unravel

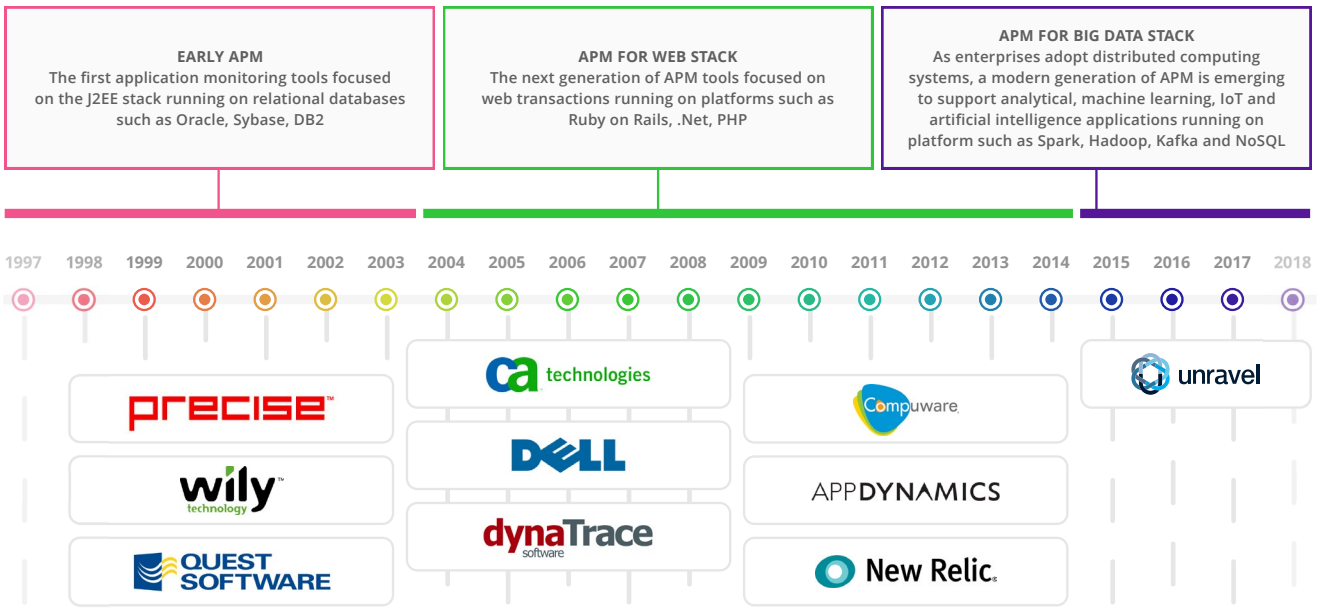
Application Performance Management for Big Data Applications and Platforms

