

A Forrester Total Economic Impact™  
Study Commissioned By Azul  
August 2020

# The Total Economic Impact™ Of Azul Platform Prime

Cost Savings And Business Benefits  
Enabled By Azul Platform Prime

# Table Of Contents

<b>Executive Summary</b>	<b>1</b>
Key Findings	1
TEI Framework And Methodology	5
<b>The Platform Prime Customer Journey</b>	<b>6</b>
Interviewed Organizations	6
Key Challenges	6
Solution Requirements	7
Key Results	7
Composite Organization	8
<b>Analysis Of Benefits</b>	<b>9</b>
Increased Developer Efficiency Due To Code Simplification	9
Avoided High Performance Server Cost Due To Platform Prime's Extended Memory Capability	10
Avoided Cost Of Rearchitecting Performance Critical Solution Components	11
Reduced Developer Time For Troubleshooting And Recovery Of JVM- Resolvable Issues	12
Avoided JVM-Resolvable Outage/Failover Cost	13
Eliminated Cost Of Previous JVM	13
Unquantified Benefits	14
Flexibility	15
<b>Analysis Of Costs</b>	<b>16</b>
Platform Prime Licenses	16
POC, Cutover Testing, And Cutover Labor	17
<b>Financial Summary</b>	<b>18</b>
<b>Azul Platform Prime: Overview</b>	<b>19</b>
<b>Appendix A: Total Economic Impact</b>	<b>20</b>

**Project Director:**  
Eric Hall

## ABOUT FORRESTER CONSULTING

Forrester Consulting provides independent and objective research-based consulting to help leaders succeed in their organizations. Ranging in scope from a short strategy session to custom projects, Forrester's Consulting services connect you directly with research analysts who apply expert insight to your specific business challenges. For more information, visit [forrester.com/consulting](https://forrester.com/consulting).

© 2020, Forrester Research, Inc. All rights reserved. Unauthorized reproduction is strictly prohibited. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change. Forrester®, Technographics®, Forrester Wave, RoleView, TechRadar, and Total Economic Impact are trademarks of Forrester Research, Inc. All other trademarks are the property of their respective companies. For additional information, go to [forrester.com](https://forrester.com).

# Executive Summary

Platform Prime is Azul's flagship Java virtual machine (JVM). It provides a better runtime for Java workloads, on-premises or in the cloud, by eliminating latency issues that are common to Java applications. Azul commissioned Forrester Consulting to conduct a Total Economic Impact™ (TEI) study and examine the potential return on investment (ROI) enterprises may realize by deploying Platform Prime. The purpose of this study is to provide readers with a framework to evaluate the potential financial impact of Platform Prime on their organizations.

To better understand the benefits, costs, and risks associated with this investment, Forrester interviewed several customers with years of experience using Platform Prime.

Prior to using Platform Prime, customers addressed their critical performance Java workloads with specialized coding to attempt to prevent latency issues. They also expanded heap memory, by adding servers to improve service levels and unavoidable troubleshooting and recovery activities, when performance issues occurred. As a result, a considerable amount of developers' time was being redirected from writing code to performance tuning, troubleshooting, and debugging, while hardware or cloud resource costs were significantly rising. In addition, Java coding libraries and Java-based environments were not an option for some of the customers because they were not able, or did not want, to invest the time and resources to the specialized coding that was necessary to minimize performance issues.

During interviews, Platform Prime customers indicated that they greatly improved their Java performance, substituted increased memory instead of adding high cost servers, and wasted less time for developers on latency and troubleshooting issues that are typically associated with standard Java environments. More specifically, customers shared the following:

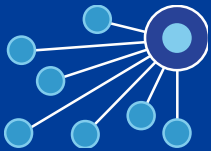
1. Pauses, jitter, and timeout due to garbage collection issues commonly associated with Java workloads were significantly reduced or eliminated.
2. Java startup and warm-up behaviors improved enough so that all customers met their SLAs for normal startups and reached peak performance more quickly.
3. Heap addressing threshold that is associated with Java environments has been removed, allowing customers to have heap memory sizes from 0.5 GB to over 1 TB.
4. Compiler-generated code runs either the same or faster than with customers' prior JVMs.

## Key Findings

**Quantified benefits.** The following risk-adjusted present value (PV) quantified benefits are representative of those experienced by the companies interviewed:

- › **Increased developer efficiency due to code simplification, valued at \$705K over three years.** Customers found that approximately 20% of development time for coding performance critical applications was spent writing specialized code to minimize garbage collection and startup issues. Platform Prime eliminated the garbage collection delays for all

## Benefits And Costs



Increased developer efficiency due to code simplification:

**\$705,124**



Avoided high performance server cost due to Platform Prime's extended memory capabilities:

**\$622,089**



Eliminated cost of prior JVM:

**\$515,414**



Platform Prime POC, cutover, and licensing costs:

**\$693,697**



**ROI**  
**224%**



**Benefits PV**  
**\$2.25 million**



**NPV**  
**\$1.55 million**



**Payback**  
**<3 months**

four customers without any specialized coding. Customers were also able to eliminate specialized coding and processes related to application warm-up.

- › **Avoided high performance server cost due to Platform Prime's extended memory capability, valued at \$622K over three years.** Platform Prime eliminates the 32 GB memory constraints common to other major JVMs, with customers addressing more than 1 TB of memory. For some customers, eliminating this memory constraint resulted in a reduction in the number of high-priced servers and Java licenses required for their workloads.
- › **Avoided cost of rearchitecting performance critical solution components, valued at \$92K over three years.** Customer demand growth, typically at 20% to 30% per year, was not only driving server growth but it was also driving periodic rearchitecting of solutions to ensure that performance requirements could be met. Platform Prime has enabled customers to have a simpler architecture that does not have to be redesigned every few years.
- › **Reduced developer time for troubleshooting and recovery of JVM-resolvable issues, valued at \$176K over three years.** Platform Prime has eliminated the need for customers to troubleshoot and recover from garbage collection-related freezes and failover issues, as well as warm-up issues that have material impact. Note that the issues frequently would appear in streaks, causing significant development disruptions for a period of time. This would frequently distract developers.
- › **Avoided JVM-resolvable outage/failover costs, valued at \$134K over three years.** Since implementing Platform Prime, none of the customers have had any outages or failovers. Outage magnitude varied among customers; the composite organization had a significant event every other month with an average cost of \$10,000 per event.
- › **Eliminated cost of previous JVM, valued at \$515K over three years.** Customers' move to Platform Prime meant that they benefited from a reduction in the licensing and support costs associated with their original JVM for existing Java servers, as well as for the Java servers that were no longer necessary to support future growth.

