Meradia

Performance, Risk & Analytics

Risk Statistics in Performance Calculators: Suitable and Scalable? – Part 2

SUMMARY

Investment performance calculators have witnessed steady growth in functionality during the past couple of decades. Leveraging huge data management platforms that contain exception management and workflow capabilities, calculators provide upstream integration capabilities. By expanding calculation breadth to include attribution effects and risk statistics, they are knocking on front office doors. As vendor consolidation occurs and performance systems endeavor to deliver one-stop solutions, it is important to understand design mechanisms related to each functionality that drives system efficiencies. In this paper, we examine returns and ex-post risk statistics.

Refer to **Part 1** of this two-part series which lists the subset of risk statistics with which we are concerned and the benefits of calculating these in a performance calculator.

This Part 2 delves into systemic factors that enable efficient computation of risk statistics and why conventional architectures encounter scaling issues.

CONVENIENT AND COMPELLING; BUT IS IT EFFICIENT?

In **Part 1** we provided valid arguments to justify the benefits of calculating risk statistics in a performance calculator. The reasons varied from how performance operations function, to ease of access to data and mathematic calculation functionality.

While there are certainly advantages, we intend to examine efficiency. As the number of risk statistics increase and frequency of calculation becomes higher, computational efficiency plays a critical role in addressing scalability concerns. To meet the growing needs of the industry, the question is not ‘what’ or ‘where.’ It is ‘how.’

HOW IS THE DATA MANAGED?

Performance systems have proliferated during the past couple of decades. Relational database architectures played a prominent role in the structural foundation of most performance systems. An important design aspect in relational databases is the *account* date construct. It is defined as the minimum combination of key elements[1] (account and date) that serve to uniquely identify a set of calculated values. Stated differently, returns and risk characteristics are stored in a way that they intricately tie to *both* the account and date values.

TIME SERIES CONSTRUCT

Another data management construct widely used in statistical packages is the *time* series construct. Here, the return element implicitly defines the underlying account category (portfolio or benchmark or risk free rate), hence it is necessary only to couple it to a date (or date and time) construct. There are at least two benefits of using this construct.

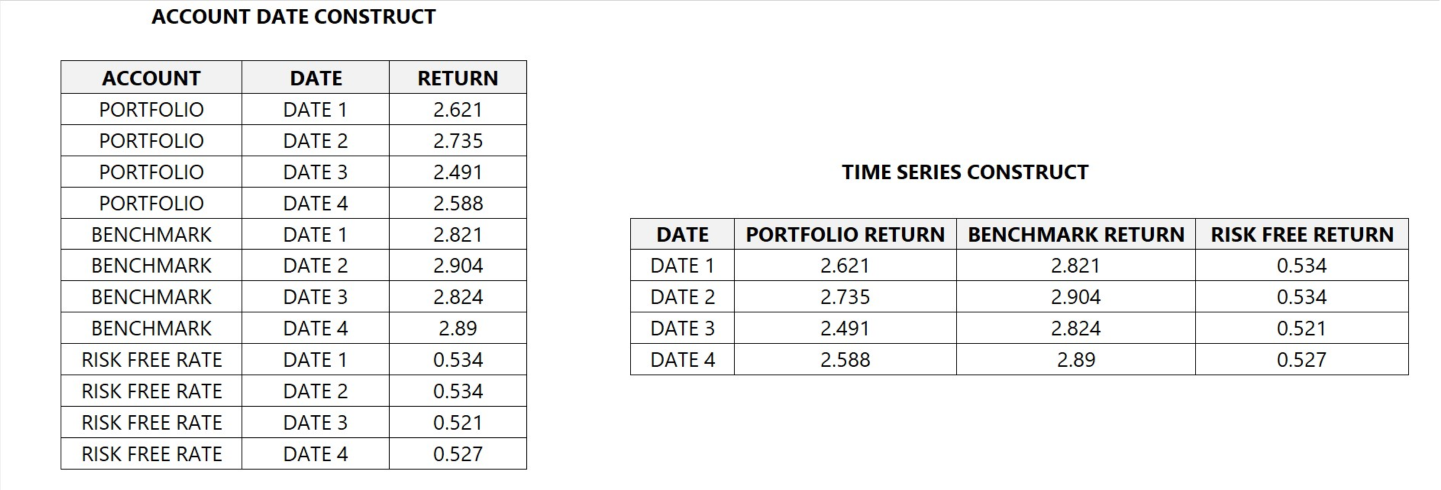
1. Data stored in a time series format helps to parse much more quickly. What would have been as many as three different fetches to a database (e.g. one for account, one for benchmark and another for risk free return series in an account date construct) is replaced by a single fetch for a given set of dates.