» Operations Guide

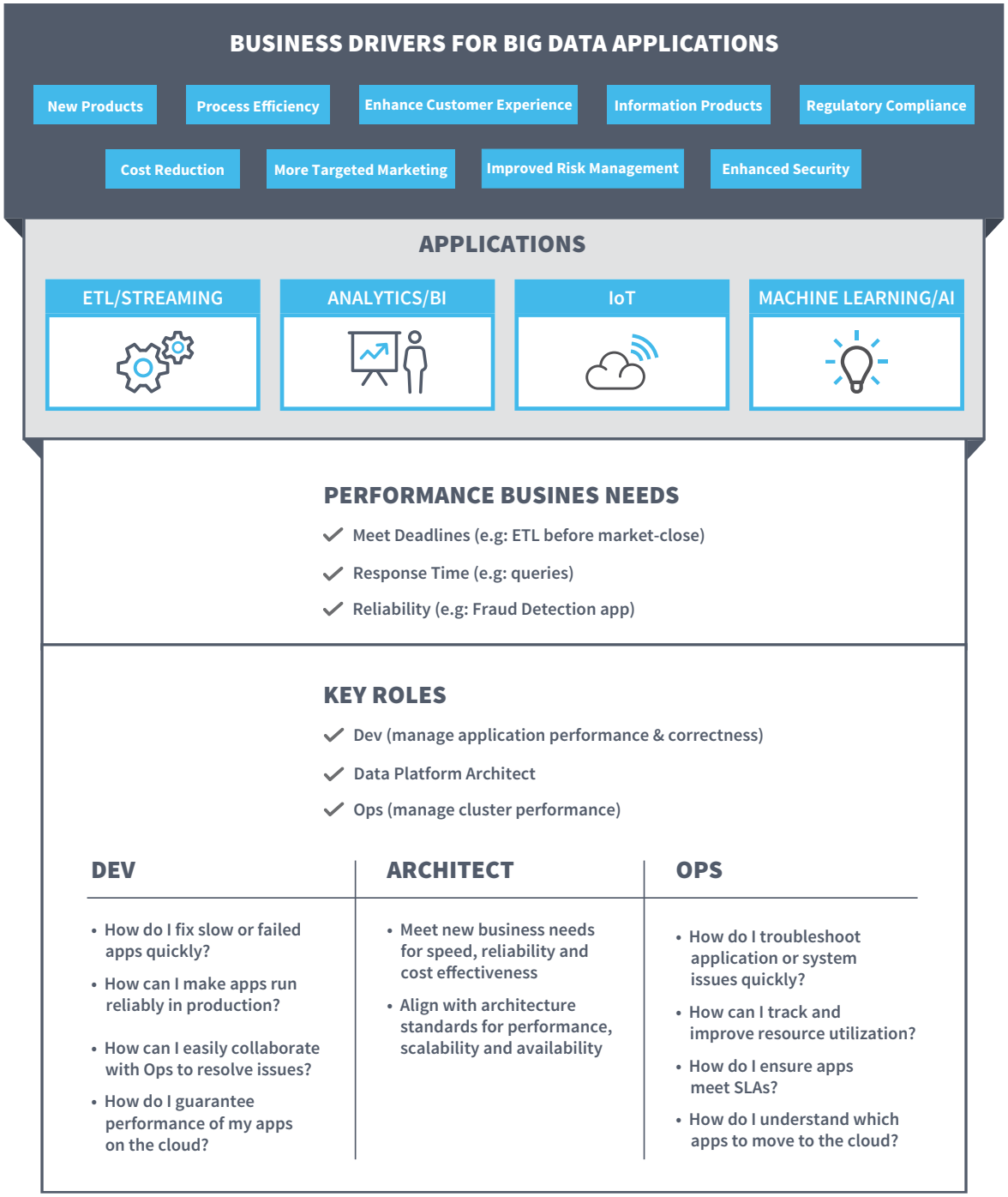
What is Big Data APM and why do enterprises need it?

Big data is trending from experimental projects to becoming a mission-critical data platform offering a range of big data applications. Enterprises look to these big data applications (e.g., ETL offload, Business Intelligence, Analytics, Machine Learning, IoT, etc.) to drive strategic business value. Application Performance Management (APM) is not a new discipline, but it is a new best practice for big data – adopting an application-first approach to guarantee full-stack performance, maximize utilization of cluster resources, while minimizing the TCO of the infrastructure.

EVOLUTION OF APPLICATION PERFORMANCE MANAGEMENT (APM)



As applications get deployed in a multi-tenant cluster, whether it's on-prem or in the cloud, Operations Teams face many critical challenges to support business needs to meet SLAs, support many users, especially as the number of users and systems continue to grow.

