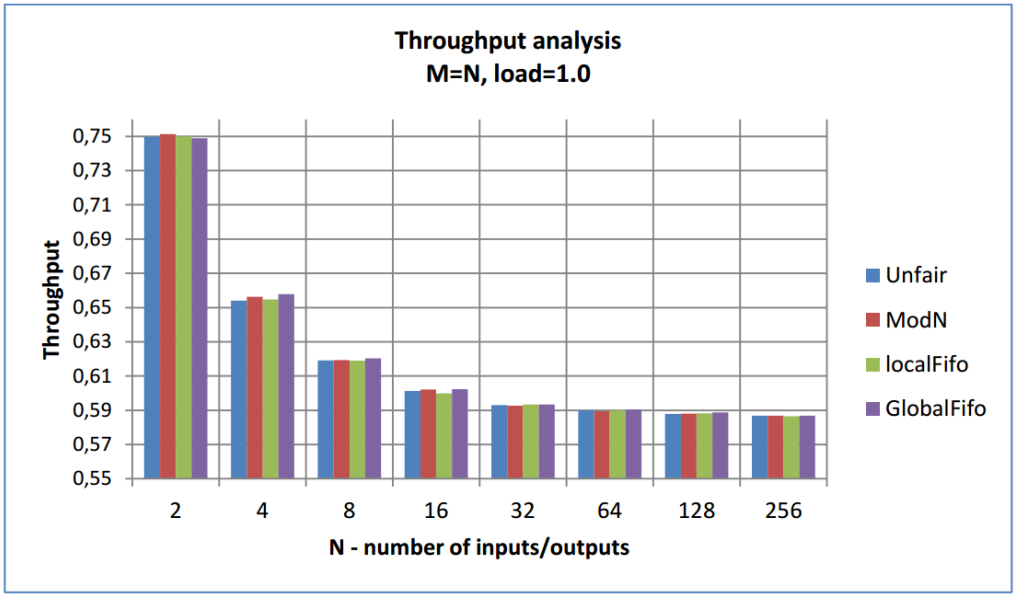
NAME	DESCRIPTION
Average input delay	Mean delay time a cell spent into input buffer
Average output delay	Mean delay time a cell spent into output buffer. It is different from zero only if output speed-up is greater than one
Average total delay	Mean total delay time
Average lost delay	Mean value of lost packets
Throughput	Mean throughput value
Input packet loss probability	A packet is loss if the input buffer is busy.
Average iterations	Mean number of iterations needed by advanced scheduling algorithms to obtain the best input-output matching (in one slot time)

Performance analysis

Base algorithm: throughput

- Demonstration of the HOL blocking problem
- 58,6 % throughput limit when $N = M \rightarrow \infty$
- Every single value in the table is the output of one simulation session

N	Theorical	unfair	modN	localFifo	globalFifo
2	0.7500	0.7497	0.7513	0.7499	0.7490
4	0.6553	0.6540	0.6563	0.6547	0.6578
8	0.6184	0.6191	0.6193	0.6190	0.6203
16	-	0.6013	0.6022	0.5999	0.6024
32	-	0.5930	0.5927	0.5933	0.5933
64	-	0.5900	0.5894	0.5899	0.5905
128	-	0.5879	0.5880	0.5883	0.5887
256	-	**0.5868**	**0.5869**	**0.5865**	**0.5869**
∞	0.5858	-	-	-	-