**Unquantified benefits.** The interviewed organizations experienced the following benefits, which are not quantified for this study:

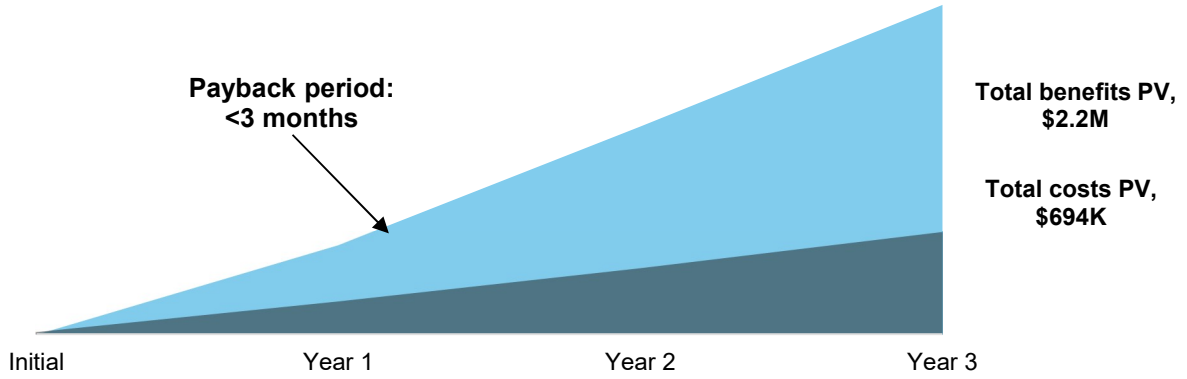
- › **Exceeding service-level agreements (SLAs).** Customers are consistently exceeding their service-level commitments. Note that two customers measure their performance in microseconds; and one customer previously failed to meet a 99.5% SLA while now performing between 99.99% and 99.999%.
- › **Use of open source libraries.** The use of open source software is deemed a necessity by the customers with extremely high transaction scale, and Platform Prime enables this by eliminating garbage collection and startup issues.
- › **Employee satisfaction.** Developers would prefer to build solutions versus writing performance-focused code to prevent JVM-related issues or, worse yet, troubleshoot and recover from JVM-related issues that occur.
- › **Partnership with Azul.** The four customers are each unique, with specific needs, and they all mentioned that Azul worked to understand their specific needs and to individually address those needs. They feel respected by Azul and consider them a partner.

**Costs.** The interviewed organizations experienced the following risk-adjusted PV costs:

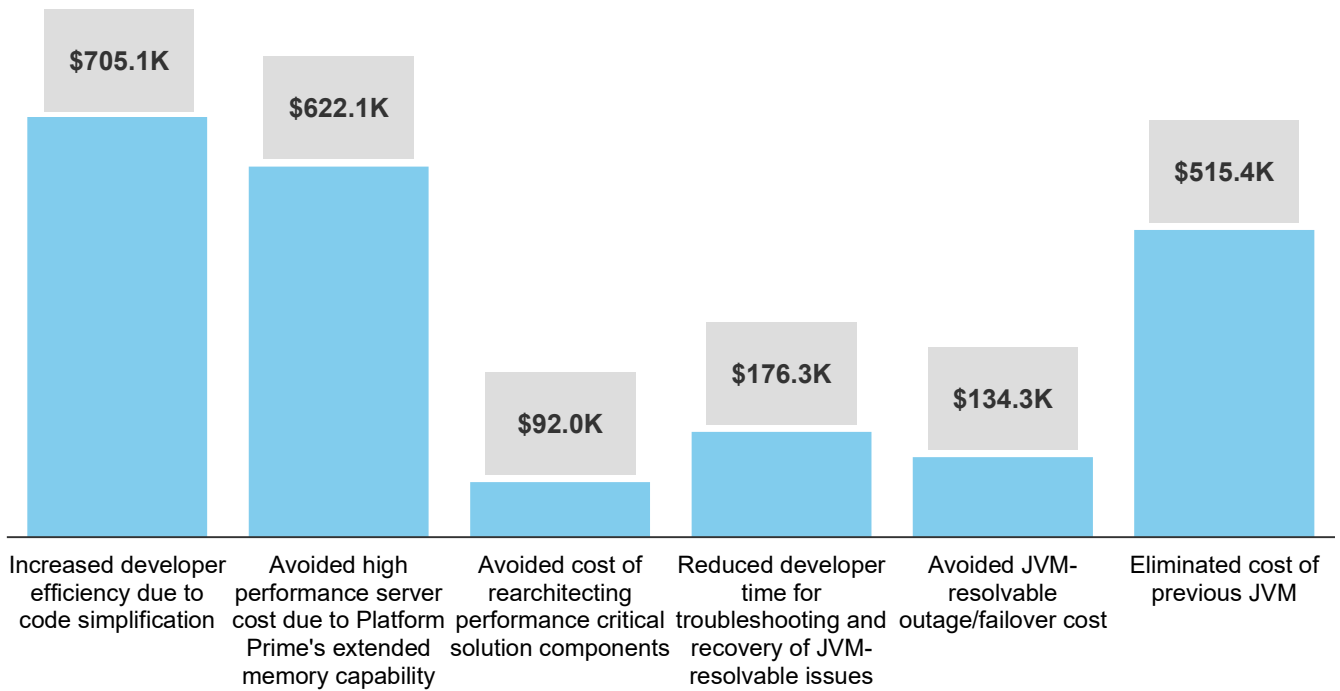
- › **Platform Prime licenses valued at \$677K over three years.** The composite organization used Platform Prime on performance critical servers.
- › **Proof of concept (POC), testing, and cutover valued at \$16,500 over three years.** Platform Prime is very easy to evaluate and implement for the typical organization. The composite organization accomplished this in under six weeks.

