model.bricks

Model Bricks:

Fast moving, volatile markets are putting today's risk management applications under great stress, as risk measures need to constantly be re-computed for large derivatives books. Often, by the time risk measures have been refreshed, they are invalidated as markets have by then, already moved by large amounts. Increasingly, many market participants must manage risk by instinctive approximation; in such markets, timeliness IS usefulness, information delayed IS information denied.

EigenSystems presents Model.Bricks, the analytics framework of Red.Shift that delivers continuous real-time risk management, even for very large portfolios.

Constraints of inflexible software and slow calculation times are no longer a concern – the unique flexibility and speed of Model.Bricks allows you to focus your intellectual assets on designing stress scenarios, analyzing results and formulating risk management strategy and policy. Designed from the ground up for performance on multi-core processors and compute grids, Model.Bricks makes extremely efficient use of hardware. The fast parallel algorithms at its core are designed to discover and exploit available CPU cores to the fullest.

Model.Bricks is the analytics framework used for valuation and risk measurement within Red.Shift, our risk management application. Like a number of core Red.Shift components, it can be used independently outside of the Red.Shift framework as an Excel add-in, as well as with Ruby bindings in customized spreadsheets or scripts. As a result, consistency between spreadsheets and mainframe risk management systems is no longer a distant goal but a reality.

Model.Bricks can optionally be used with Eigen. Spaces, a library developed by Eigen.Systems that unifies symmetric multiprocessing (multi-core computing) with grid computing. The result is the delivery of a highly parallel and fault tolerant risk computation engine with low latency market making / algorithmic trading platforms.

Model.Bricks delivers this radical performance using a unique design that assembles valuation algorithms from a small set of highly optimized building blocks ("bricks").The bricks hold state information that is retained from one computation to the next, eliminating the need to re-compute partial and intermediate results that have been previously calculated. Great care is taken to ensure that this does not result in over-use of memory by minimizing context switching in the on-chip CPU cache.

Exceptional flexibility is delivered by designing these bricks to be elements that can be used to construct a wide variety of analytic models.The result is an unsurpassed ability to achieve 'fine-grain' parallelism using multi-core processes, exceeding what can be achieved by traditional grid computing methods hundreds of times over.