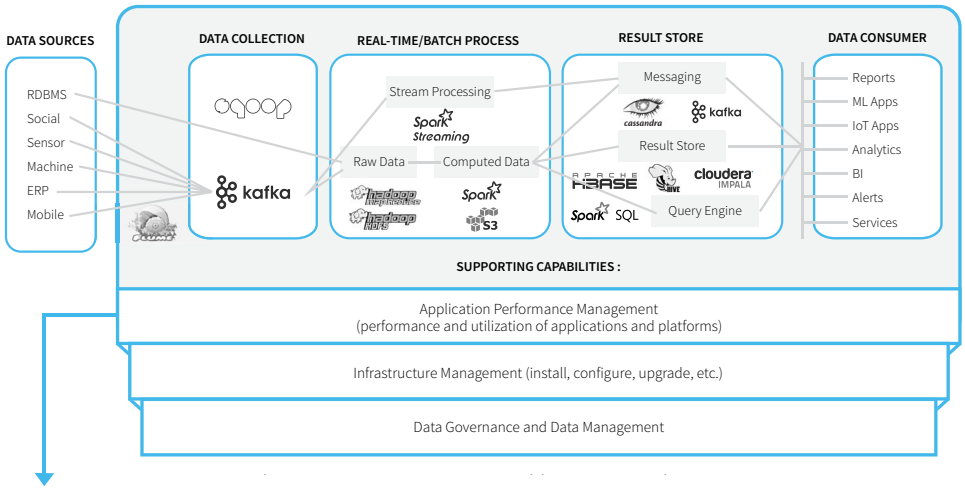


As big data applications move to production, performance expectations also need to be production-grade. The business needs answers in seconds and not hours, hardware and resources need to be continuously optimized for cost, and deadlines / SLAs need to be guaranteed. This means that APM needs to become a strategic component of a big data architecture in order to eliminate risks and costs associated with poor performance, availability, and scalability.

Current challenges

The fundamental problem is that the big data stack is complex due to its' distributed nature where infrastructure, storage and compute are spread across many layers, components, and heterogeneous technologies. This problem exists regardless of the specific architecture (i.e., traditional, streaming analytics, Lambda, Kappa, or Unified). From the perspective of the production big data platform and the applications that run on it, everything must run like clockwork: ETL jobs must happen at fixed intervals; users expect dashboards to be up-to-date in real time; user-facing data products must work constantly. But from the perspective of the underlying platform, the application is not an isolated job, but rather a set of processing steps that are threaded through the big data stack. **For example, a fraud detection application (i.e., Data Consumer) would be comprised of a chain of many systems from Spark SQL, Spark Streaming, HDFS, MapReduce, and Kafka, as well as many processing steps within each system.** As a result, the entire process of managing performance and utilization across all these layers is exponentially complex.

From the perspective of the production big data platform and the applications that run on it, everything must run like clockwork: ETL jobs must happen at fixed intervals; users expect dashboards to be up-to-date in real time ; user-facing data products must work constantly



PERFORMANCE INFO LEVELS	CURRENT APPROACHES AREN'T ENOUGH
Recommendations	None
Insight	None
Correlated Information	Manual process or coding
Logs	Look at component logs
Jobs KPIs	Not always available

Example big data architecture

CURRENT CHALLENGES:

Lack of reliability
Missed SLA and revenue

Sub-optimized resources
High infrastructure and project costs

Inefficient resolution
Long MTTR (mean time to resolution)

This complexity makes it very hard to implement Application Performance Management services that provide a single view to manage performance and utilization across the full-stack. In particular, there is no rationalized instrumentation across the stack to enable a holistic approach to guarantee performance and maximize utilization. Instead, performance and utilization information is scattered across disjointed metrics, buried in logs, or spread across performance monitoring / management tools that only provide an incomplete infrastructure view as opposed to a full stack view.

Challenges for Operations Teams

As a result, the process of planning, operationalizing, and scaling the performance and utilization across applications, systems, and infrastructure is not production-ready. This challenge is called out in Gartner's March 2017 **Market Guide for Hadoop Operations Providers**. The report states "scaling Hadoop from small, pilot projects to large-scale production clusters involves a steep learning curve in terms of operational know-how that many enterprises are unprepared for."

The lack of production readiness spans multiples areas across business units, developers, and operations. At its core, it makes it impossible to implement a multi-tenant cluster model, where a small Ops team needs to support a large number of applications, business units, and blended workloads with a combination of SLA-bound jobs vs. data discovery. The ultimate impact affects adoption of big data and business value realization.

PLAN

- No understanding of dependencies across the big data stack
- Lack of understanding of application usage
- Guesswork to size and tune clusters

OPERATIONALIZE

- Manual troubleshooting
- Trial-and-error resolution
- Escalating support tickets

SCALE

- Lack of control to maximize utilization
- Lack of self-service to troubleshoot issues
- Lack of governance to optimize storage and compute utilization

Common Big Data Operations Challenges

Lack of visibility of Cluster usage from an application perspective

- Example 1: Which application(s) cause my cluster usage (CPU, memory) to spike up?
- Example 2: Are queues being used optimally
- Example 3: Are various data sets actually being used by the different applications?
- Example 4: Need a comprehensive chargeback/showback view for planning/ budgeting purposes.

Not having good visibility and understanding when Data Pipelines miss SLAs? Where do we start to triage these issues?

- Example 1: An Oozie orchestrated data pipeline that needs to complete by 4AM every morning is now consistently getting delayed.