Forrester's interviews with four existing customers and subsequent financial analysis found that an organization based on these interviewed organizations experienced benefits of \$2,245,243 over three years versus costs of \$693,697, adding up to a net present value (NPV) of \$1,551,546 and an ROI of 224%.

## Financial Summary



### (Three-Year)



The TEI methodology helps companies demonstrate, justify, and realize the tangible value of IT initiatives to both senior management and other key business stakeholders.

## TEI Framework And Methodology

From the information provided in the interviews, Forrester has constructed a Total Economic Impact™ (TEI) framework for those organizations considering implementing Azul Platform Prime.

The objective of the framework is to identify the cost, benefit, flexibility, and risk factors that affect the investment decision. Forrester took a multistep approach to evaluate the impact that Azul Platform Prime can have on an organization:



### **DUE DILIGENCE**

Interviewed Azul stakeholders and Forrester analysts to gather data relative to Platform Prime.



### **CUSTOMER INTERVIEWS**

Interviewed four organizations using Platform Prime to obtain data with respect to costs, benefits, and risks.



### **COMPOSITE ORGANIZATION**

Designed a composite organization based on characteristics of the interviewed organizations.



### **FINANCIAL MODEL FRAMEWORK**

Constructed a financial model representative of the interviews using the TEI methodology and risk-adjusted the financial model based on issues and concerns of the interviewed organizations.



### **CASE STUDY**

Employed four fundamental elements of TEI in modeling Azul Platform Prime's impact: benefits, costs, flexibility, and risks. Given the increasing sophistication that enterprises have regarding ROI analyses related to IT investments, Forrester's TEI methodology serves to provide a complete picture of the total economic impact of purchase decisions. Please see Appendix A for additional information on the TEI methodology.

## DISCLOSURES

Readers should be aware of the following:

This study is commissioned by Azul and delivered by Forrester Consulting. It is not meant to be used as a competitive analysis.

Forrester makes no assumptions as to the potential ROI that other organizations will receive. Forrester strongly advises that readers use their own estimates within the framework provided in the report to determine the appropriateness of an investment in Azul Platform Prime.

Azul reviewed and provided feedback to Forrester, but Forrester maintains editorial control over the study and its findings and does not accept changes to the study that contradict Forrester's findings or obscure the meaning of the study.

Azul provided the customer names for the interviews but did not participate in the interviews.

# The Platform Prime Customer Journey

## BEFORE AND AFTER THE PLATFORM PRIME INVESTMENT

### Interviewed Organizations

For this study, Forrester conducted four interviews with Azul Platform Prime customers. Interviewed customers include the following:

INDUSTRY	REGION	INTERVIEWEE	JAVA SERVERS
Financial technology	Global	Head of software	250 Platform Prime servers
Web ad technology	Global	VP IT	1,750 Platform Prime servers; 2,000 other Java servers
Financial services	Global	Chief platform architect	1,000 Platform Prime servers
Online shopping	Global	Java architect	100 Platform Prime servers; 10,000+ other Java servers

### Key Challenges

The four interviewees had the same primary challenge: to eliminate garbage collection and warm-up issues within their Java environments requiring low latency. Their approaches to consistent performance all had the common element of higher costs:

- › **Development time necessary to produce special code to minimize garbage collection issues or accelerate warm-up process.** Developers working on performance critical applications were spending 20% of their time writing code with the sole purpose of minimizing performance issues.
- › **Purchasing of high cost servers; customers living within their legacy JVM's limited memory address constraint.** Customers were forced to buy servers only to expand their available memory. This happened due to memory amounts being crucial in maintaining consistent performance. Customers' legacy JVMs only allowed them to address 32 GB of memory per server.
- › **Troubleshooting and recovering from poor performance events; attempting to code to reduce future issues.** With their previous JVMs, customers would have performance issues; developers would resolve the immediate issue and attempt to modify code to reduce the probability that the issue would repeat itself, or to minimize its magnitude.
- › **Rearchitecting to higher cost development environment.** Customers would rearchitect within the Java environment or, in some cases, customers evaluated changing to other development environments. In all cases, the development environment would be more costly.

"The big drive was to remove the unexpected stalls so that we could be consistently fast. We measure pauses in the hundreds of microseconds, not milliseconds — that level of performance is a big win for us."

*Head of software, financial technology*





## Solution Requirements

The interviewed organizations searched for a solution that could:

- › Resolve their garbage collection issues so that they would consistently meet their SLAs. This will eliminate development time to attempt to prevent garbage collection issues, as well as eliminate use of developer time to troubleshoot and resolve these issues.
- › Reduce warm-up issues, time to full performance, and sub-SLA performance. This will eliminate development time to attempt to prevent these issues as well as eliminate use of developer time to troubleshoot and resolve these issues.
- › Remove JVM-based memory constraints which will allow a shift from horizontal memory scaling across servers to vertical memory scaling within servers, saving a significant hardware cost. This is a necessary component to resolving garbage collection issues.
- › JVM licensing cost increases reduce the threshold for changing JVMs.

“The plan for just one service included a 40% increase in servers for the upcoming year and similar increases the following year. Something had to change.”