2. Software programming languages provide functions that perform arithmetic operations on time series attributes.[2] Hence calculating the difference between portfolio and benchmark returns for a period of dates is akin to subtracting two numbers.

[1] Primary/candidate key in technical terms.

[2] For more information refer to ‘A little book of R for time series’ Release 0.2 by Avril Coghlan. Time series databases (such as Influxdb) also provide certain statistical functions.

TIME SERIES CONSTRUCT (continued)



ACCOUNT DATE CONSTRUCT

ACCOUNT	DATE	RETURN
PORTFOLIO	DATE 1	2.621
PORTFOLIO	DATE 2	2.735
PORTFOLIO	DATE 3	2.491
PORTFOLIO	DATE 4	2.588
BENCHMARK	DATE 1	2.821
BENCHMARK	DATE 2	2.904
BENCHMARK	DATE 3	2.824
BENCHMARK	DATE 4	2.89
RISK FREE RATE	DATE 1	0.534
RISK FREE RATE	DATE 2	0.534
RISK FREE RATE	DATE 3	0.521
RISK FREE RATE	DATE 4	0.527

TIME SERIES CONSTRUCT

DATE	PORTFOLIO RETURN	BENCHMARK RETURN	RISK FREE RETURN
DATE 1	2.621	2.821	0.534
DATE 2	2.735	2.904	0.534
DATE 3	2.491	2.824	0.521
DATE 4	2.588	2.89	0.527

Time series databases are not new to the asset management industry. They have been employed in trading applications to compute intraday volatility for quite some time. With the need to store gargantuan datasets and calculate risk statistics, time series databases are critically situated to make inroads into back office functions. Among the different types from databases available, time series represent the fastest growing segment during the past two years.[3]

Relational databases are optimized for modification and deletion while time series databases are optimized for aggregating data and performing comparisons. Returns finalized after operational review are utilized for risk calculations; hence the underlying data structure that holds returns for risk calculation inputs rarely changes over time and is therefore a good candidate for time series databases.

HOW IS THE CALCULATION DONE SYSTEMICALLY?

A state-of-the-art calculation engine must deliver on accuracy and scalability. With the advent of cloud computing and big data concepts, processing time has been significantly reduced for complex tasks. Horizontal and vertical scaling techniques are being used by applications to complete tasks in reduced time. Vertical techniques incorporate the use of additional memory and processor capabilities on a single machine to perform more operations in less time. Horizontal scaling techniques involve the use of additional systems (nodes) over a network to execute tasks in parallel. While vertical scaling techniques have their limits and do not necessitate change of code, horizontal techniques permit virtually boundless scaling possibilities, but might require changes to the application.

Regardless of the scaling technique used, the potential to identify tasks that can be done simultaneously holds the key to achieving high efficiency. Parallelism is neither new to technology nor to performance calculators; but the level at which parallelism is in action will determine the scalability of risk calculations. We call this 'parallelism depth.'

PARALLELISM DEPTH

Parallelism depth is defined as the most granular level at which a performance calculator can execute tasks simultaneously. Each granule can be thought of as a block of work that can be processed independently of other granules. For example, an account can calculate returns independently of another account. Certain use cases might also permit calculating performance returns for several dates within an account simultaneously. Conventional performance calculators certainly incorporate this mechanism and accomplish a fair degree of parallelism from a return calculation standpoint. We assign the parallelism depth to be 1 for the account scenario and 2 for the account and date case.

