Improving the scalability of an environmental modelling framework to allow for large-scale high-resolution geosimulations

Oliver Schmitz, Kor de Jong and Derek Karssenberg

Introduction
The high computational requirements for stochastic spatio-temporal modelling, and an increasing demand to run models over large areas at high resolution, e.g. in global hydrological modelling or epidemiology, require an optimal use of available, heterogeneous computing resources. Domain-specific modelling software used by environmental scientists, however, often do not provide built-in capabilities to distribute model runs over the compute nodes of a supercomputer. We propose to enhance the PCRaster model building framework with built-in capabilities to run models on various hardware platforms, resulting in hardware scalable models that can be constructed by environmental modellers.

The PCRaster modelling framework
The PCRaster modelling framework
- Is targeted at the development of spatio-temporal models
- Fast model development and execution
- Scripting environments: PCRcalc and Python
- Rich set of model building blocks for manipulating raster maps
- Framework for data assimilation
- Tool for visualisation of spatio-temporal stochastic data
- Framework for stochastic spatio-temporal model building
- Targeted at the development of spatio-temporal models
- Is open source and can be downloaded for free
- Can be run on Linux, Microsoft Windows and Apple OS X

Stochastic spatio-temporal modelling
Model
\[ \mathbf{z}_t = \left\{ \mathbf{z}_{t-1}, \mathbf{i}, \mathbf{p} \right\} \] 
for all time steps \( t = 1, \ldots, T \)

Solution scheme
for each \( n \) in Monte Carlo samples:
for each \( t \) in time steps:
\[ \mathbf{z}_t^{(n)} = \mathbf{f}(\mathbf{z}_{t-1}^{(n)}, \mathbf{i}^{(n)}, \mathbf{p}^{(n)}) \]

Building blocks
- discharge = kinematicFlowDir(precipitation,...)
- result map = spatial function (input maps)
- agent support field = support field
- field support IO libraries = low level libraries
- domain specific libraries code = general public libraries code

Solution framework (Python)
```python
for each n in Monte Carlo Samples:
    for each t in time steps:
        z_t(n) = f(z_{t-1}(n), i^{(n)}, p^{(n)})
```

Data assimilation
```
period: model variable: time
update model state when observations are available using Bayes' theorem
```

Solution scheme
for each \( n \) in Monte Carlo Samples:
for each \( t \) in period:
\[ z_t^{(n)} = f(z_{t-1}^{(n)}, i^{(n)}, p^{(n)}) \]

Solution framework (Python)
```python
def setup(self):
    self.report(self, snow, 's')
def update(self):
    return weight

def resume(self):
    self.read('s')
```

PCRaster on shared memory systems
A binding between PCRaster and Fern provides about 50 parallel local and focal operations. Fern is a is a highly generic C++ software library for raster processing that can be tailored to the configuration of a modelling framework.

Policy
- Behaviour
  - Execution: kind of parallelism
  - Input no-data: how to handle no-data in input
  - Output no-data: how to handle no-data in output
  - Out-of-range: how to handle out-of-range values in input
  - Out-of-range: how to handle out-of-range values in algorithm result

Configuration options per algorithm
- Excerpt of C++ implementation of square root
- Operation as used by modeller

PCRaster on distributed memory systems
Algorithms that operate on an irregular topology, such as material transport over a local drainage network, require a decomposition into fine grained sets of concurrent tasks for efficient execution. These tasks will be connected with other tasks from multiple algorithms into a task-graph, and an external HPX runtime library executes all tasks both on shared and variable distributed memory systems.

Download and further information
http://www.pcraster.eu
https://github.com/geoneric/fern
http://stellar.cct.lsu.edu/