Solvency II Overview and High Performance Computing Using Excel for Solvency II and Principles-Based Models

When one computer just isn't enough

William C. Scheel and David Dorfman

ACHS Annual Meeting on Mon. May 23 at Marriott Ricky Hill. 10:00-11:30AM

DFA TECHNOLOGIES, LLC

Speaker: William Scheel, Ph.D.

Bill is a consultant with DFA Technologies, LLC currently working in the area of High Performance Computing (HPC) and using HPC Excel for Solvency II (S II) internal models.

During the last couple of decades, Bill has designed systems for actuarial, accounting, hedge fund and commercial software companies. He served as consultant to the U.S. Department of Labor, the Federal Trade Commission, the U.S. Justice Department and the National Association of Insurance Commissioners. Bill was a Professor of Risk and Insurance for over a decade.

DFA TECHNOLOGIES, LLC

Speaker: David Dorfman

Dave is a subject matter expert on Actuarial Modeling with Microsoft. He came to this role with over 20 years of prior experience working in the area of High Performance Computing (HPC).

During the last few years, Dave has worked with key modeling framework providers to meet the computational challenges of PBA for valuation, Catastrophic risk modeling, and VA Hedging programs. Prior to Microsoft, Dave was a senior staff engineer in the design of the first Cloud Computing Data Center and consulted on several other interesting projects such as FX transfer systems of record and downrange missile telemetry capture.

References

Microsoft, *Step By Step* Windows HPC Server 2008 R2, 2010 Microsoft Corporation, 302 pp.

http://resourcekit.windowshpc.net/doc31.htm

This is an all inclusive reference. Chapter 7 is a must-read for HPC Excel

Why HPC Excel?

- **Performance gains**
- **Precision gains**
- Re-tool existing workbook models while retaining extensible modeling capabilities
- New brute force simulation possibilities
- Mathematica Graphics Processing Units (GPUs) wrappers (*new*)
- Highly scalable scoring methodologies using scoring rules and algorithms in workbooks for data cleansing, classification, credit risk, and myriad other HPC F9-ers

Gaining Modeling Precision Using HPC Excel 2010

5th Year Surplus .01 Percentile

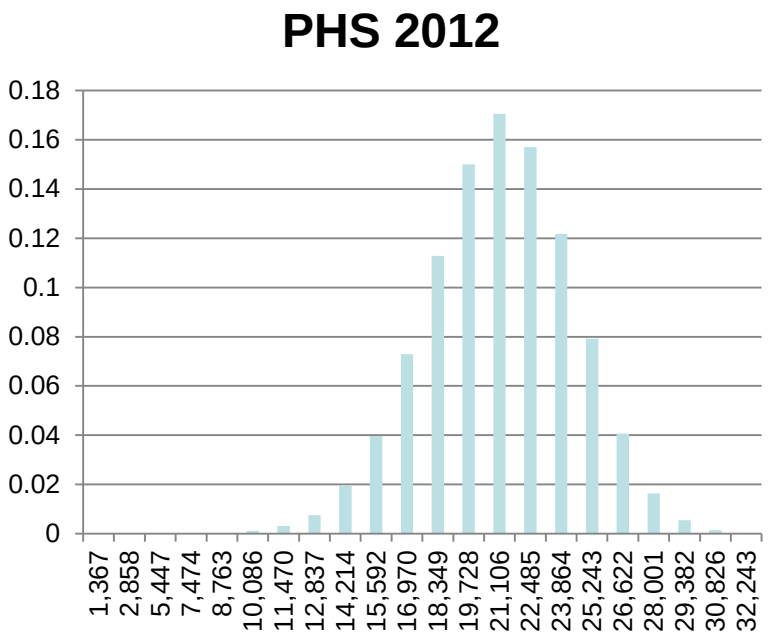
Simulations (Partitions on Compute Nodes)		
5K	25K	100K
14,911	-13,755	9,317
13,727	4,101	8,896
15,905	15,421	12,989
14,189	14,248	7,827
15,065	15,878	3,002
Only when more extreme observations occur during a 100K simulation do we see that we must lower surplus expectations given the business plan.		

DFA TECHNOLOGIES, LLC

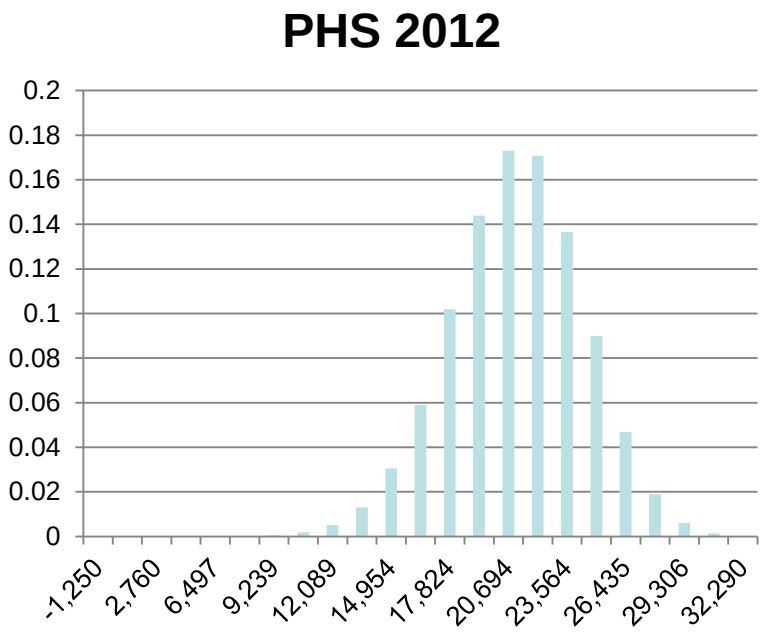
Gaining Modeling Performance Using HPC Excel 2010

