

# Intel ® Math Kernel Library

Авторы:

к.т.н. А.А.Романенко  
проф. д.ф.-м.н. Ю.М.Лаевский  
к.ф.-м.н. С.В.Гололобов  
к.ф.-м.н. А.А.Калинкин

# О чем курс

- Что такое MKL
- Где и для чего можно применять MKL
- Сборка программ
- Функции в MKL
- Параллелизм в MKL
- Примеры использования

# Структура курса

- 11 лекций
- 11 практических занятий
- По окончании курса предлагается пройти тест для получения сертификата
- 5 вопросов. Проходной балл - 3

# Знакомство с МКЛ

## Лекция 1

# Поддерживаемые ОС

- Linux\*
- MS Windows\*
- MacOS\*



\* - Конкретные версии ОС смотри в документации

# Поддерживаемые компиляторы

- Intel ® Fortran compiler
- Intel ® C/C++ compiler
- MS Visual Studio \*
- MS Windows SDK \*
- GNU Compiler collection \*
- PGI \*
- Absoft ® Pro Fortran
- ...

\* - Конкретные версии компиляторов смотри в документации

# Почему именно MKL

- Производительность!
- Вычисления
- Разработка
- Поддержка функций
- BLAS (level 1, 2, 3)
- LAPACK
- FFT
- пр.
- Оптимизирована под процессоры Intel®

# Области применения

- Энергетика
- Экономика
- Химия
- Вычислительная математика
- Физика
- Информатика
- Биология
- Медицина
- пр.



# Компоненты MKL

- BLAS
- Sparse BLAS
- LAPACK
- ScaLAPACK
- DFT
- Cluster DFT
- Sparse Solvers (PARDISO, DSS и ISS)
- VML (Vector Math Library)
- VSL (Vector Statistical Library)
- PDEs (Partial Differential Equations)
- Optimization Solvers
- GMP

# Сборка программ

- <Компилятор>

<files to link> -L<MKL path> -I<MKL include>

[-I<MKL include>/{ia32|intel64|{ilp64|lp64}}]

[-lmkl\_blas{95|95\_ilp64|95\_lp64}]

[-lmkl\_lapack{95|95\_ilp64|95\_lp64}]

[<cluster components>]