No good understanding of which "knobs" (configuration parameters) to change at the cluster level to improve overall application performance and resource usage.

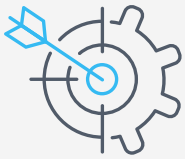
Unable to control/manage runaway jobs that could end up taking way more resources than needed effecting overall cluster and starving other applications

- Example 1: Would need an automated way to track and manage these runaway jobs

How to quickly identify inefficient applications that can be reviewed and acted upon?

- Example 1: Most Ops team members do not have deep Hadoop Application expertise. So having a way to quickly triage and understand root-causes around application performance degradation is very helpful

OPERATIONS TEAM'S FRAMEWORK



PLANNING

OPERATIONALIZING

SCALING

Complex infrastructure landscape
Globally used multi-tenant platform with a complex landscape of technologies and application usage patterns.

100+ projects
5,000+ jobs/day
600+ users globally
3PB+ of data

Debugging performance problems is a challenge
Data Science and applicant development teams do not have access to streamlines UI for viewing job metrics and performance bottlenecks.

>5 different interfaces for job monitoring
>10 different logs for debugging a single workflow

Increased costs
Lack of tooling and absence of a top down approach to debug application performance issues results in teams spending significant time to understand, diagnose and determine root cause for slow running jobs/workflows.

1-2 weeks
to determine root cause for performance issues

Sub-optimal capacity management
Missing capacity and charge back reporting lowers visibility for project owners and absence of a capacity forecasting solution causes conservative forecasts for tin leading to wastage.

>1m spend
an un-utilized storage in 2017

Missing insights on data operations
Lack of insights on data usage reduces the ability to optimize data storage and access patterns and does now allow chargeback models to be data driven.

80% of the datasets
can be candidates for lower cost storage/change in retention rules

Ineffective alerting and automatic actions
Inability to alert and pro-actively act on events leads to performance bottlenecks in a busy multi-tenant platform.

99% of all the current alerting
cannot be co-related with performance issues

Customer Case Study: How Unravel is helping a Global Fortune 200 Financial Services Company

Here is a customer example of a typical situation for a multi-tenant cluster that illustrates the mission critical problems that Big Data Operations Teams face. This example provides the business case justification to implement APM for Big Data.

8

FPO

In order to answer these questions: you need full stack data (apps, datasets, resources, users) and intelligence that can automatically correlate, root-cause and provide fast and accurate answers to these questions.

Best practices

Managing a big data platform in production requires a set of best practices. The following is a list of best practices that pertain to application performance management for production big data applications and clusters.

1. TEAR DOWN SILOS AND BE PRO-ACTIVE

- Adopt a lifecycle and 24/7 production approach. It will enable you to notice changes in data and compute distribution over time. In addition, a lifecycle approach will allow you to immediately pinpoint any negative changes introduced by new applications, users, or changes in the big data platform.
- Don't just wait to fix problems in production. The ultimate goal is to become proactive by fixing issues in development or testing—before they spill over into production.
- Instead of relying on fragmented and incomplete sets of tools, use one single solution that is designed to be used

across the application lifecycle, from development to production, as well as all the systems the application will run on. It will make it easier to share application performance data between lifecycle stages and understand the dependencies across all the underlying systems and cluster resources.

2. DOWN TO "SOURCE CODE" TROUBLESHOOTING

- Knowing if it was the code or infrastructure or something in between that is causing poor performance of applications is key to expedient troubleshooting. An APM solution needs to lower the Mean-Time-To-Resolution (MTTR) from as much as several days down to minutes. In order to do so, full-stack information needs to be available and correlated in order to pin-point the root-cause quickly and confidently.

3. CENTRALIZE AND CORRELATE DATA COLLECTION OF PERFORMANCE INFORMATION

- It is not feasible to manually collect different types of performance information from different big data systems and clusters. It would require multiple monitoring tools, logs and custom data transformation to create a single view that can be analyzed – this approach is both expensive and slow to implement and requires a large number of experts. But the biggest drawback is that the data from different tools is either incomplete or doesn't "line up."
- Instead of this approach, use a single view of performance and utilization across the stack to all key stakeholders from Ops, to Dev, and support teams.
- Correlating, storing, and accessing performance data from a single, centralized repository enables fast and powerful analysis and visualization with a correlated view of performance metrics.

