Scaling Research

In Computational Finance | May 5, 2015

David Lin, Managing Director
# Table of Contents

**Background**
- Empirical Issue Observation (Buy-side Research)
- Scope of Discussion
- Common Techniques
- Existing Tools

**Challenges**
- Visualization
- Computation (Focus)
- Data

**Solution**
- Scaling Research: Transition Research (Desktop) to Production (Server-side)
- Parallel Compute: Leveraging MATLAB® Production Server™ (MPS)
- Empirical Issue Resolution

**Q & A**
Empirical Issue Observation (Buy-side Research)

- **Factor Analysis**
  - Research Activity: Return decomposition (>100 factor-dimension)
  - 1 time-series decomposition (desktop MATLAB) returns < 60 Sec
  - Decom of 1,000-10,000 return time-series “does not” complete
  - Long-run processes prone to “failures” and take time to “repeat”

- **Portfolio Construction**
  - Research Activity: Optimization-driven Security-Selection
  - 1 Portfolio Construction (desktop MATLAB) returns in time
  - Optimize 500 Portfolio “does not” complete in time
  - Long-run processes prone to “failure”
  - Portfolio Construction frequently “misses” Market
Scope of Discussion (Buy-side Research)

- **Alpha Modeling**
  - Bottom-up Research - Investible Universe Selection, Ranking and Pairing
  - Top-down Research - Asset Allocation by:
    - Region
    - Country
    - Sector
    - Factors
    - Asset Classes
  - Currency Overlay Algorithm
  - Exotic Alpha signal generation

- **Risk Modeling**
  - Fundamental Models
  - Macro Models
  - Statistical Models

- **Tax Efficiency Research**

- **Transaction Cost modeling**
  - Linear/Non-Linear Models
  - T-Cost Estimates (Pre-Trade)
  - T-Cost Analysis (Post-Trade)
  - Real-time Trading signal Generation
Scope of Discussion (Buy-side Research)

- **Portfolio Construction**
  - MVO
  - Robust
  - Black-Litterman

- **Scenario & What-if Analysis**

- **Back-test, supporting:**
  - Multi-Asset
  - Off-Cycle
  - Market-Neutral
  - Total-Return

- **Stress-test, supporting:**
  - Macro Scenario
  - Fundamental Per-security Scenario

- **Performance Contribution & Attribution:**
  - Region/Country
  - Sector
  - Asset Class
  - Factor
Techniques

Large scale:

1. Simulations (including flavors of Monte Carlo)
2. Regressions
3. Optimizations
4. Exotic, Big and Unstructured Data collection
5. Data Analysis of #4, with AI capability
6. High-speed transactional and market data collection
7. Data Analysis of #6
8. Historical time-series data collection (Analytics)
9. Data Analysis of #8
10. Back-Test
11. Stress-Test
Existing Tooling

Desktop-driven Research (common):

- Terminals:
  - Bloomberg/FactSet/MorningStar/Lipper/etc

- A flavor of the following compute tools (desktop)
  - Excel
  - MATLAB
  - SAS
  - R/Revolution
  - Stata
  - Mathematica
  - C/C++
  - Python

- A flavor of the following databases
  - Access
  - Mysql
  - PostgreSQL
  - SQL Server
Challenges

Desktop-driven Research faces challenge

- Lack of Advanced visualization
  - Highly customizable UI
  - Cutting-edge visual elements
  - Speed and Performance
  - Portable UI

- Lack of Enterprise Data Platform
  - Enterprise Data Platform
  - Big and Unstructured Data

- Lack of Scalable computational platform, specifically:
  - In-Engine Data Usage, lack of:
    - Parallel Data extraction
    - Large Scale and Distributed Data caching
  - Desktop-based Computational engine – scale challenged:
    - Some lack of threading capability
    - Most lack of horizontal scalability
Scaling Research (generalized solution)

- Transition from Research (Desktop) to Production (Server-side)
- String it together:

**Research Activity:**
- Desktop:
  - MATLAB (Compute)
  - Terminals (Analytics Data)
  - Database (User Data)

**Production Activity:**
- Desktop:
  - Visualization (Advanced UI)
- Compute Server(s):
  - MPS (Compute)
  - Data Preparation Service Provider
- Data Server(s):
  - Analytic Data
  - User Data
Scaling Research (generalized solution)

Diagrammatic View:

UI Client [visualization]