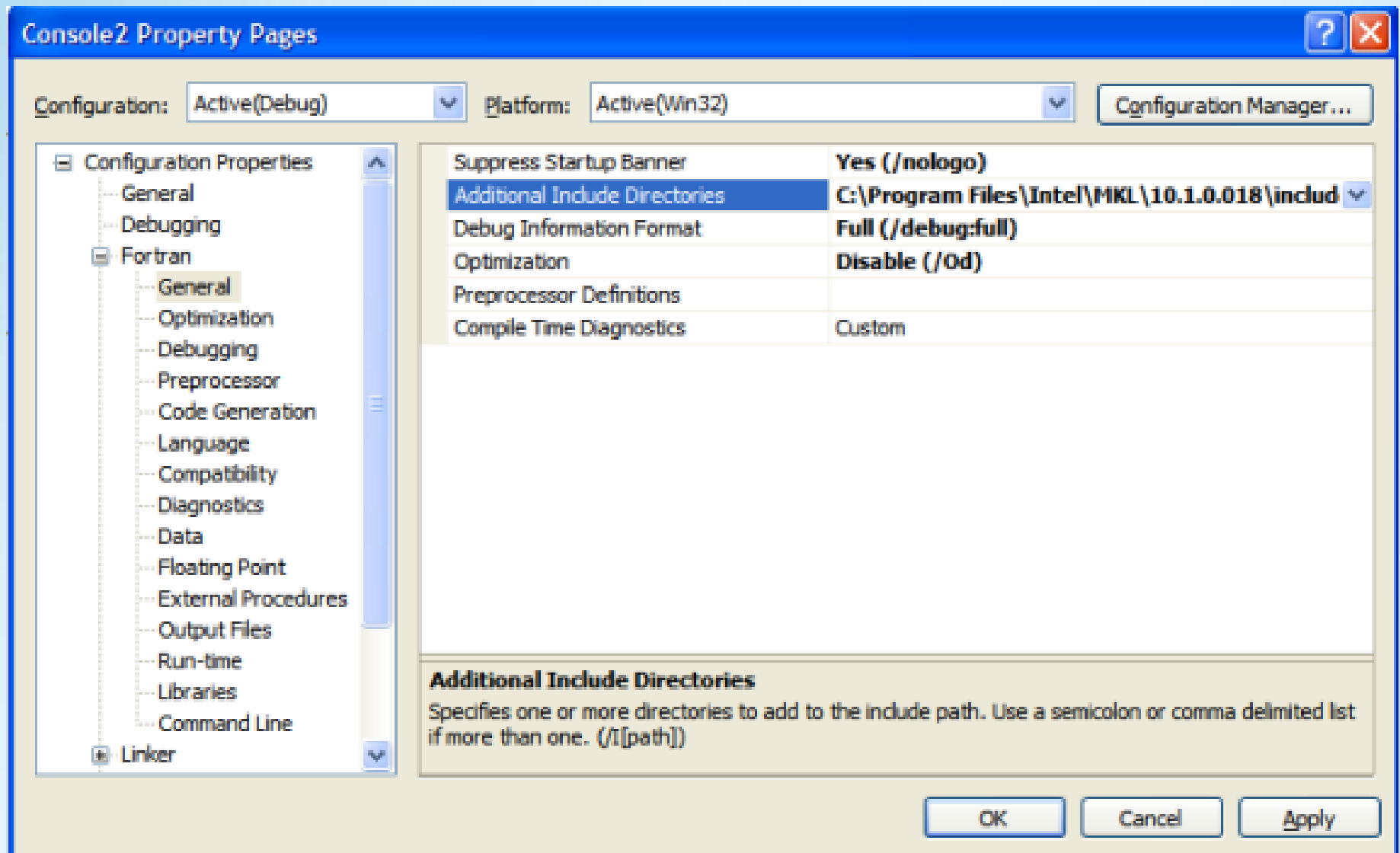
-lmkl\_{intel|intel\_ilp64|intel\_lp64|intel\_sp2dp|gf|gf\_ilp64|gf\_lp64}

-lmkl\_{intel\_thread|gnu\_thread|pgi\_thread|sequential}

-lmkl\_core

-liomp5 [-lpthread] [-lm]

# Настройка MS Visual Studio



# Console2 Property Pages



Configuration: Active(Debug) ▼

Platform: Active(Win32) ▼

Configuration Manager...

- [-] Configuration Properties
  - General
  - Debugging
  - [+] Fortran
  - [-] Linker
    - General
    - Input
    - Manifest File
    - Debug
    - System
    - Optimization
    - Embedded IDL
    - Advanced
    - Command Line
  - [+] Resources
  - [+] MIDL
  - [+] Manifest Tool
  - [+] Build Events
  - [+] Custom Build Step

Output File	\$(OutDir)\\$(ProjectName).exe
Show Progress	Not Set
Version	
Enable Incremental Linking	No (/INCREMENTAL:NO)
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Additional Library Directories	C:\Program Files\Intel\MKL\10.1.0.018\ia32\lib\
Link Library Dependencies	Yes

**Additional Library Directories**  
Specifies one or more additional paths to search for libraries; configuration specific; use semi-colon delimited list if more than one. (/LIBPATH[:dir])

OK Cancel Apply

# Console2 Property Pages



Configuration: Active(Debug) ▼

Platform: Active(Win32) ▼

Configuration Manager ...

## Configuration Properties

- General
- Debugging
- Fortran
- Linker
  - General
  - Input**
  - Manifest File
  - Debug
  - System
  - Optimization
  - Embedded IDL
  - Advanced
  - Command Line
- Resources
- MIDL
- Manifest Tool
- Build Events
- Custom Build Step

Additional Dependencies	<b>mkl_c.lib ; libguide40.lib</b>
Ignore All Default Libraries	No
Ignore Specific Library	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded DLLs	

### Additional Dependencies

Specifies additional items to add to the link line (ex: kernel32.lib); configuration specific.