*Java architect, online shopping*



## Key Results

The interviews revealed that key results from the Platform Prime investment include:

- › **Garbage collection issues are practically eliminated.** The chief platform architect for the financial service firm described it best: “We work in microseconds. With Platform Prime, we are meeting our SLAs between 99.99% and 99.999% of the time.”
- › **Warm-up issues are practically eliminated.** Customers see a shorter time to full production performance and faster response times during startup. As described by the Java architect at the online shopping company: “We went from having performance penalties throughout the first 5 minutes of the warm-up process to meeting our service-level objectives with Platform Prime. These performance issues affected prospective customers and were considered material.”
- › **Hardware costs are reduced.** Platform Prime allowed customers to transition from many servers with limited memory to fewer servers with practically unlimited memory. The VP of IT at the web ad technology company, with 1,750 servers, noted that the server approach would cost them 20% to 30% more than the memory approach that Platform Prime enables.
- › **Developers are writing code focused on creating solutions.** Developers are no longer writing code to attempt to avoid garbage collection and startup issues, nor are they frequently troubleshooting and recovering from negative performance events.
- › **There is no need to rearchitect environments.** Rearchitecting considerations that were eliminated included adding servers, switching to commercial packages from Java, and transitioning away from bigger heap sizes. The Java architect at the online shopping company said, “Platform Prime eliminated the need to rearchitect one of their systems, 100% eliminating a four-person, six-month project.”
- › **Platform Prime licensing is partially offset by JVM cost increases.** Recent price changes in the market lowered the threshold for selecting Platform Prime.

“Developers now focus on implementing new features and functionality without work arounds for JVM-related penalties. That was code that the developers no longer have to write, no longer have to maintain, and no longer have to troubleshoot.”

*Java architect, online shopping*



“We reduced our infrastructure cost 20% to 30% by converting to Platform Prime, while meeting our service levels. We now have around 2,000 Platform Prime servers, are meeting our service levels, and are very happy with where we are.”

*VP IT, web ad technology*



## Composite Organization

Based on the interviews, Forrester constructed a TEI framework, a composite company, and an associated ROI analysis that illustrates the areas financially affected. The composite organization is representative of the four companies that Forrester interviewed and is used to present the aggregate financial analysis in the next section. The composite organization that Forrester synthesized from the customer interviews has the following characteristics:

- › The composite organization has performance critical workloads on 20% of its servers (or an organization with high threshold SLAs). There are 200 performance critical servers.
- › The performance critical workloads are growing 20% per year.
- › The organization has 50 developers.
- › Performance critical development work comprises 20% of development activities.



### Key assumptions:

- 200 performance critical servers
- 20% volume growth
- 50 developers
- 20% performance critical development

# Analysis Of Benefits

## QUANTIFIED BENEFIT DATA AS APPLIED TO THE COMPOSITE

Total Benefits						
REF.	BENEFIT	YEAR 1	YEAR 2	YEAR 3	TOTAL	PRESENT VALUE
Atr	Increased developer efficiency due to code simplification	\$270,000	\$286,200	\$297,000	\$853,200	\$705,124
Btr	Avoided high performance server cost due to Platform Prime's extended memory capability	\$0	\$360,000	\$432,000	\$792,000	\$622,089
Ctr	Avoided cost of rearchitecting performance critical solution components	\$101,250	\$0	\$0	\$101,250	\$92,045
Dtr	Reduced developer time for troubleshooting and recovery of JVM-resolvable issues	\$67,500	\$71,550	\$74,250	\$213,300	\$176,281
Etr	Avoided JVM-resolvable outage/failover cost	\$54,000	\$54,000	\$54,000	\$162,000	\$134,290
Ftr	Eliminated cost of previous JVM	\$172,800	\$207,360	\$248,832	\$628,992	\$515,414
	Total benefits (risk-adjusted)	\$665,550	\$979,110	\$1,106,082	\$2,750,742	\$2,245,243

### Increased Developer Efficiency Due To Code Simplification

All customers saw increased developer productivity after switching low latency workloads to Platform Prime. Platform Prime simplified coding by eliminating specialized coding required to minimize latency issues and allowed customers to use open source libraries to further reduce custom coding.

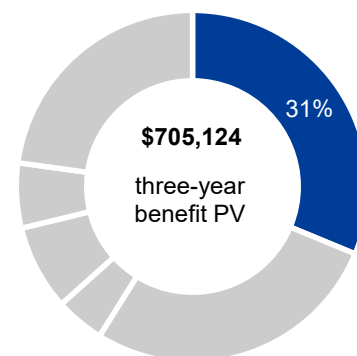
As described by the head of software at the financial technology company, "Our developers now code more simply because they don't have to worry about garbage collection — coding is faster, simpler, easier to maintain, and less prone to bugs."

Explaining that they process tens of billions of transactions a day, the VP of IT at the web ad technology company said: "To process at our transaction scale and remain cost effective requires use of open source software, much of it [is] based upon Java. Platform Prime provides a huge difference in performance and the ability to scale."

Forrester identified potential risks, including both the magnitude of the developer time savings as well as the potential of developers taking inappropriate shortcuts.

To account for these risks, Forrester adjusted this benefit downward by 10%, yielding a three-year, risk-adjusted total PV of \$705,124.

The table above shows the total of all benefits across the areas listed below, as well as present values (PVs) discounted at 10%. Over three years, the composite organization expects risk-adjusted total benefits to be a PV of more than \$2.2 million.



**Increased developer efficiency due to code simplification: 31% of total benefits**

Impact risk is the risk that the business or technology needs of the organization may not be met by the investment, resulting in lower overall total benefits. The greater the uncertainty, the wider the potential range of outcomes for benefit estimates.

