About Showers and Streams: Benchmarking Big Event Data

Hans-Arno Jacobsen, Middleware Systems Research Group, MSRG.org

Summary: In this abstract, we argue for the need to (also) develop benchmarks and standards for systems that handle big event data. By that we meant applications that handle massive amounts of events, which we refer to as applications of big event data, in analogy to big data applications.

Extended Abstract: There is a growing number of applications that produce massive amounts of event data, also simply referred to as events. One the one hand, these events need to be filtered and correlated in real-time in order to detect patterns and emergent behaviour, and, on the other hand, these events need to be stored for subsequent analysis and record keeping. Filtering requirements range from needing to evaluate expressive filter expressions, process many filter expressions simultaneously, and handle large amounts of events. Storage requirements range from needing to store raw events, aggregated events, and derived events.

Examples include application performance management and emerging smart grid application scenarios. Practitioners of application performance management (APM) report event rates of one million events per second and more and applications for smart grids can easily reach such volumes when one considers energy consumption and production monitoring data across large cities or so called mega-cities. Other applications are found in finance, telecommunication, and health-care.

In the APM context, the event data or event, may represent the monitored system metrics, such as a specific method invocation in a given application tier running in some virtual machine on a given physical host. In the smart grid context, events may represent energy consumption information for specific devices, households, buildings, streets, and neighbourhoods.

In many of these scenarios, it is a requirement to not only filter events, but to also store events for subsequent processing. While much effort has been devoted to either end of this spectrum, that is to the filtering of events as well as to the storage of data, little attention has been devoted to enable both kinds of operations running in tandem in one and the same system. We argue that this is an important consideration for designing novel event processing capabilities as well as benchmarks and standards in this area.

Colloquially, we refer to solutions for event processing as "event showers" as pendat to event stream processing approaches. The difference is meant to emphasis that there is a class of applications that require event processing capabilities that can handle events of different types, shapes, and evaluate events efficiently against many event pattern specifications rather than relying on the assumption that every event in an event stream follows the same schema and processing generally involves a single stream query at a time. Approaches to the former have emerged in the context of content-based publish/subscribe-style processing, while the latter have emerged in the context of streaming databases. Another term that is used in the literature is "event cloud" to refer to the scenarios that require capabilities we attribute to "event showers". To not confuse "event clouds" with the in popularity gaining concept of cloud computing, we

opted for a new term.

Overall, there is neither a standard language model nor language algebra, for defining event expressions, events, streaming queries or streams. This lack of common ground makes it a challenge to compare, let alone to evaluate different approaches and identify their differences. Also, as of this writing, no de facto standard exists. Similarly, we are not aware of any benchmark development efforts in this space. Given the grounding of event processing and event stream processing in applications across essentially any vertical markets, we feel it is essential to consider standardization and benchmark development as a priority.