Operation (Simulation Partitions)	Microsoft HPC Cluster (approx 225 cores)	Local HPC Cluster (approx 20 Cores)
Standalone 50K Cluster 50K	37.83 mins 3.12 mins	37.45 mins 4.87 mins
50K 75K	3.12 mins 3.77 mins	4.15 mins 9.28 mins
250K 500K 1 mil	8.77 min 15.28 mins 29.6 mins	27.0 mins 1.08 hrs ???

Solvency Metrics and Probability Distributions



50K Partitions



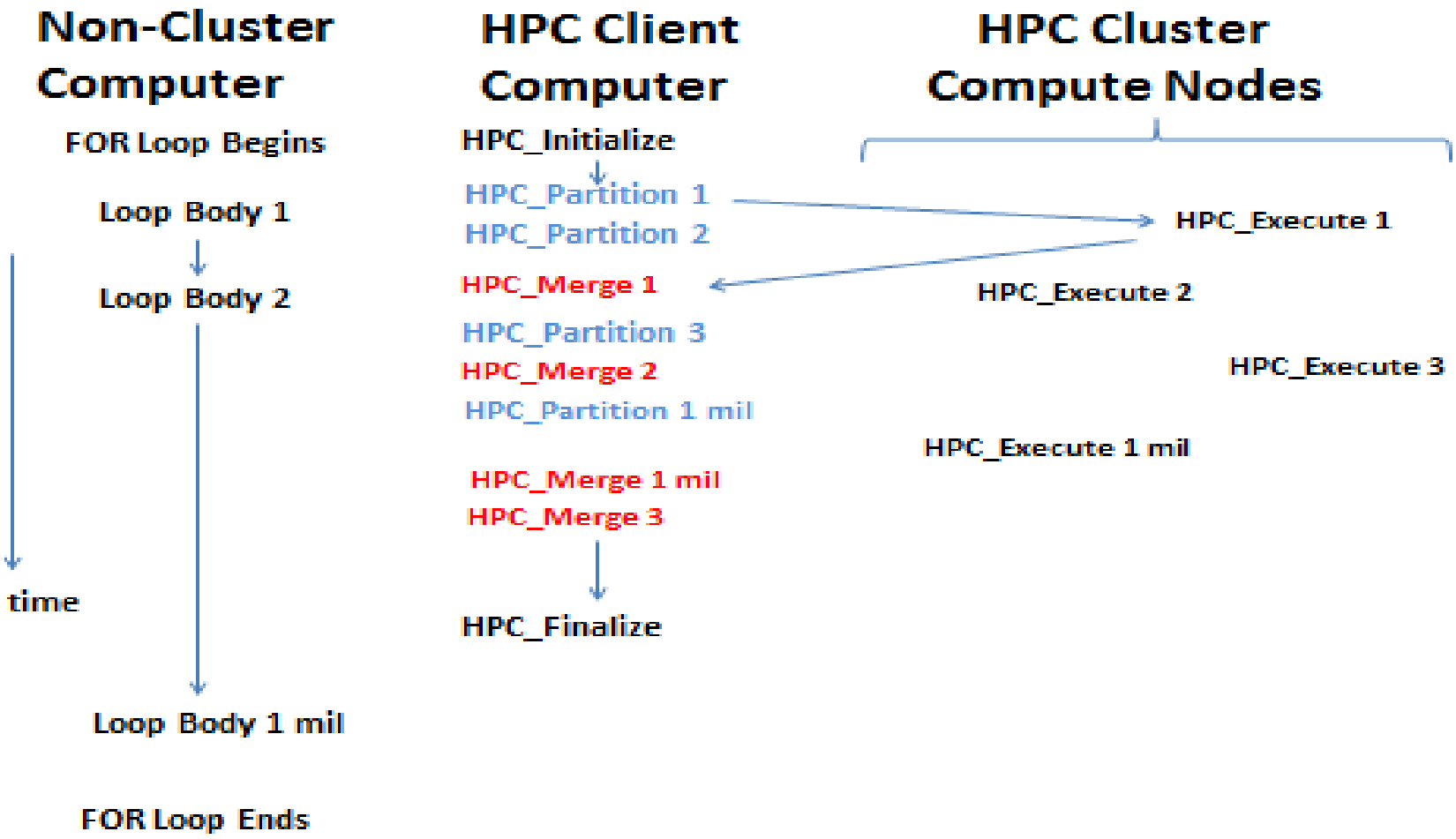
100K Partitions

DFA TECHNOLOGIES, LLC

These Probability Distributions May Look the Same, But They Aren't!