### Increased Developer Efficiency Due To Code Simplification: Calculation Table

REF.	METRIC	CALC.	YEAR 1	YEAR 2	YEAR 3
A1	Number of developers		50	53	55
A2	Percent performance critical development		20%	20%	20%
A3	Development productivity due to Platform Prime		20%	20%	20%
A4	Cost per developer		\$150,000	\$150,000	\$150,000
At	Increased developer efficiency due to code simplification	$A1 \cdot A2 \cdot A3 \cdot A4$	\$300,000	\$318,000	\$330,000
	Risk adjustment	↓10%			
Atr	Increased developer efficiency due to code simplification (risk-adjusted)		\$270,000	\$286,200	\$297,000

### Avoided High Performance Server Cost Due To Platform Prime's Extended Memory Capability

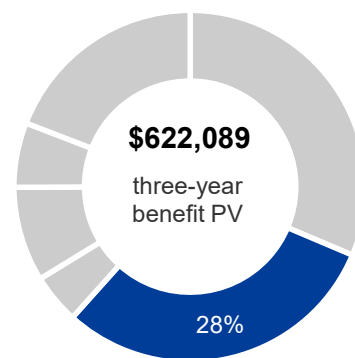
Heap memory size plays a significant role in avoiding garbage collection issues, both with the original JVM and with Platform Prime. The original JVM for all of the customers could only address 32 GB of heap memory per server, while with Platform Prime some of the customers are addressing 1.5 TB of heap memory per server. The chief platform architect at the financial services firm noted, "We moved from horizontal scaling to vertical scaling."

By adding memory to existing servers, Platform Prime has significantly reduced hardware costs for its customers. One rapidly growing customer was able to avoid adding 50 new servers for over the first two years that they had Platform Prime.

For the composite organization, Forrester assumes that the server growth would have been 20% without Platform Prime.

- › Forrester identified potential risks, such as lower growth or lower server costs.

To account for these risks, Forrester adjusted this benefit downward by 10%, yielding a three-year, risk-adjusted total PV of \$622,089.



Avoided high performance server cost due to Platform Prime's extended memory capability: **28%** of total benefits

### Avoided High Performance Server Cost Due To Platform Prime's Extended Memory Capability: Calculation Table

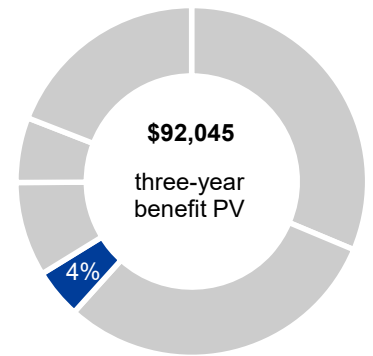
REF.	METRIC	CALC.	YEAR 1	YEAR 2	YEAR 3
B1	Additional server growth without Platform Prime		0	40	48
B2	Avoided server cost		\$10,000	\$10,000	\$10,000
Bt	Avoided high performance server cost due to Platform Prime's extended memory capability	$B1 \cdot B2$	\$0	\$400,000	\$480,000
	Risk adjustment	↓10%			
Btr	Avoided high performance server cost due to Platform Prime's extended memory capability (risk-adjusted)		\$0	\$360,000	\$432,000

## Avoided Cost Of Rearchitecting Performance Critical Solution Components

Within their existing architecture, some customers were seeing growth with associated performance issues that would have required rearchitecting their performance critical solution components. All rearchitecting options were expensive, ranging from adding additional servers to gain more memory, to implementing code optimization schemes, and to the extreme of moving to a high-cost commercial development platform outside of Java.

The composite organization stayed within Java, rearchitecting the solution with a combination of server additions and coding strategies. A team of three developers worked full time on rearchitecting the solution over a three-month period.

Forrester identified a potential risk of an overestimate of the rearchitecting effort. To account for this risk, Forrester adjusted this benefit downward by 10%, yielding a three-year, risk-adjusted total PV of \$92,045.



Avoided cost of rearchitecting performance critical solution components: **4%** of total benefits

### Avoided Cost Of Rearchitecting Performance Critical Solution Components: Calculation Table

REF.	METRIC	CALC.	YEAR 1	YEAR 2	YEAR 3
C1	Developer team (FTEs)		3		
C2	Months for rearchitecture		3		
C3	Cost per developer		\$150,000		
Ct	Avoided cost of rearchitecting performance critical solution components	$C1 * C2 / 12 * C3$	\$112,500		
	Risk adjustment	↓10%			
Ctr	Avoided cost of rearchitecting performance critical solution components (risk-adjusted)		\$101,250		

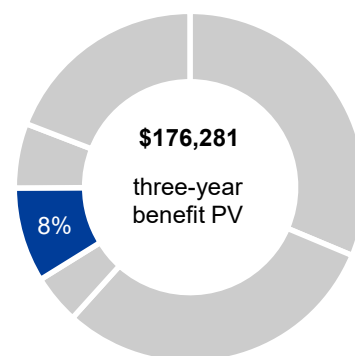
## Reduced Developer Time For Troubleshooting And Recovery Of JVM-Resolvable Issues

All customers have eliminated, or practically eliminated, all performance issues that can be associated with the JVM. Prior to Platform Prime, a fair amount of developer hours was used for troubleshooting JVM-related issues and resolving them.

The composite organization has 20% of developers spend part of their time troubleshooting and resolving JVM-related issues, which was common among the four customers. Those designated developers spent 5% of their time working on troubleshooting and resolving issues, instead of on value-add development work.

Forrester identified the potential risks that were associated with hardware purchases or coding actions which reduce troubleshooting and recovery times.

To account for these risks, Forrester adjusted this benefit downward by 10%, yielding a three-year, risk-adjusted total PV of \$176,281.



Reduced developer time for troubleshooting and recovery of JVM-resolvable issues: **8% of total benefits**

**Reduced Developer Time For Troubleshooting And Recovery Of JVM-Resolvable Issues: Calculation Table**

REF.	METRIC	CALC.	YEAR 1	YEAR 2	YEAR 3
D1	Number of developers		50	53	55
D2	Percent of developers performing troubleshooting and recovery activities		20%	20%	20%
D3	Troubleshooting and recovery time per developer		5%	5%	5%
D4	Cost per developer		\$150,000	\$150,000	\$150,000
Dt	Reduced developer time for troubleshooting and recovery of JVM-resolvable issues	$D1 * D2 * D3 * D4$	\$75,000	\$79,500	\$82,500
	Risk adjustment	↓10%			
Dtr	Reduced developer time for troubleshooting and recovery of JVM-resolvable issues (risk-adjusted)		\$67,500	\$71,550	\$74,250

