VINCA ProFlow: A Multi-tenant SaaS Business Process System Postprint

Authors: Li Houfu, Sun Junyi, Ma Qiang, Wang Lei, Zhao Zhuofeng

Date: 2017-03-10T00:00:00+00:00

Abstract

Business processes, as an effective way to support business collaboration, play an important role in internal refined management and upstream/downstream partner integration within enterprises, enabling decision-makers to grasp business data in real time. However, building business process systems often requires substantial investment in hardware/software procurement and staffing. This paper addresses the contradiction between "demand" and "investment capacity" currently faced by small and medium-sized enterprises (SMEs) in building business process management systems, and proposes a multi-tenant SaaS business process system called VINCA ProFlow, which emphasizes utilizing cloud infrastructure capabilities to provide process application hosting and runtime environments, offering SMEs an on-demand process service (BPM-as-a-Service). While each tenant shares the same process service instance, they can also enjoy exclusive independent data spaces, ensuring effective data isolation and privacy protection between tenants. The paper specifically discusses the research motivation, key issues, and solution approaches of VINCA ProFlow.

Full Text

Preamble

Vol. 9 No. 1
Information Technology Letters
VINCA ProFlow: A Multi-Tenant SaaS Business Process System
Li Houfu, Sun Junyi, Ma Qiang, Wang Lei, Zhao Zhuofeng

Abstract

Business processes serve as an effective means to support business collaboration, playing a crucial role in the refined management of internal operations and the integration of upstream and downstream partners, enabling decision-makers to grasp operational data in real time. However, constructing a business process system often requires substantial investment in hardware/software procurement and personnel allocation. This paper addresses the contradiction between "demand" and "investment capacity" that currently exists for small and medium-sized enterprises (SMEs) in building business process management systems. We propose VINCA ProFlow, a multi-tenant SaaS business process system that leverages cloud infrastructure capabilities to provide process application hosting and runtime environments, offering SMEs an on-demand process service (BPM-as-a-Service). While tenants share the same process service instance, each enjoys exclusive access to an independent data space, ensuring effective data isolation and privacy protection among tenants. The paper specifically discusses the research motivations, key challenges, and solution approaches of VINCA ProFlow.

Keywords: cloud computing, software as a service, multi-tenancy, business process software service

1. Introduction

Process management technology can provide a supporting environment for integrated applications, helping enterprises and organizations gain visibility and control over business processes. It is widely used in office automation, logistics, scientific computing, emergency response coordination, and other domains. However, the implementation cost of traditional process management software systems—including server hardware, process management software procurement costs, and corresponding technical personnel expenses—remains prohibitively high, deterring numerous startups and SMEs. On the other hand, some enterprises and organizations may experience sudden process management needs during specific periods. The temporary nature of these demands, combined with high investment requirements, discourages them from investing in process application development, forcing them to rely on manual process execution instead. This leads to data delays and inconsistencies across multiple related organizations, reducing their ability to respond to emergencies (such as earthquake relief efforts).

To reduce the cost of using process systems for enterprises, providing business process service capabilities within the industry cloud resource center VINCA TARC has become a critical issue. To this end, based on cloud computing technology 0 and the SaaS 1 model, this paper proposes VINCA ProFlow, a multi-tenant SaaS process system that aims to provide on-demand process services (BPM 2 as a Service) in cloud computing environments. Under the SaaS model, operators can centrally control costs related to IT infrastructure, software systems, operations, and maintenance upgrades. By leveraging economies of scale, they can reduce software operation costs, thereby lowering the total cost of ownership for each customer.

2. Related Work

With the rise of cloud computing, SaaS process systems based on cloud environments have begun to emerge. This section first provides a brief overview of the current state of SaaS process systems, then introduces research progress on related key technologies.

2.1 SaaS Process Systems

Cloud computing is an IT service consumption and delivery model built upon abstract, shared, and elastic resources, while the SaaS delivery model represents one of various cloud computing usage patterns. SaaS applications are deployed and operated in the provider' s data center, allowing multiple end users to remotely access the same application instance. A key characteristic of the SaaS delivery model is its tenancy model. Currently, SaaS vendors' tenancy models can be broadly summarized into three types.

Currently, VINCA ProFlow' s SaaS tenancy model adopts the data-isolated, application-instance-shared pattern. Unless explicitly stated by the SaaS vendor, it is difficult to distinguish between fully shared tenancy and data-isolated tenancy from an external perspective.

With the rise of cloud computing, workflow products based on cloud environments have also begun to appear. For example, IBM' s BPM BlueWorks [3], Fujitsu' s Cloud BPM [4], and TIBICO' s Spotfire. IBM BPM BlueWorks provides a series of cloud-based business process management services. With BPM BlueWorks, business leaders, analysts, and professionals can create, share, and collaborate on content through a browser, leveraging pre-built process templates and contributions from process management experts and global users to rapidly complete the entire workflow from strategy formulation to process execution. Fujitsu' s Cloud BPM provides multi-tenant support and can tailor workflow functions according to tenant requirements. TIBICO' s Spotfire, in addition to providing basic process management services, also offers business activity monitoring and critical data analysis capabilities.

Beyond these industrial cloud workflow products, academia has also increasingly focused on research issues related to workflows in cloud computing environments. This includes both research on using cloud infrastructure to improve the performance of existing workflow engines and research exploring how to design workflow architectures for cloud environments. Christina Hoffa et al. from Indiana University explored performance comparisons of the scientific workflow Pegasus-WMS between local clusters and cloud-based virtual clusters [5]. Tim Dörnemann et al. from the University of Marburg developed a BPEL 3 engine plugin [6][7] that can dynamically schedule certain compute-intensive activities in BPEL processes to Amazon EC2 cloud nodes. Vinod Muthusamy et al. from the University of Toronto discussed architectural challenges for cloud workflows and proposed an event-based, SLA (Service Level Agreement)-driven architecture [8] that dynamically decomposes processes into multiple activities with publish/subscribe relationships to adapt to the distributed cloud environment. Evidently, workflows based on cloud computing environments are gradually becoming a focus of attention in the workflow field.