		PHS 2012	
Statistic	10K Simulations	50K Simulations	100K Simulations
Mean	21,101.206	21,096.978	21,098.193
Growth in Mean	.064	.064	.064
Standard Deviation	3,233.008	3,246.951	3,248.577
Coef of Variation	.153	.154	.154
Minimum	5,859.801	-1,011.965	-2,242.016
Maximum	32,927.563	33,461.019	33,638.336
.010 percentile	13,179.515	13,219.961	13,161.036
.100 percentile	16,879.891	16,914.593	16,909.70
.250 percentile	18,980.185	18,956.536	18,956.360
.500 percentile	21,217.692	21,168.816	21,182.596
.750 percentile	23,296.016	23,328.155	23,313.425
.900 percentile	25,214.779	25,216.176	25,215.291
.990 percentile	28,183.047	28,304.902	28,292.670
EPD	7,892.984	8,056.314	8,015.338
TVaR	3,290.422	3,293.325	3,292.605
VaR	12,351.229	12,198.633	12,248.119

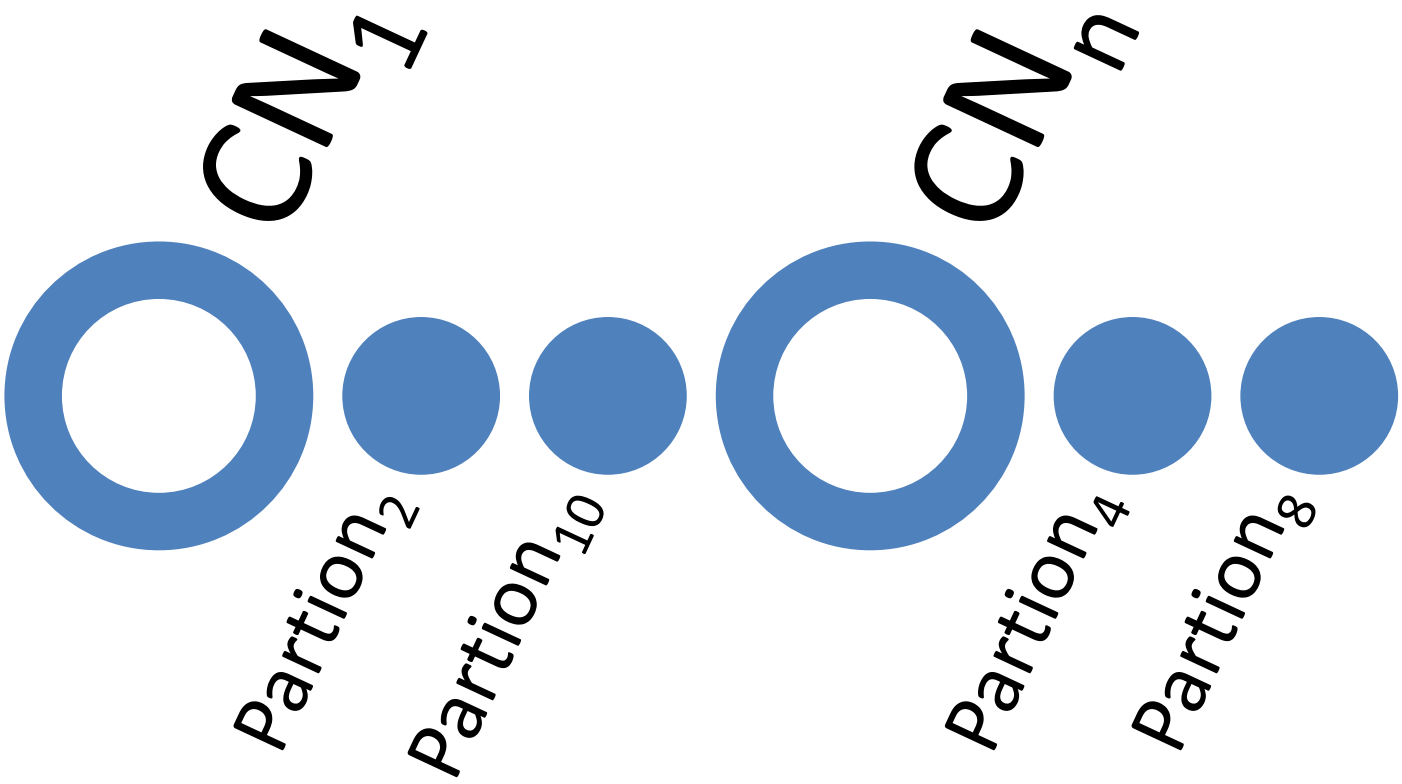
Deconstructed SERIAL LOOP Embarrassingly Parallel