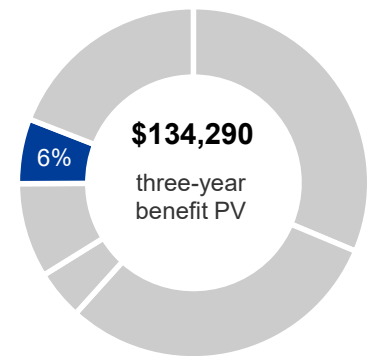
## Avoided JVM-Resolvable Outage/Failover Cost

Customers had outages, failovers, pauses, etc., that led to additional cost, revenue loss, customer satisfaction issues, or other problems. Their relevance varied per use case. As the chief platform architect with the financial services firm noted, “We work in microseconds — a pause of 3 seconds will lose 15K to 30K transactions, which is unacceptable.” The Java architect at the online shopping company mentioned, “One system experienced frequent failovers due to garbage collection issues that virtually disappeared when we implemented Platform Prime.”

The composite organization has one performance event every other month, which has an average cost of \$10,000 per event.

- › Forrester recognizes that the frequency of these events and the average cost per event may vary considerably per organization.

To account for these risks, Forrester adjusted this benefit downward by 10%, yielding a three-year, risk-adjusted total PV of \$134,290.



Avoided JVM-resolvable outage/failover cost: **6%** of total benefits

### Avoided JVM-Resolvable Outage/Failover Cost: Calculation Table

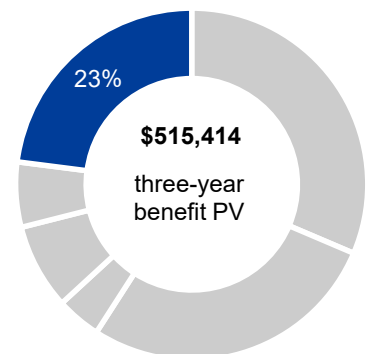
REF.	METRIC	CALC.	YEAR 1	YEAR 2	YEAR 3
E1	Number of JVM-resolvable outages/failovers		6	6	6
E2	Cost per outage		\$10,000	\$10,000	\$10,000
Et	Avoided JVM-resolvable outage/failover cost	E1*E2	\$60,000	\$60,000	\$60,000
	Risk adjustment	↓10%			
Etr	Avoided JVM-resolvable outage/failover cost (risk-adjusted)		\$54,000	\$54,000	\$54,000

## Eliminated Cost Of Previous JVM

Implementing Platform Prime means that existing JVM licenses for the performance critical servers are eliminated, with some future planned servers being no longer necessary. Note that it is very possible that some existing servers are no longer necessary due to the shift from heap memory expanding vertically versus horizontally, but they were not included in this composite organization’s server changes.

The composite organization converted 200 servers from the previous JVM to Platform Prime. Annual server growth was 20%. Pricing for the previous servers was based upon approximate volume pricing for the original 1,000 servers.

Forrester identified potential risks, such as other pricing discounts. To account for these risks, Forrester adjusted this benefit downward by 10%, yielding a three-year, risk-adjusted total PV of \$515,414.



Eliminated cost of previous JVM: **23%** of total benefits

## Eliminated Cost Of Previous JVM: Calculation Table

REF.	METRIC	CALC.	YEAR 1	YEAR 2	YEAR 3
F1	Number of servers moved to Platform Prime		200	240	288
F2	Eliminated monthly license cost (assuming four CPUs per server)		\$960	\$960	\$960
Ft	Eliminated cost of previous JVM	F1*F2	\$192,000	\$230,400	\$276,480
	Risk adjustment	↓10%			
Ftr	Eliminated cost of previous JVM (risk-adjusted)		\$172,800	\$207,360	\$248,832

## Unquantified Benefits

In addition to the benefits of implementing Outreach detailed above, numerous additional benefits do not have specific financial implications. The customers benefited in the following ways:

- › **Exceeding service-level agreements.** With Platform Prime as the JVM for performance critical workloads, the four interviewed customers are able to meet their service-level commitments. Prior to Platform Prime, service-level commitments were frequently not met even though the customers were doing special coding to reduce or eliminate performance issues. The customers were building out larger server farms to increase their heap memory amounts, and they placed significant focus on troubleshooting and recovery activities, frequently rearchitecting solutions in an attempt to meet their service-level commitments. Customers literally eliminated all of these activities with Platform Prime.
- › **Use of Java open source libraries.** Performance critical applications frequently can't use Java-based open source libraries because they don't eliminate, or minimize, garbage collection and startup issues. By eliminating these issues, Platform Prime provides low latency results for these libraries. One customer, the VP of IT at the web ad technology company said, "Platform Prime allows us to effectively use Cassandra, Hadoop, HDFS, and other open source projects." He added that, "We have an application that runs on the Hadoop community version that won't even work anymore without Platform Prime and, with Platform Prime, it has acceptable performance."
- › **Employee satisfaction.** Customers said that developers were less productive, and they found efforts to minimize, or resolve, garbage collection performance issues and warm-up performance issues. Developers can now write more value-add code while not worrying about whether performance issues will arise.
- › **Partnership with Azul.** The four customers all had very positive things to say about their relationship with Azul. To sum it up, the chief platform architect with the financial services firm said: "I have gotten superb support from Azul. If I could get other vendors to be like Azul then I would be much happier dealing with vendors — a ten-fold difference in support."

"Platform Prime is like pressing a turbo button."

*Head of software, financial technology*





## Flexibility

The value of flexibility is clearly unique to each customer, and the measure of its value varies from organization to organization. There are multiple scenarios in which a customer might choose to implement Platform Prime and later realize additional uses and business opportunities, including:

- › Most notable, customers can consider using open source libraries that they hadn't considered before to reduce licensing costs and development time.
- › Customers can begin testing, then expanding their environments to levels that they previously hadn't considered. For example, some of the customers have expanded heap memory into 100+ GB, while one of the interviewees noted that they have extended their heap size to 1.5 TB. The VP of IT at the web ad technology company said, "Cassandra recommends that each node of a cluster not exceed 1 TB — we operate clusters with 10 TB of data."
- › Recent increases in competitor licensing prices may make expanding the use of Platform Prime something to consider.