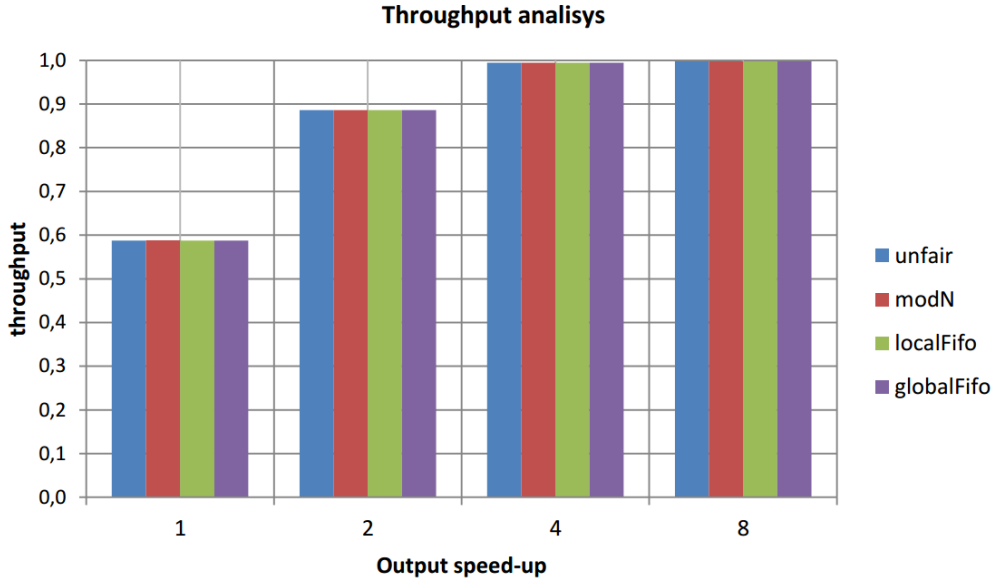


Performance analysis

Base algorithm: throughput with output speed-up

- To overcome the HOL problem one solution is to increase the ratio between output and input or equivalently using **output speed-up**.

K	Theorical	unfair	modN	localFifo	globalFifo
1	0.586	0.5877	0.5878	0.5875	0.5876
2	0.885	0.8861	0.8863	0.8862	0.8862
4	0.996	0.9947	0.9947	0.9947	0.9947
8	1.000	0.9976	0.9976	0.9976	0.9976

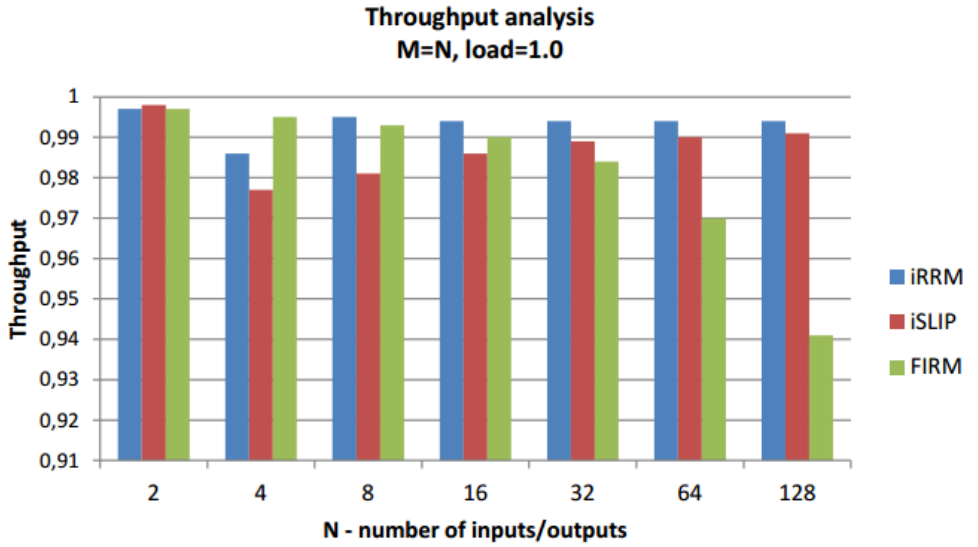


Performance analysis

Advanced algorithm

- To fix HOL blocking problem without change the ratio between number of output and number of input is to use advanced scheduling algorithm.

N	iRRM	iSLIP	FIRM
2	0.997	0.998	0.997
4	0.986	0.977	0.995
8	0.995	0.981	0.993
16	0.994	0.986	0.99
32	0.994	0.989	0.984
64	0.994	0.99	0.97
128	0.994	0.991	0.941



More details

- You can download the template project realized in these slides and the complete “Input-buffered switches simulation program” project from the following link:
  - https://www.dropbox.com/s/s2pv1qa2p9vkkor/20131129_sr_presentation.zip