High Performance Grid Computational Setup for HPC Actuarial Cluster

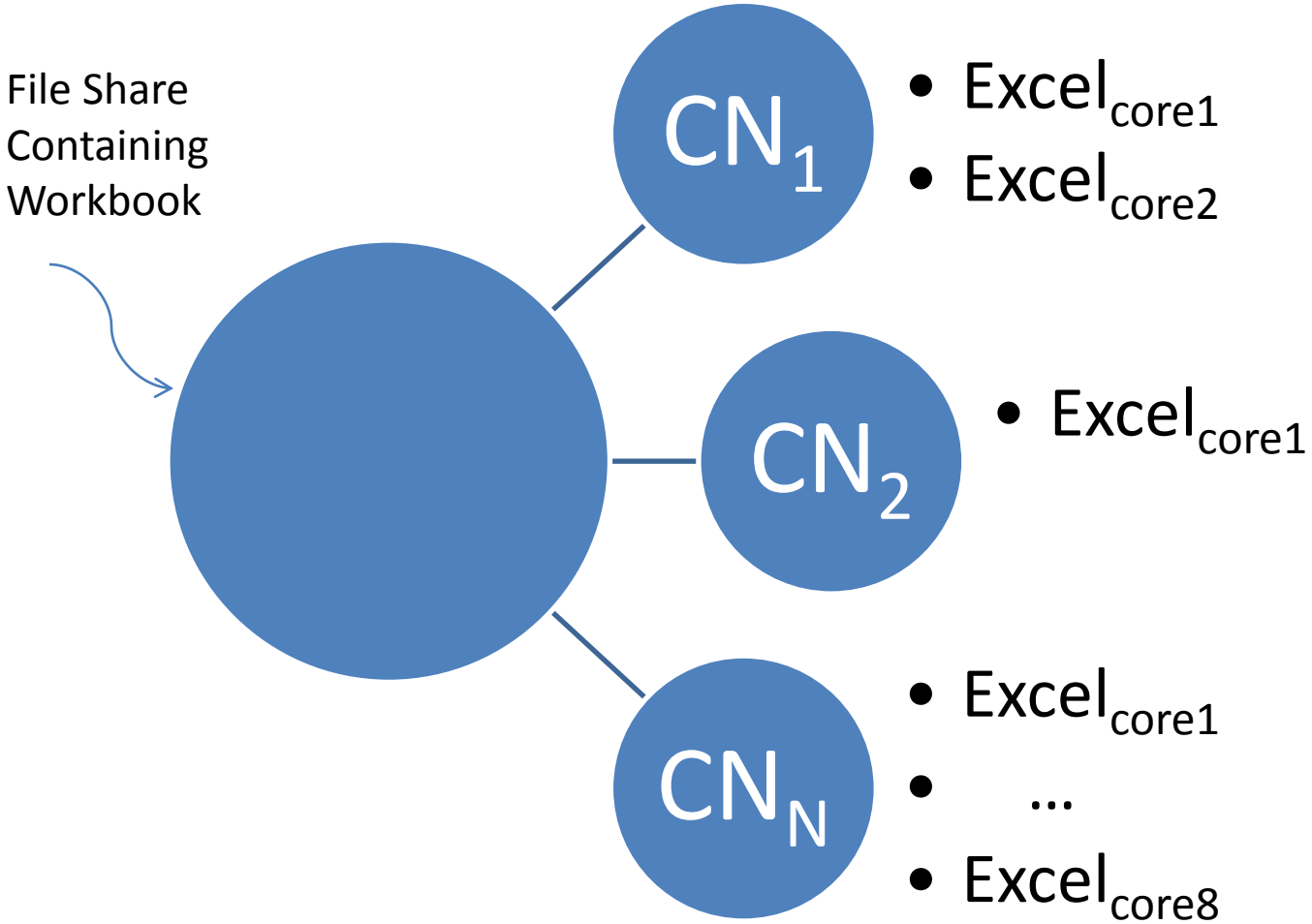
- HPC Cluster with 20-200 cores (less actual computer nodes)
- Software requirements
  - Microsoft Server 2008 HPC
    - Head and broker nodes
    - Compute nodes
  - HPC Pack (all computers)
  - Windows 7—compute nodes
  - 1 copy of Excel 2010 on each computer