Non-UI Client [Batch/Report]

Parallel-able Server(s)

Service Proxy

Computing Component

Computing Model
- Optimization
- Regression
- Simulation
- ... AI Analysis

Data Component
- Cache 1
- Cache 2
- ... Cache N

Computation Engine Abstraction

Platform Computing (IBM)
Graphic Process Unit (GPU)
MATLAB Prod Server (MPS)

Internal DBs
External DBs
Scaling Compute (generalized solution)

- Transition from Desktop to Server-side “Compute”
- How: Separation of “Computation Infrastructure” from “Engine”

  - Compute Infrastructure (Container)
    - Data retrieval and caching - scalable and robust
    - Data preparation to/from compute engine
    - Data servicing to external client
    - Front-to-back exception-handling
    - Horizontal scaling facility
    - HA and DR facility

  - Compute Engine
    - Functional area interface abstraction - implementation varies
    - Pure engine decoupling data retrieval from compute
    - Scaling consideration specific to computation challenge
“Decomposable” Computation Task:

- Substantial “compute task” can be sub-divided into repetitive “compute unit”
- Sub “compute unit” are independently stateless
- Use solutions like “Platform Computing” (IBM) to farm out jobs
- Compute Container interfaces with underlying Compute Farm
Computation Engine – Selection

“Indivisible” Computation Task:

- MATLAB® Production Server™ - MPS is a practical solution
- Intuitive Math and Statistical expression and construct
- Pre-build financial concept/function and wide-adoption
- Professionally supported scaling infrastructure (multi-process and expandable)
- “Client-side connectivity” provided to interact with Compute Container
- Smooth migration from “Research Activity” to scalable “Production Activity”
Computation Engine – MPS Solution

Computation Engine Abstraction:

```
<computation name="garch">
  <inputloader id="timeseriesDataloader">
    <inputloader.RuntimeParams minNumberOfInputRecords="1261" dataProcessor="SKIP_NON_DAILY_RECORD_FROM_BEGIN">
      <timeseries field='FX_LAST' freq='DAILY' defaultStartDate='2005-03-31'/>
    </inputloader.RuntimeParams>
  </inputloader>
  <function id="concurrentMpsFunction"/>
  <outputHandler id="timeseriesDataPersistor">
    <outputHandler.RuntimeParams datasourceCode="..." freq='DAILY' timeseriesTypeCode='SIGNAL' datapointTypeCode='VENDOR' operation="..."/>
  </outputHandler>
</computation>
```
Computation Engine – MPS Solution

Remote MPS Implementation:

1. Compute Nodes: MPS
   1. 24 Concurrent-Core/purchase
   2. Arbitrarily Configurable
   3. Expandable to “Unlimited” Cores
   4. Horizontally Scalable (white boxes)

2. Client Side Library: Provided
   1. Java
   2. .Net
   3. C/C++
   4. Python 2.7

Execute a MATLAB function

```matlab
function jsonOut = executeFunction(jsonIn)
    % Convert from json string to matlab object.
    inObj = loadjson(jsonIn)
    % Execute the function.
    fcn = getFunction(inObj.functionName);
    outObj = fcn.execute(inObj);
    % Convert output to json
    jsonOut = savejson('', outObj);
end
```

Base Function

```matlab
% This is the root class of all our asset classes
classdef BaseFunction < handle
    methods (Abstract, Access=protected)
    getInput(self, dataIn);
    validateInput(self, paramsIn);
    doExecute(self, paramsIn);
    handleOutput(self, result);
end
```

```matlab
methods (Access=public)
    function result = execute(self, dataIn)
        % Extract parameters from input data
        paramsIn = self.getInput(dataIn);
        self.validateInput(paramsIn);
        result = self.doExecute(paramsIn);
        result = self.handleOutput(result);
end
```
### Example Application:

**Background:**
1. Regression on security “return/price” time-series by n factor dimensions.
2. A Regression of 7 Factors (permutated) – 5000 cases, costs 60 sec.

**Challenge:**
1. Regress 1,000 funds (5mm case) on desktop, will cost about 1 day.

**Solution:**
1. Data: Parallel/concurrent Connections to DBs, or using cache.
2. Compute: Leverage 10 Servers, each of which contains 10 MPS Runtime-Core. Total of 100 Cores.
3. Compute: should return in minutes.
4. Runtime can be reduced by more MPS cores.
Scaling Research – Q & A

Q & A