Flexibility, as defined by TEI, represents an investment in additional capacity or capability that could be turned into business benefit for a future additional investment. This provides an organization with the "right" or the ability to engage in future initiatives but not the obligation to do so.

# Analysis Of Costs

## QUANTIFIED COST DATA AS APPLIED TO THE COMPOSITE

Total Costs							
REF.	COST	INITIAL	YEAR 1	YEAR 2	YEAR 3	TOTAL	PRESENT VALUE
Gtr	Platform Prime licenses	\$0	\$227,040	\$272,448	\$326,938	\$826,426	\$677,197
Htr	POC, cutover testing, and cutover labor	\$16,500	\$0	\$0	\$0	\$16,500	\$16,500
	Total costs (risk-adjusted)	\$16,500	\$227,040	\$272,448	\$326,938	\$842,926	\$693,697

### Platform Prime Licenses

Platform Prime pricing considers volume, term, payment terms, reference discounting, among other things. The composite organization used a price of \$1,032 per server.

Forrester identified potential risks, such as greater growth or price variations.

To account for these risks, Forrester adjusted this cost upward by 10%, yielding a three-year, risk-adjusted total PV of \$677,197.

The table above shows the total of all costs across the areas listed below, as well as present values (PVs) discounted at 10%. Over three years, the composite organization expects risk-adjusted total costs to be a PV of more than \$690K.

Platform Prime Licenses: Calculation Table						
REF.	METRIC	CALC.	INITIAL	YEAR 1	YEAR 2	YEAR 3
G1	Count performance critical servers			200	240	288
G2	Platform Prime license per server			\$1,032	\$1,032	\$1,032
Gt	Platform Prime licenses	G1*G2	\$0	\$206,400	\$247,680	\$297,216
	Risk adjustment	↑10%				
Gtr	Platform Prime licenses (risk-adjusted)		\$0	\$227,040	\$272,448	\$326,938

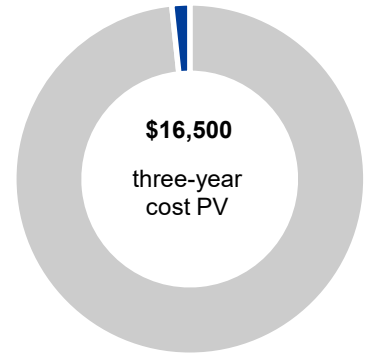
## POC, Cutover Testing, And Cutover Labor

Substituting one JVM for another is technically quite simple. A POC with Platform Prime can be as simple as doing feature-function testing in a development area, followed by swapping Platform Prime into a production server, and then adjusting load with a load balancer, measuring either before/after metrics or comparable servers. Note that testing varied among customers; for example, the two customers that measure performance in microseconds perform more significant testing with each new version of Platform Prime, tailoring the configuration so that feature use is balanced with its general performance overhead.

Cutover may be done all at once or in phases. Either way, the effort is not significant since the POC and testing covered feature-function and load performance.

The composite organization performed the POC described above, basic testing, and a straight-forward cutover. The overall effort took under six weeks. Forrester identified potential risks, such as the need for additional testing.

To account for these risks, Forrester adjusted this cost upward by 10%, yielding a three-year risk-adjusted total PV of \$16,500.



POC, cutover testing, and cutover labor: **2%** of total costs

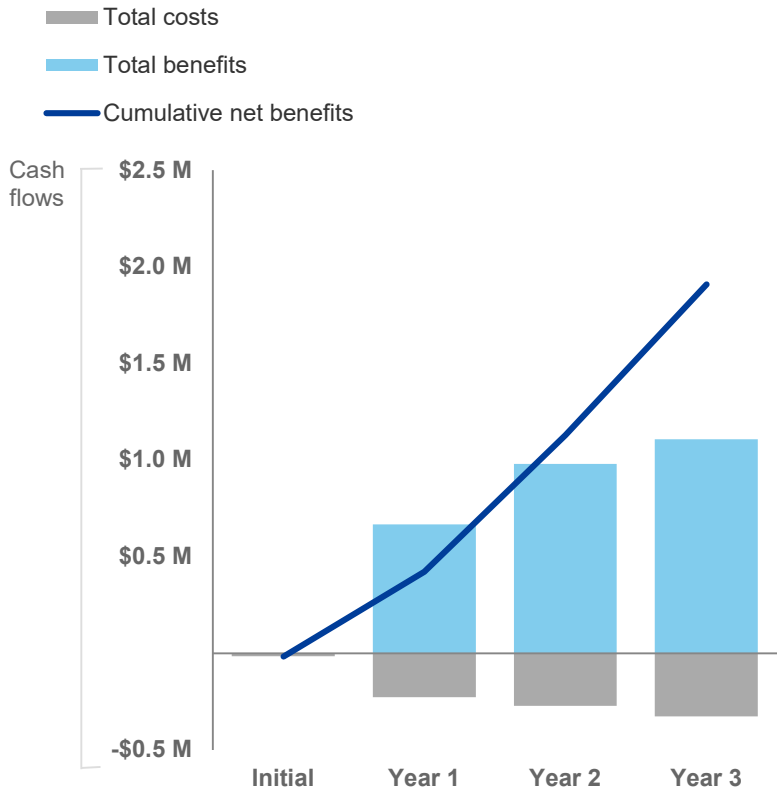
**POC, Cutover Testing, And Cutover Labor: Calculation Table**

REF.	METRIC	CALC.	INITIAL	YEAR 1	YEAR 2	YEAR 3
H1	Developer FTEs for POC, cutover testing and cutover		0.10			
H2	Cost per developer		\$150,000			
Ht	POC, cutover testing, and cutover labor	H1*H2	\$15,000	\$0	\$0	\$0
	Risk adjustment	↑10%				
Htr	POC, cutover testing, and cutover labor (risk-adjusted)		\$16,500	\$0	\$0	\$0

# Financial Summary

## CONSOLIDATED THREE-YEAR RISK-ADJUSTED METRICS

### Cash Flow Chart (Risk-Adjusted)



The financial results calculated in the Benefits and Costs sections can be used to determine the ROI, NPV, and payback period for the composite organization's investment. Forrester assumes a yearly discount rate of 10% for this analysis.