Excel Instances and HPC_Execute Code Is Reused with Different Data Partitions

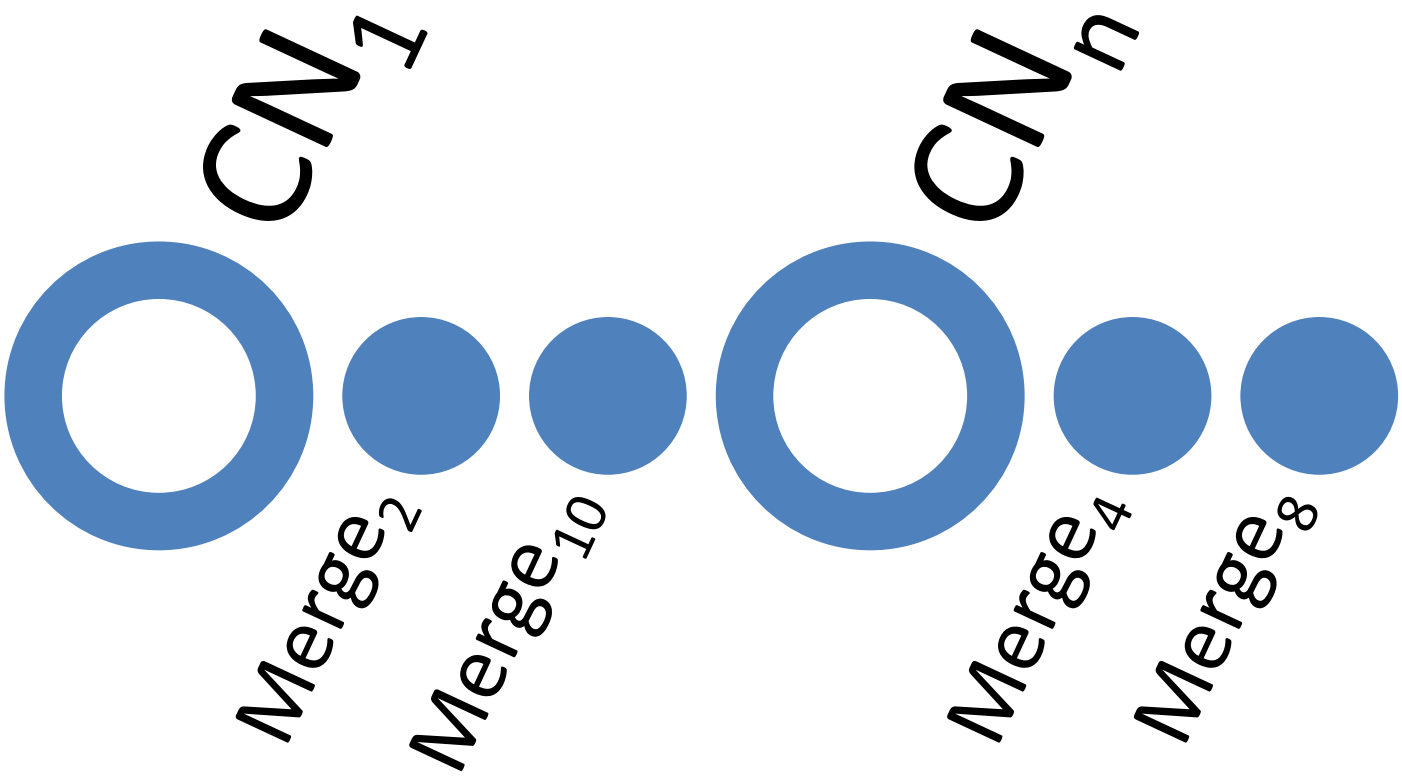


DFA TECHNOLOGIES, LLC

File Share is used by all compute nodes



HPC_Execute Code Derives and Returns Calculated Results to Client



What Is Parallelized in a Life Model?

Alternative 1

- Financial scenarios are the partitioned packets sent by HPC from Client to CNs.
- Every CN must have knowledge about all of the policies.

Alternative 2

- Single policy is a data packet sent by HPC from Client to a CN
- Every CN must have knowledge about all of the financial scenarios.

Programming and data handling considerations determine the best method of parallelization. Neither is inherently the best.

Internal Model (Seriatim Alternative 1)

- Policies are identified by unique parameters (attained age, in-force duration, classification).
- There is a valuation methodology uniquely associated with a policy.
- The parameters of the valuation and valuation function **must be rapidly associated using an addressing procedure** (e.g., given attained age and in-force duration, mortality and lapsation rates can be quickly accessed and used).
- The policy block is valued along a trajectory for a financial scenario. Each CN works with a different financial scenario.
- The desired valuation is a market consistent rendering off Delta-NAV for each policy in the block given a financial scenario.

How Is the Seriatim Used?

Alternative A

1. Each n-tuple is a row with parameter columns.
2. These are arguments ***to a macro*** returning a score such as a reserve or provision

Alternative B.

2. There are no macros; everything for an n-tuple is handled by cellular logic (and, maybe, helper macros.

Scoring using Macros

- This type of scoring is a prime target for HPC evaluation.
- Each n-tuple must be independent
- HPC_Partitions contain row numbers
- HPC_Execute is Range.Execute where the range is the cell containing the macro for scoring! (Do forget to turn off Auto calculation.)

How Much Excel VBA Do I Need To Know?

Possibly more than you do now ☺

Probably less than you might expect ☺☺

1. The macro recorder will not help with HPC-specific code.
2. You need to be comfortable with the Range object and moving data in and out of variants.
3. You need to work with (COM) objects.
4. You will need to learn asynchronous programming.

Data Packets Are Small (64K)

- Public Function HPC_Partition() As Variant
- Public Function HPC_Execute(data As Variant) As Variant
- Public Function HPC_Merge(data As Variant)

1. Variants can be anything EXCEPT objects (e.g., Collections).
2. Variants can be reshaped as they move and need not contain the same structure or data points
3. **Variants can also contain state information that moves from one entry point to another!**
4. **Variant arrays can be quickly moved into worksheet ranges!**

Demonstration on Microsoft HPC Cluster

- HPC Pack References and job launch
- Illustration of HPC callbacks to VBA using code breakpoints on live code
- Introduction to Cluster Manager

This application simulates mortality and lapsation for 50K equity-linked life policies and does a market-consistent valuation of provisions for 500K Vasicek interest rate paths to create a probability distribution.