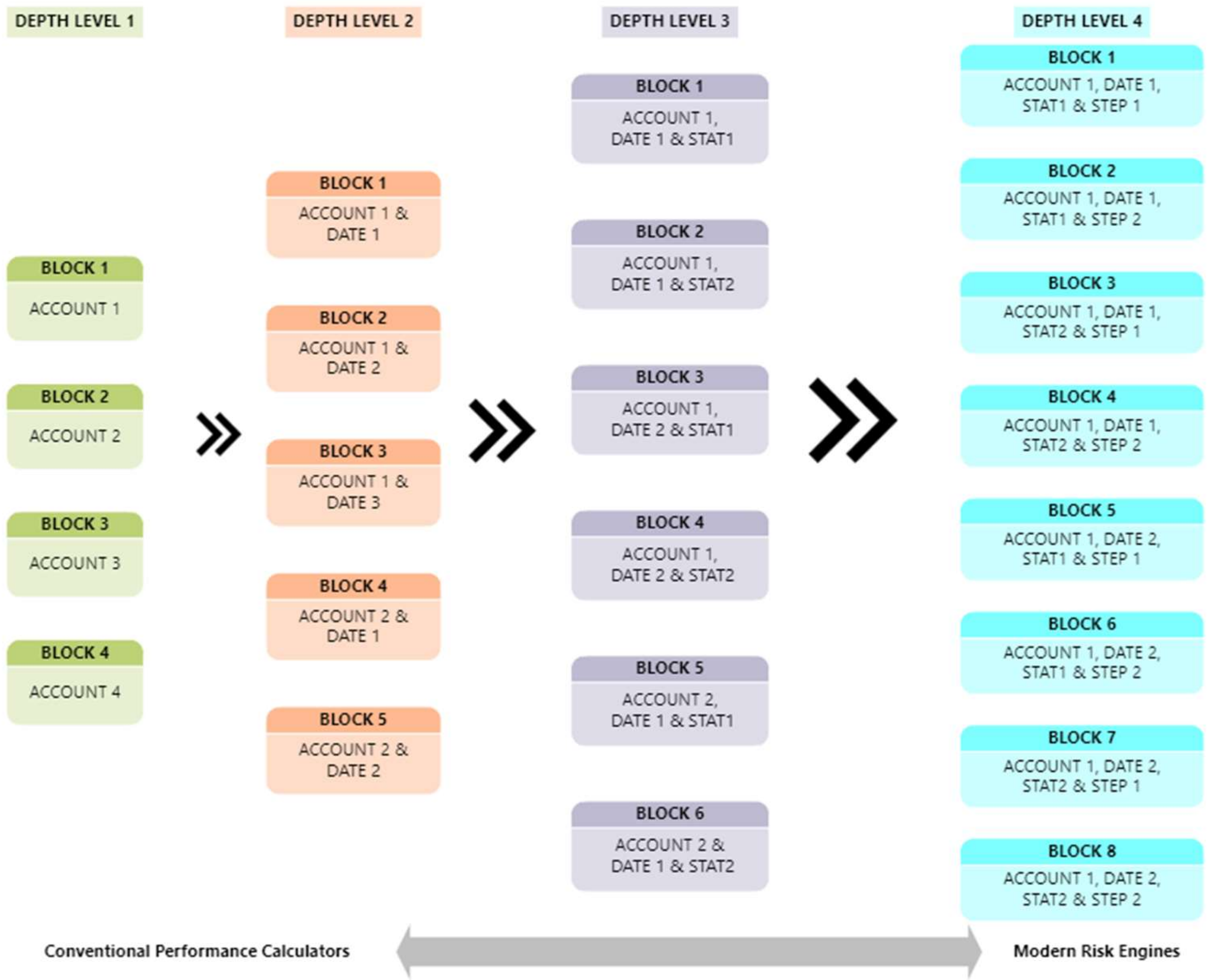
When it comes to risk, each statistic within an account can be computed independently of another statistic, termed as depth level 3. For example, Sharpe ratio and Treynor ratio can be calculated simultaneously for an account. Furthermore, there are several steps within a risk statistic that can be computed in parallel. For example, let's assume that Information Ratio must be calculated based on returns data available over a one-year period. The calculation can be split into multiple steps:

[3] Based on a survey of 1100 participants conducted by Percona, Feb 2017.