4. ABILITY TO EXTEND AND INTEGRATE

- The big data stack is an ever-changing landscape of new systems and environments.
- An APM solution must be able to adapt to integrate with the big data stack as it evolves (e.g., support for new technologies like Impala, Tensorflow, Kudu, etc.).

5. DON'T JUST THROW MORE AND MORE HARDWARE AT THE PROBLEM

- Although hardware is cheaper, it is not cheap. But you also have to consider the operational drag. Every node you add makes the cluster more complicated. Instead, ensure that you can understand and optimize all jobs and workflows across SLA-bound and "rogue"

applications to reduce both the time and resources it takes to run them.

6. INSTITUTIONALIZE A PERFORMANCE BEST-PRACTICE

- Implement APM as a discipline to drive an application-driven mindset in the organization. Start by implementing a Performance Center of Excellence. This will serve as the foundation to help the organization move from reactive performance troubleshooting to proactive performance prevention.

Critical capabilities for Application Performance Management

There are critical capabilities required to manage performance and utilization of big data applications and the platform they run on. These capabilities span both business and technical requirements to ensure all needs are met between business and IT.

FULL STACK SUPPORT

Big data applications don't execute in isolation-everything is part of some larger workflow. The ability to support full-stack means that performance and utilization need to be managed not just for one job, but correlated across all the jobs the application is dependent on. This capability is essential in order to thread an end-to-view from application down to infrastructure, because jobs depend on other jobs, and code contributions come from several development teams which then need to be orchestrated by a different infrastructure team.

SLA MANAGEMENT

SLA management ensures that business needs in terms of responsiveness and uptime are defined, agreed-upon, and will be met. This includes identifying adherence to contracted service-level application

BEST PRACTICES

1. Tear down silos and be pro-active
2. Down to "source code" troubleshooting
3. Centralize data collection of performance information
4. Ability to extend and integrate
5. Don't just throw more and more hardware at the problem
6. Institutionalize a performance best-practice

availability and performance thresholds. SLA metrics provide visibility to LOB and application owners, as well as IT operations and DevOps teams, in order to have a shared view of key business needs.

ANOMALY DETECTION AND AUTOMATED ACTIONS

The ability to detect and rapidly respond to situations that begin to fall outside normal operating parameters. The earlier this detection takes place, the better, in order to prevent cascading problems. Anomaly detection is critical for IT operations, DevOps and even application support teams. Moreover, the ability to take automated actions further improves the speed with which downstream problems can be prevented, as well as frees up a small Ops or support team to focus on more critical issues.

AUTOMATED TROUBLESHOOTING AND INSIGHTS

There are many potential sources of application performance degradation, and root cause analysis can span an exponential number of parameters. As such, manual troubleshooting cannot scale in a multi-tennant production environment. The ability to automatically identify the source of the degradation, as well as provide insight as to the causes, can dramatically improve the productivity of DevOps and support teams while drastically reducing the Mean-Time-To-Resolution.

AUTOMATED RECOMMENDATIONS

Automated recommendations go one step further than automated troubleshooting and insights. Automated recommendations eliminate the need for trial-and-error in fixing problems during production, by providing a specific fix which DevOps or supprt teams can more readily deploy. Automated recommendations can also accelerate debugging during the application development process.