F9-ers—the Easiest HPC Excel Applications

1. Workbook dependencies calculated from database elements during workbook calculation.
2. Step 1 repeated many times with different data elements.
3. HPC_Execute is really just Workbook.Calculation.
4. HPC_Partition is just moving database elements to CNs.
5. This is referred to as *scoring*, and it is manifest in Excel apps everywhere.

Steps

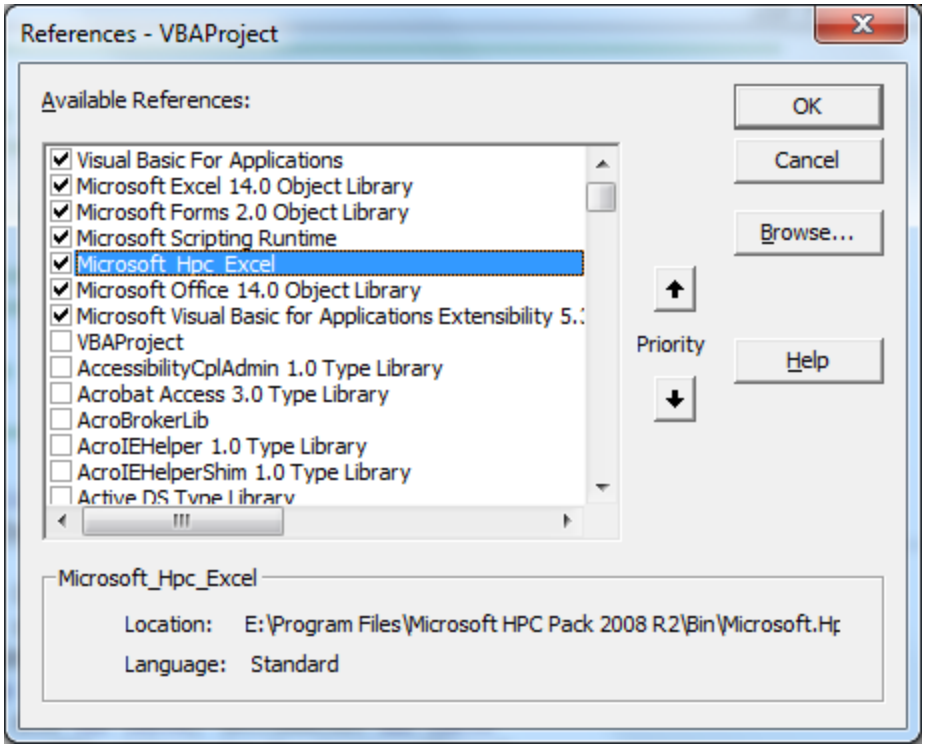
1. **The workbook file is moved programmatically (or manually using file explorer) to the HPC file share.**
2. **HPC starts instances of Excel and identifies the workbook to be opened by each instance. There may be a single instance of Excel or multiple instances started on each HPC compute node (CN) depending on how resources for the HPC job are specified.**
3. **HPC begins an asynchronous callback to the client to get packets of data for use by each Excel CN instance.**
4. **HPC delivers a partition data packet to CN and calls VBA procedure HPC_Execute to use it. All Excel functionality is available during HPC_Execute.**
5. **HPC_Execute fashions a new data packet containing results, and HPC sends it to the client.**
6. **HPC runs a client procedure, HPC_Merge, to use or merge the calculated results into the client workbook.**
7. **Steps 3-6 are repeated until all partitions have been processed.**

The HPC Launch Uses an Excel COM Object (VBA Reference)