PARALLELISM DEPTH (continued)

Step 1: Calculate one year average return of the portfolio
Step 2: Calculate one year average return of the benchmark
Step 3: Compute excess return daily over the one year time period
Step 4: Compute standard deviation of the excess return
Step 5: Compute Information Ratio based on data computed in Steps 1, 2 and 4

Steps 1, 2 and 3 can be executed in parallel. In such a highly optimized environment, the depth level is termed as 4.


DEPTH LEVEL 1
BLOCK 1
ACCOUNT 1
BLOCK 2
ACCOUNT 2
BLOCK 3
ACCOUNT 3
BLOCK 4
ACCOUNT 4
DEPTH LEVEL 2
BLOCK 1
ACCOUNT 1 & DATE 1
BLOCK 2
ACCOUNT 1 & DATE 2
BLOCK 3
ACCOUNT 1 & DATE 3
BLOCK 4
ACCOUNT 2 & DATE 1
BLOCK 5
ACCOUNT 2 & DATE 2
DEPTH LEVEL 3
BLOCK 1
ACCOUNT 1, DATE 1 & STAT1
BLOCK 2
ACCOUNT 1, DATE 1 & STAT2
BLOCK 3
ACCOUNT 1, DATE 2 & STAT1
BLOCK 4
ACCOUNT 1, DATE 2 & STAT2
BLOCK 5
ACCOUNT 2, DATE 1 & STAT1
BLOCK 6
ACCOUNT 2 & DATE 1 & STAT2
DEPTH LEVEL 4
BLOCK 1
ACCOUNT 1, DATE 1, STAT1 & STEP 1
BLOCK 2
ACCOUNT 1, DATE 1, STAT1 & STEP 2
BLOCK 3
ACCOUNT 1, DATE 1, STAT2 & STEP 1
BLOCK 4
ACCOUNT 1, DATE 1, STAT2 & STEP 2
BLOCK 5
ACCOUNT 1, DATE 2, STAT1 & STEP 1
BLOCK 6
ACCOUNT 1, DATE 2, STAT1 & STEP 2
BLOCK 7
ACCOUNT 1, DATE 2, STAT2 & STEP 1
BLOCK 8
ACCOUNT 1, DATE 2, STAT2 & STEP 2
Conventional Performance Calculators
Modern Risk Engines

Concurrent processing of accounts or accounts and dates is optimized to address high volume processing for performance calculator while maintaining order of operations. Parallel processing below account level creates a potential source of risk of handling transactions out of order. Optimized processing for risk calculations occurs at the account + date + stat level + step level. Risk statistic calculations need not worry about the order of certain steps. The greater the depth of parallelism, the quicker the computation. Higher efficiency results in greater scalability.

Risk Statistics in Performance Calculators: Suitable and Scalable? – Part 2

CONCLUSION

While there are undeniable benefits to performing risk calculations in a performance engine, this two-part series highlights two factors, time series construct and depth of parallelism that determine the scalability (efficiency) to a considerable extent. It is also an attempt to start a broader dialogue among the performance systems community around data management concepts and processing paradigms that play a central role in designing performance systems.

HOW MERADIA CAN HELP

Meradia, with its combination of domain experts and technology experience, is uniquely positioned to address challenges in the performance systems arena. Whether it's performance systems re-engineering, integration or new product development, our collective expertise with several platforms will provide your organization with valuable perspectives.

Jose R. Michaelraj, CIPM,
is a Senior Consultant who brings 14 years of progressive, varied investment services experience. His strong domain, functional and technical skills; combined with end-to-end process knowledge, data management expertise and project leadership result in effective and efficient solutions for our investment clients. Fluent in SQL and an expert in translating business needs, Jose is competent in helping clients design, test and implement practical, long-term solutions. For Meradia he has been analyzing, designing and implementing data management and performance measurement systems on the Eagle platform. For the global risk solutions group of an asset servicing client, Jose redesigned important aspects of investment performance systems, established best practices for the information delivery framework and defined standardized performance validation procedures that applied to US, Europe and India.

Meradia
119 North High Street
West Chester, PA 19380
Phone: 610-738-7787

Info@Meradia.com
www.Meradia.com