11/25/15

Multithreading with ITK

Keep MITK running!

Jonas Cordes

Multithreading classes

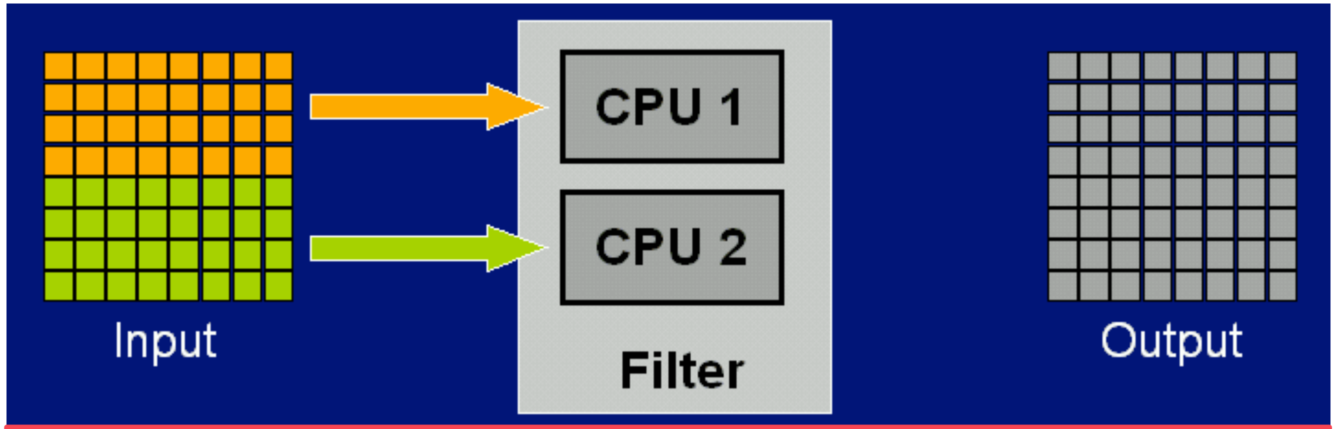
- Filter/Application Level:
  - itk::ImageToImageFilter
  - itk::DomainThreader
  - itk::MultiThreader
- GUI Level:
  - QConcurrent and QFuture
  - QFutureWatcher (Asynchronous)

→BS Vortrag von Peter & Andi
→http://www.mitk.org/wiki/Bug_Squashing_Seminars

ITK ImageToImageFilter

- Implement ThreadedGenerateData instead of GenerateData
- A superclass will spawn several threads
    - (Usually) matching the number of processors in the system

```
void ThreadedGenerateData(const OutputImageRegionType &
outputRegionForThread, ThreadIdType threadId);
```



- ThreadedGenerateData method must be Thread-Safe!!!

ITK ImageToImageFilter

- *Thread-local storage*
  - BeforeThreadedGenerateData() → initialize storage for each thread
  - ThreadedGenerateData(…) → do the work!
  - AfterThreadedGenerateData() → merge storages

- *Image region iterators*

```
void ThreadedGenerateData(const OutputImageRegionType & outputRegionForThread, ThreadIdType threadId)
{
        auto mit = ImageRegionConstIterator< InputImageType >(this->GetInput(),
        outputRegionForThread);
        auto oit = ImageRegionIterator< OutputImageType >(this->GetOutput(),
        outputRegionForThread);
        while(!mit.IsAtEnd()) … do something
}
```

ITK DomainThreader

```
template< typename TDomainPartitioner, typename AssociateType >
class DomainThreader: public Object
{
public:
void Execute( AssociateType * enclosingClass, const DomainType & domain );
protected:
virtual void BeforeThreadedExecution(){}
virtual void ThreadedExecution( const DomainType& subdomain, const
ThreadIdType threadId ) = 0;
virtual void AfterThreadedExecution(){}

AssociateType * m_Associate;
};
```

ITK DomainThreader - Partitioner

- Abstract class itk::ThreadedDomainPartitioner<TDomain>
  - *PartitionDomain*(const ThreadIdType threadId, const ThreadIdType requestedTotal, const DomainType& completeDomain, DomainType& subDomain) const = 0;
- itk::ThreadedImageRegionPartitioner<ImageRegion<VDimension>>
  - Splits an image space into several “blocks”
  - Can handle arbitrary image dimensions
- itk::ThreadedIndexedContainerPartitioner<Index<2>>
  - Splits an index range into several pieces
  - Handles residuals

ITK MultiThreader

- Can handle multiple methods
- Is used by DomainThreader

```
void SingleMethodExecute(); ← blocking call
void SetSingleMethod(ThreadFunctionType, void *data);

void MultipleMethodExecute(); ← blocking call
void SetMultipleMethod(ThreadIdType index, ThreadFunctionType, void *data)

static ITK_THREAD_RETURN_TYPE ThreaderCallback( void *arg ){
        typedef itk::MultiThreader::ThreadInfoStruct ThreadInfoType;
        ThreadInfoType * infoStruct = static_cast< ThreadInfoType * >( arg );
        ProcessData * data = (ProcessData *)(infoStruct->UserData);
        …
}
```

Discussion

- Questions?
- OpenMP policy?

  → If possible avoid using it!

- http://insightsoftwareconsortium.github.io/ITKBarCamp-doc/ITK/WriteMultiThreadedCode/index.html

Auf Wiedersehen im DKFZ!

Weitere Informationen unter www.dkfz.de

DEUTSCHES KREBSFORSCHUNGSZENTRUM IN DER HELMHOLTZ-GEMEINSCHAFT

50 Jahre – Forschen für ein Leben ohne Krebs