These risk-adjusted ROI, NPV, and payback period values are determined by applying risk-adjustment factors to the unadjusted results in each Benefit and Cost section.

### Cash Flow Table (Risk-Adjusted)

	INITIAL	YEAR 1	YEAR 2	YEAR 3	TOTAL	PRESENT VALUE
Total costs	(\$16,500)	(\$227,040)	(\$272,448)	(\$326,938)	(\$842,926)	(\$693,697)
Total benefits	\$0	\$665,550	\$979,110	\$1,106,082	\$2,750,742	\$2,245,243
Net benefits	(\$16,500)	\$438,510	\$706,662	\$779,144	\$1,907,816	\$1,551,546
ROI						224%
Payback period						<3 months

# Azul Platform Prime: Overview

The following information is provided by Azul. Forrester has not validated any claims and does not endorse Azul or its offerings.

Azul Platform Prime is a robust, proven, highly scalable Java Virtual Machine (JVM) that matches the needs of today's business, from newly developed microservice-based applications and modern Java-based infrastructure, to traditional monoliths, services, and legacy systems. Just as Azul itself is the ideal cornerstone for a best-of-breed Java strategy, Azul Platform Prime is the best JVM choice for all Java workloads. Today, Platform Prime use cases include an array of SaaS, cloud-based, or on-premises deployments, with application domains ranging from online retail and insurance portals to multiuser gaming platforms, and uses ranging from API gateways through business logic, stream processing, and big data.

Applications that have been proven to run more efficiently with Platform Prime include widely used Java-based infrastructure software, including high-value open source technologies like Cassandra, Elastic, Solr, Kafka, Hadoop, and Spark. Platform Prime is fully compatible with Kubernetes and Docker-based workloads as well. Platform Prime has also been widely deployed in low-latency systems for capital markets — and ensures smooth operation anywhere predictable, glitch-free Java is essential.

Platform Prime enables Java developers to make more efficient and effective use of server resources or Cloud instances, driving down CapEx and OpEx by delivering more transactions at a given service level from the same infrastructure, or allowing operations teams to deploy 30% to 50% fewer servers or cloud instances while delivering increased throughput and consistent, jitter-free performance. These infrastructure savings are particularly pronounced with Azul customers who rely on Apache Spark, Cassandra, and Kafka as well as critical open source search technologies like Lucene, Solr, and particularly Elastic. Investing in Platform Prime can cut your cloud bill in half or allow critical infrastructure to be redeployed or reprovisioned for future growth.

Whether your application requires human-scale response times or is machine-scale, measuring peak response time in milliseconds or even microseconds, Platform Prime allows you to meet even the most demanding service level agreements without recoding, rearchitecting, or depending upon JVM tuning experts — helping you take advantage of new business opportunities faster and with lower operating costs.

Azul Platform Prime is simple to install and requires no coding changes to existing applications. You don't even need to recompile. Because Platform Prime has been optimized for today's servers, configuration and setup are typically reduced to just a few parameters, instead of the myriad of JVM tuning flags necessary to reach peak performance that characterize many Java production environments. Simply point your application or startup scripts to Platform Prime, and you're running on the most robust, scalable JVM with the fastest time-to-market for any business application.

For additional information and visit [www.azul.com/products/prime](http://www.azul.com/products/prime).

# Appendix A: Total Economic Impact

Total Economic Impact is a methodology developed by Forrester Research that enhances a company's technology decision-making processes and assists vendors in communicating the value proposition of their products and services to clients. The TEI methodology helps companies demonstrate, justify, and realize the tangible value of IT initiatives to both senior management and other key business stakeholders.

## Total Economic Impact Approach



**Benefits** represent the value delivered to the business by the product. The TEI methodology places equal weight on the measure of benefits and the measure of costs, allowing for a full examination of the effect of the technology on the entire organization.



**Costs** consider all expenses necessary to deliver the proposed value, or benefits, of the product. The cost category within TEI captures incremental costs over the existing environment for ongoing costs associated with the solution.



**Flexibility** represents the strategic value that can be obtained for some future additional investment building on top of the initial investment already made. Having the ability to capture that benefit has a PV that can be estimated.



**Risks** measure the uncertainty of benefit and cost estimates given: 1) the likelihood that estimates will meet original projections and 2) the likelihood that estimates will be tracked over time. TEI risk factors are based on "triangular distribution."

The initial investment column contains costs incurred at "time 0" or at the beginning of Year 1 that are not discounted. All other cash flows are discounted using the discount rate at the end of the year. PV calculations are calculated for each total cost and benefit estimate. NPV calculations in the summary tables are the sum of the initial investment and the discounted cash flows in each year. Sums and present value calculations of the Total Benefits, Total Costs, and Cash Flow tables may not exactly add up, as some rounding may occur.



### Present value (PV)

The present or current value of (discounted) cost and benefit estimates given at an interest rate (the discount rate). The PV of costs and benefits feed into the total NPV of cash flows.



### Net present value (NPV)

The present or current value of (discounted) future net cash flows given an interest rate (the discount rate). A positive project NPV normally indicates that the investment should be made, unless other projects have higher NPVs.



### Return on investment (ROI)

A project's expected return in percentage terms. ROI is calculated by dividing net benefits (benefits less costs) by costs.



### Discount rate

The interest rate used in cash flow analysis to take into account the time value of money. Organizations typically use discount rates between 8% and 16%.



### Payback period

The breakeven point for an investment. This is the point in time at which net benefits (benefits minus costs) equal initial investment or cost.