WORKLOAD PLANNING AND CHARGEBACKS

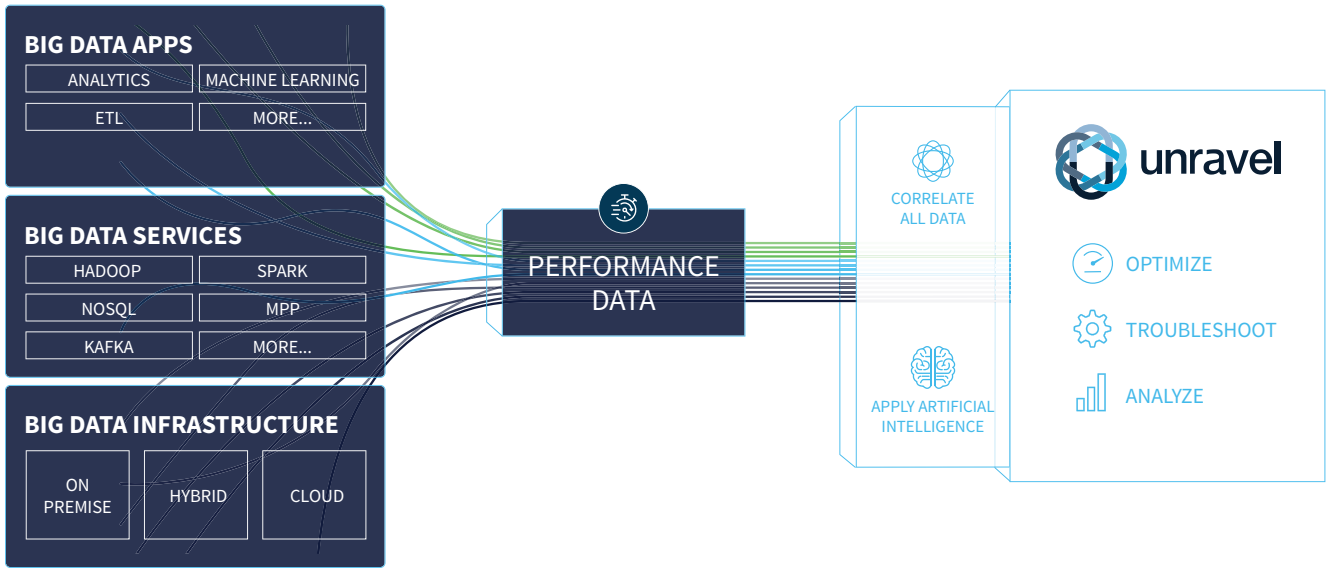
Storage and compute utilization need to be optimized to maximize performance as well as reduce cost. It is key to understand utilization with an application-first perspective in order to track all the processing steps the application depends on across systems. This capability provides DevOps with the ability to plan and optimize cluster resources, tell business stakeholders of the potential impact for any planned changes as well as accurate chargebacks.

Automated recommendations eliminate the need for trial-and-error in fixing problems during production, by providing a specific fix which DevOps or support teams can more readily deploy.

The Unravel solution

Unravel is Application Performance Management (APM) for big data. Unravel was designed to be the simplest way to manage performance and utilization of big data applications and big data platforms.

Unravel was designed specifically for big data platforms to be full-stack (centralize and correlate all performance data from infrastructure, services, to apps), intelligent (generate reports, insights and recommendations to optimize, troubleshoot, and analyze performance and utilization), and autonomous (e.g., automatically detect and resolve issues).



OPTIMIZE

Maximize application performance and cluster utilization:

- Applications
- Storage and compute
- Service performance
- Container sizing and utilization

TROUBLESHOOT

Get to the root cause of application performance issues instantly:

- Failure and slow-down
- Resource contention
- Service degradation
- Infrastructure health