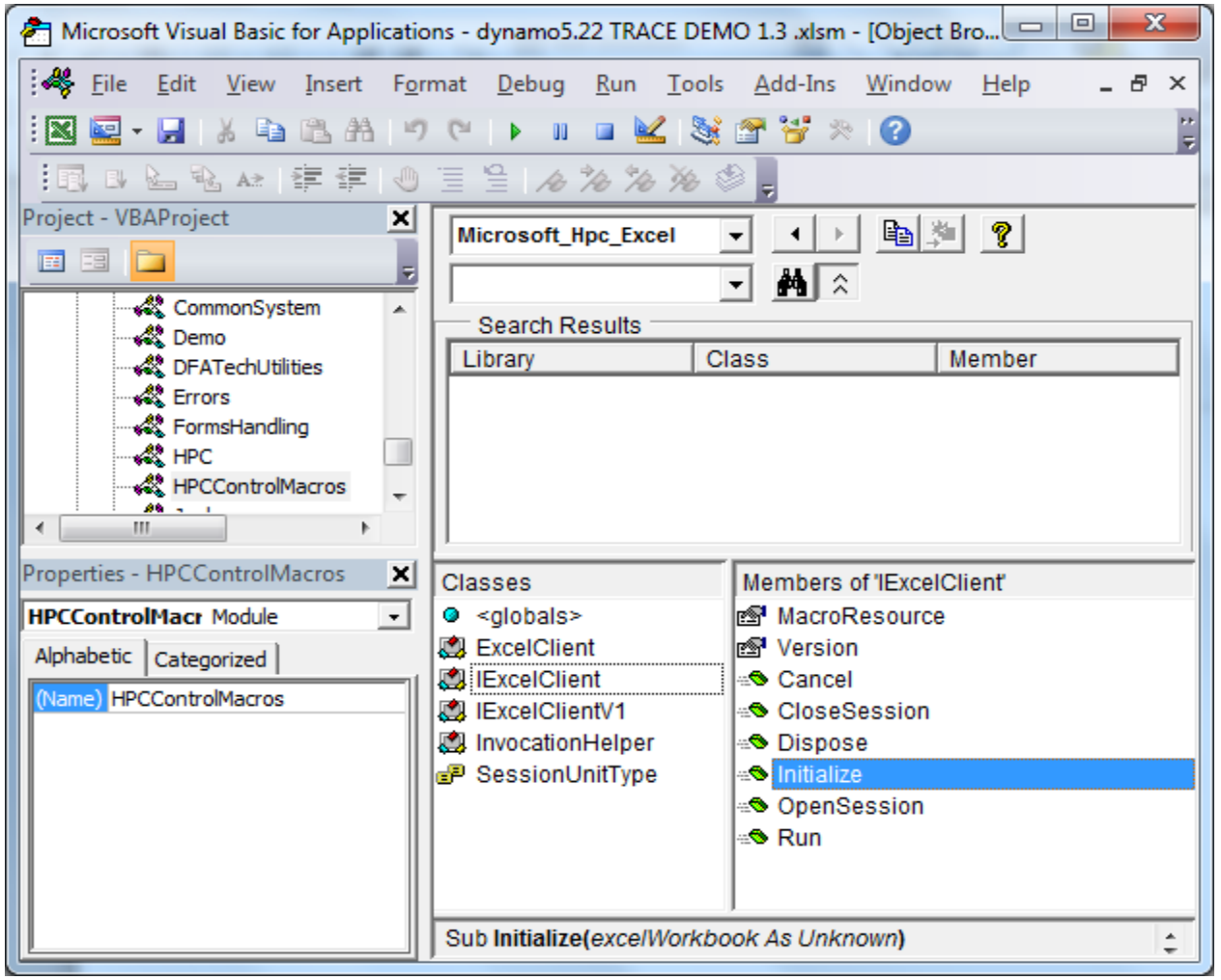
```
Public HPCExcelClient As IExcelClient 'V1

Set HPCExcelClient = New ExcelClient
ThisWorkbook.SaveCopyAs sPath
With HPCExcelClient
  .Initialize ThisWorkbook
  .OpenSession sHeadNode, sPath, _
        ws.Range(MinResources).Value, _
        ws.Range(MaxResources).Value, _
        ws.Range(ResourceType).Value
  .Run CalculateOnDesktop
End With
```

Microsoft_HPC_Excel Object



IExcelClient Methods



HPC_Initialize
(not shown)

HPC_Partition

```
Public Function HPC_Partition() As Variant
        If nPartitionCnt + 1 <= numsims
            nPartitionCnt = nPartitionCnt + 1
            HPC_Partition = CreateMsg()      'this retrieves data from client workbook and
        bundles it into a variant
        Else
            HPC_Partition = Null    'this tells HPC there are no more partitions
        End If
End Function
```

HPC_Execute (compute node Excel instance)

```
Public Function HPC_Execute(data As Variant) As Variant
  'recover in-bound data
  Set ws = ThisWorkbook.Sheets(wsSimulationData)
  Set r = ws.Range(TrialNo)
  nTrial = data(1)

  ReDim data(5)       'data format outbound
  data(1) = ExecuteError.Undefined
  data(3) = nTrial
  data(5) = ComputerName  'compute node

  PartialSimulation nTrial, nEr  'basically app.calculate
  If nEr = MyError.NoError Then
    Set r = .Range(StartData).Resize( etc.)
    data(4) = r.Value
  End If
End Function
```

HPC_Merge

```
Public Function HPC_Merge(data As Variant)
  nMergeCnt = nMergeCnt + 1
  ReDim vD(3) As Variant
  vD(1) = nMergeCnt
  vD(2) = Timer
  vD(3) = data
  vMerge(nMergeCnt) = vD  'vMerge global redim'd HPC_Initialize
End Function
```

HPC_Finalize

```
Public Function HPC_Finalize()
  'looping over contents of vMerge, 1,2,…,i,….
  vD = vMerge(i)
  nMergeCnt = vD(1)
   dtT = vD(2)
   data = vD(3)
     …
   nTrial = data(3)
   vD = data(4)  'results for a trial
     …
   r(1, 0).Value = nTrial
   r(1, 1).Resize(1, UBound(vD, 2)).Value = vD
End Sub
```