OK Cancel Apply

# Или MS Visual Studio + Composer XE

The image shows a screenshot of Microsoft Visual Studio. The main window displays the source code for `ChildFrm.h`. The code is as follows:

```
// ChildFrm.h : interface of the CChildFrame class
//
#pragma once

class CChildFrame : public CMDIChildWndEx
{
    DECLARE_DYNCREATE(CChildFrame)
public:
    CChildFrame();

    // Attributes
public:

    // Operations
public:

    // Overrides
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);

    // Implementation
public:
    virtual ~CChildFrame();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

    // Generated message map functions
protected:
    DECLARE_MESSAGE_MAP()
};
```

The Solution Explorer on the right shows a project named 'test'. A context menu is open over the project, with the following items:

- Build
- Rebuild
- Clean
- Project Only
- Calculate Code Metrics
- Profile Guided Optimization
- Custom Build Rules...
- Tool Build Order...
- Add
- References...
- Add Web Reference...
- View Class Diagram
- Set as StartUp Project
- Debug
- Cut
- Paste
- Remove
- Rename
- Unload Project
- Open Folder in Windows Explorer
- Select Build Components
- Intel C++ Composer XE 2011
- Properties

Two red circles highlight the Intel C++ Composer XE 2011 icon in the top toolbar and the 'Intel C++ Composer XE 2011' menu item in the context menu.

test - Microsoft Visual Studio

File Edit View Project Build Debug Data Tools Test Analyze Window Help

Debug Win32

ChildFrm.h Start Page

(Global Scope)

test - Select Build Components

Project Configuration: Debug Project Platform: Win32

Suite to use with this configuration: <Standalone Libraries>

Intel Performance Libraries to use with this configuration

Performance Libraries Versions: Default versions:

IPP: <Not installed>

TBB: <Not installed>

MKL: <Default> Intel(R) Math Kernel Library 10.3.1

Intel Integrated Performance Primitives

Use IPP:

Intel Math Kernel Library

Use MKL:  (dropdown menu: No, Parallel, Sequential, Cluster)

Intel Threading Building Blocks

Use TBB:

Instrument for use with Intel® Threading Analysis Tools:

OK Cancel Apply

protected:  
DECLARE\_MESSAGE\_MAP()  
};

Ready

Solution Explorer - test

Solution 'test' (1 project)

test

les  
frm.h  
view.h  
ew.h  
frm.h  
utWnd.h  
ertiesWnd.h  
ource.h  
x.h  
tver.h  
n  
oc.h  
ew.h  
ree.h  
Files  
es  
frm.cpp  
view.cpp  
ew.cpp  
frm.cpp  
utWnd.cpp  
ertiesWnd.cpp  
x.cpp  
pp  
oc.cpp  
ew.cpp  
ree.cpp  
xt

Solution Explorer Class View



# Среда выполнения

- Переменные окружения MKL
- MKL\_NUM\_THREADS  $\Leftrightarrow$  OMP\_NUM\_THREADS
- MKL\_DYNAMIC  $\Leftrightarrow$  OMP\_DYNAMIC
- MKL\_DOMAIN\_NUM\_THREADS
- MKL\_THREADING\_LAYER
- MKL\_INTERFACE\_LAYER

# Документация и примеры

- Документация
  - Документация пользователя (User guide)
  - Справочные страницы (Manual pages)
- Примеры
  - C\C++, Fortran
  - Входные и выходные данные
  - Сборка или из командной строки (makefile) или в среде разработки.
  - Для сборки необходимо три файла
    - Сам пример,
    - Файл описания общих для всех примеров функций и
    - Файл реализации этих функций

# Дополнительная информация

- Информация по МКЛ (на английском) находится по ссылке
- <http://software.intel.com/en-us/articles/intel-mkl/>
- Подключение к MS Visual Studio
- <http://software.intel.com/en-us/articles/intel-math-kernel-library-intel-mkl-compiling-and-linking-with-microsoft-visual-c-c/>