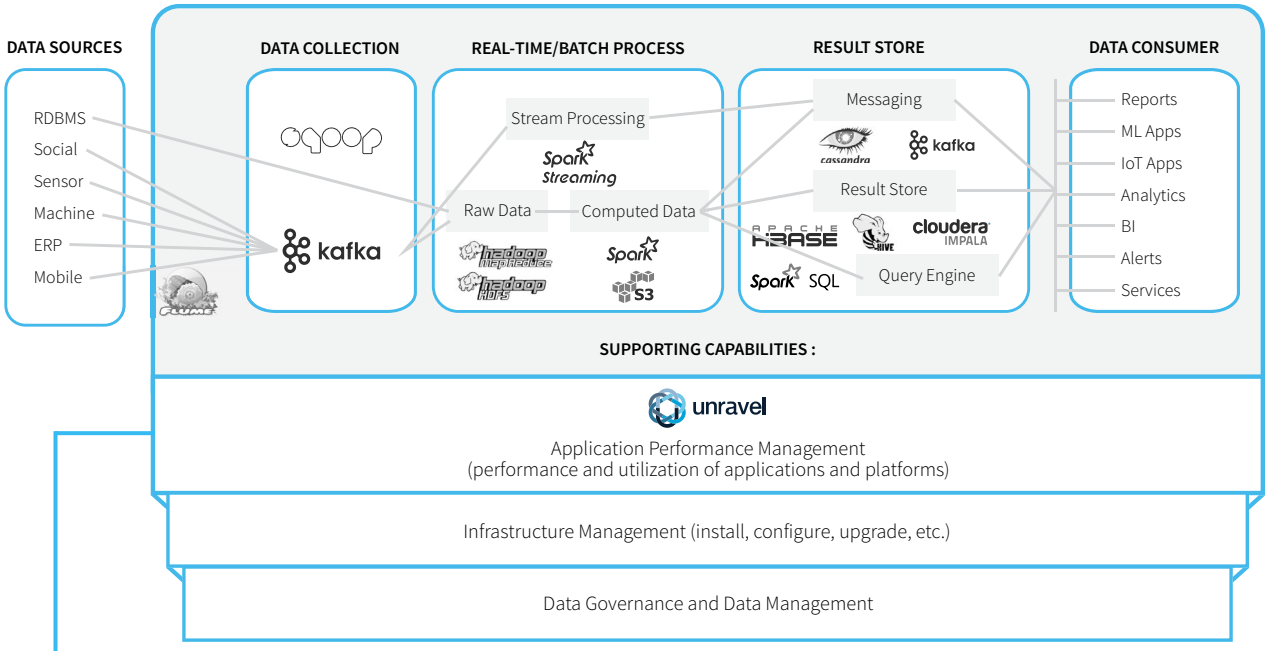
ANALYZE

Know everything from application to infrastructure:

- Accurate chargeback
- Actionable cost-savings
- 360° usage and performance
- Smart alerts and dashboards

Intelligent and autonomous APM with Unravel

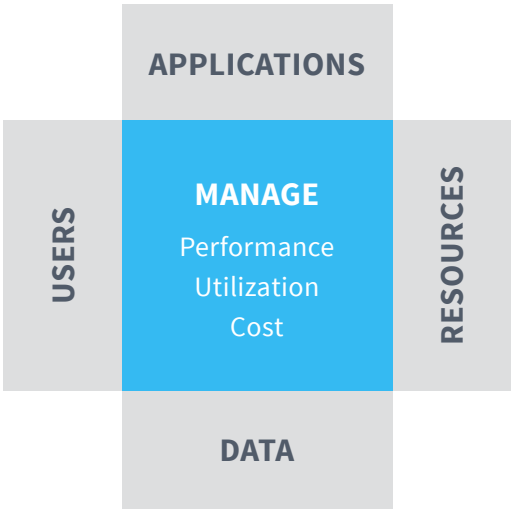
Unravel leverages Artificial Intelligence and Machine Learning to achieve the highest level of maturity for an APM solution to automatically deliver insights and recommendations just like an expert would:



PERFORMANCE INFO LEVELS	WITHOUT UNRAVEL	unravel
Recommendations	None	AI
Insights	None	Machine Learning
Correlated Information	None	Automated correlation
Logs	Manually sifting through logs	Automated parsing
Job KPIs	Fragmented metrics	Unified KPIs
Basic Performance KPIs	Platform-specific monitoring tool	Unified KPIs

A single pane of glass with a single view of performance and utilization

Unravel provides a correlated and fine-grained view of performance information across applications, users, resources, and data. Unravel enables Operations Teams to plan, operationalize and scale to maximize performance and utilization while reducing costs.



APPLICATIONS

- Plan and optimize app/workflow performance and utilization for SLA and ad-hoc apps
- Alert, troubleshoot and resolve performance issues by app; Report operational usage and chargeback by app across business units, departments and users
- Track app growth requirements to scale

USERS

- Plan and optimize infrastructure and system sizing based on user patterns
- Alert on rogue users and report on top users in the system
- Track user growth requirements to scale meet end-user requirements

RESOURCES

- Plan and optimize infrastructure and system sizing based on usage requirements and patterns
- Alert, troubleshoot and resolve infrastructure and service level issues
- Track user growth requirements to scale meet end-user requirements

DATA

- Plan and maximize storage infrastructure sizing based on data usage and access patterns
- Report fine-grained data usage for auditing, archiving, caching and managing data tiers
- Track data growth requirements to scale storage

APM is a strategic component of the big data architecture, and the business case for APM should be made early before too many applications are deployed in production. Based on customer case studies, Unravel APM can deliver tangible value across **three key areas**:

WHAT UNRAVEL DELIVERS

100% Apps On-Time

60% reduction in Infrastructure Costs

98% Reduction in Troubleshooting Time

STAY UP TO DATE

- Schedule a demo
- Register for a **free trial**
- Get the newsletter

www.unraveldata.com
hello@unraveldata.com
650.741.3442

unravel