Slight-of-Hand Helps Expand What Is Available on Cluster CNs

- Compute nodes have a single workbook opened by HPC, but there's nothing to stop you from having another workbook opened in the same Excel CN instance!
- Set wbData=Application.Workbooks.Open(sSharePath & "HelperWorkbook.xlsm")
- This can be done
  - During workbook Open event
  - During first entry into HPC_Execute

And Using It During CN HPC_Execute

Read range of policy data:

```
Dim vD as variant
vD=wbData.Sheets("Policies").Range("VarLife")(nPolicy,1).Resize(1,nParameters)
```

Use vD in HPC_Execute scoring:

```
With wb
        .Sheets("ScoringRules").Range("PolicyParameters").Resize(1,nParameters).value=vD 'write from helper to HPC wb
        .Calculate      'fire rules dependent on range
        'get scored results
End With
```

Lookups During CN Execution

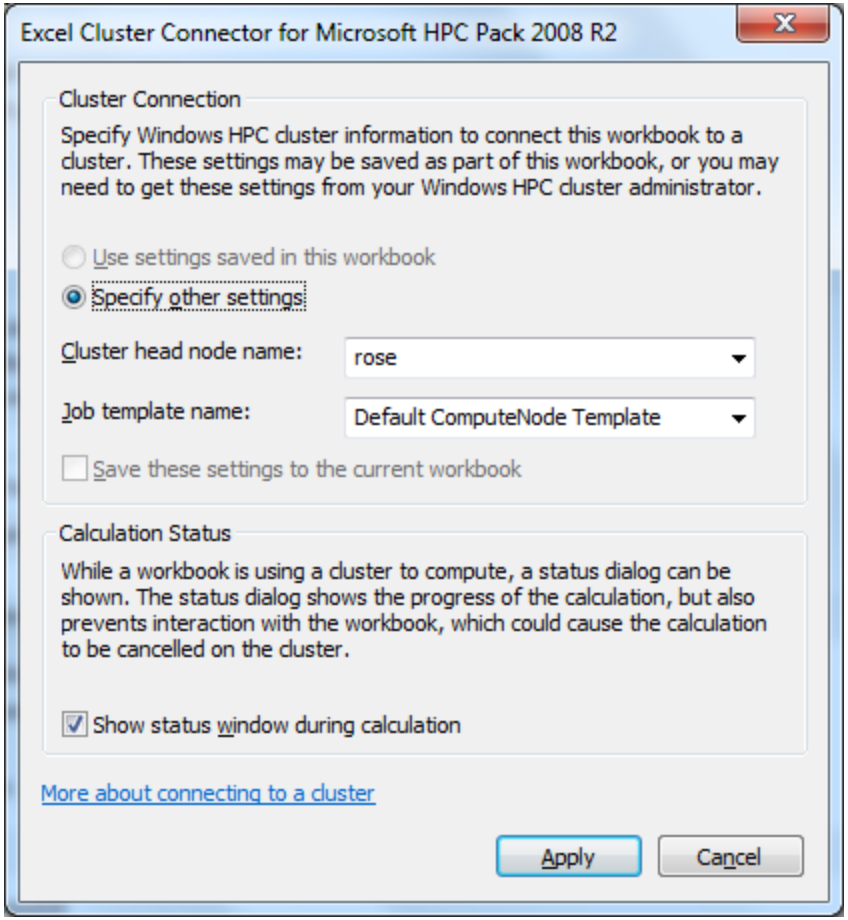
This is a formable problem. Given an attained age, other classification data and in-force duration, a "lookup" is required to obtain mortality, lapsation and other probabilities for a cash flow simulation for every policy.

Consider creating a *collection* of indirect addresses for all policies on *first* entry into HPC_Execute. It is (rapidly) reused on *subsequent* entries for all policies.

Scoring Done by Add-in Functions

- These are candidates for HPC usage.
- Must be cluster safe.

File|Options|Advanced ”Allow UDF XLL…” gets you to this dialog:



What's an XLL?

- C/C++ code with exported functions that become Excel cellular functions through an add-in.
- The XLL must be recompiled to be cluster safe.
- HPC cluster must be provisioned to use the XLL functions.
- If use an XLL, you probably should consider using Mathematica's code generator.

What's Next?

- HPC Excel applications running in Microsoft Azure—a cloud service.
- You may want to experiment first using an actuarial HPC cluster.
- An inventory of existing Excel workbooks with performance issues is a good place to start.
- Think F9-ers.
- If the following Excel VBA code is foreign to you, you need to polish up your VBA skills for HPC:

```
ws.Range(sR).Resize(n,m).Value=vA
vA=ws.Range(sR).Resize(n,m).Value
```

Questions?

93 Silkey Road
North Granby, CT 06060

E: scheel@cox.net
P: 860.653.7169
M: 860.558.0968

Bill Scheel
scheel@cox.net
scheel@dfatech.com

Important HPC (Excel) Reference:
Microsoft, *Step By Step* Windows HPC Server 2008 R2, 2010
Microsoft Corporation, 302 pp.
http://resourcekit.windowshpc.net/doc31.htm