Despite numerous practical and research efforts in both industry and academia, no consensus has been reached on how to define SaaS process systems. This is because process systems themselves encompass multiple components including process discovery, modeling, simulation testing, rules, monitoring, and workflow engines. Although current business process management developers have recognized the benefits of delivering process systems via the SaaS model, they have not reached agreement on which parts of the process system should be delivered as SaaS. The connotation of SaaS business process management differs among vendors. Therefore, this paper does not attempt to provide a definitive definition. However, overall, current foreign vendors' SaaS services primarily focus on providing process and business rule modeling, such as IBM (Lombardi Blueprint and Blueworks) and Software AG Alignspace. A few vendors have the capability to provide process execution environments via SaaS, such as Appian 4, Cordys (Cordys Business Operations Platform), and Vitria. Only a very small number of vendors can simultaneously provide both process modeling and runtime environments via SaaS, such as PegaSystems 5 and Adeptia BPM 6. These vendors also adopt different tenancy models. For example, Adeptia BPM adopts the isolated tenancy model, providing each tenant with one or several virtual machines to deploy their business process management instances, while IBM adopts the shared application instance model.

2.2 Business Data Linkage Between Client and Cloud

As discussed in Section 1, "pure" cloud workflows detached from the client side are unrealistic in practical applications, because the client side often has business systems that meet specific needs. Even general office software maintains important business data. To achieve real-time linkage of business data between the client and cloud, we investigated relevant data synchronization technologies, particularly data synchronization technologies in cloud computing environments.

Data synchronization is a classic problem. As early as 1996, Andrew Tridgell and Paul Mackerras proposed the rsync algorithm [9], which employs incremental synchronization technology and has been widely used in various Unix distributions. With the maturation of distributed file systems and the emergence of cloud storage services, research on utilizing cloud computing environments for data synchronization has begun to appear. James Broberg et al. from the University of Melbourne developed Cloud Storage Mashup [10], which can integrate cloud storage services such as Amazon S3 and Nirvanix to achieve high-performance, low-cost content distribution and synchronization. Huajian Mao from the National University of Defense Technology and Weisong Shi from Wayne State University developed WuKong [11], a cloud-oriented data service that enables real-time synchronization of PCs and other handheld devices via the cloud. Its key feature is providing an abstraction layer that transparently integrates various cloud storage services.

Research on data synchronization technologies has a long history, and the gradual development of cloud storage has provided new possibilities for the evolution of this technology. Current research has emerged on using cloud storage to achieve data synchronization, but it mainly relies on interface integration, lacking algorithmic innovation in this new context.

2.3 Scalability of Workflow Data Management

Cloud-based workflows provide services across the Internet, so the intensity of process requests they handle and the scale of data are far higher than traditional workflow systems. In scenarios with intensive requests and large data volumes, process data read/write operations often become system bottlenecks. With the emergence of distributed data storage systems like Google BigTable [12], researchers have begun to consider using open-source distributed data storage systems such as HBase [13] to enhance the scalability of process management systems.

Jianwu Wang et al. from the University of California studied the integration of the scientific workflow software Kepler with Hadoop, reducing process data processing time and improving the usability of Map-Reduce [14]. Chen Zhang et al. from the University of Waterloo developed a lightweight process management software CloudWF [15] based on Hadoop, using HBase to store process variables and HDFS to store business documents, making the system highly scalable. John Abraham et al. from the University of Texas used HBase to store RDF 7 data output by services and optimized HBase' s storage structure to improve SPARQL query efficiency [16].

However, current research on scalable workflow data management mechanisms has primarily focused on scientific workflows. Scalable data solutions for general-purpose workflow systems are still lacking, while general workflow systems have higher requirements for data consistency and customizable queries.

3. Key Challenges

The usage patterns of rentable SaaS process systems based on cloud computing environments have changed significantly compared to the past: from single-tenant exclusive use to multi-tenant shared use; process systems are no longer deployed within enterprises but in third-party operation centers; the number of users and data volume have become unpredictable rather than predictable; and quality of service guarantees are now constrained by service level agreements rather than being self-determined. These changes present numerous technical challenges for providing available process services in industry cloud centers. These include: how to provide browser-based process modeling tools to support multi-user collaborative modeling; how to ensure that user data cannot be peeked at or stolen by other tenants; how to guarantee business continuity for tenants when some servers fail; how to improve system scalability to meet continuous growth in users and data; how to support integration between users' local business systems and cloud-based process systems; and how to support cross-organizational business collaboration in cloud environments. Some of these issues have been resolved in related work, while others still lack good solutions. Due to space limitations, we only discuss two issues closely related to this paper' s theme, partially illustrating the research motivations behind VINCA ProFlow.

3.1 Data Management in Process Systems

The differences between data management in SaaS process systems and traditional process systems are mainly manifested in three aspects:

1. **Isolated Data Storage**: In traditional process systems, data is stored on the enterprise' s own servers, not exposed on the public Internet, so security concerns are minimal. However, in SaaS-based process systems, to prevent business data leakage, users require reliable guarantees that their process data is isolated from other users' data.

2. **Client-Cloud Data Linkage**: Due to the cross-administrative-domain nature of cloud computing environments, the business data that cloud-based workflows need to process may be distributed across different autonomous domains. There are two schemes for process engines to access client-side business data: one is for the client to encapsulate business data as services for cloud invocation, but this requires high IT infrastructure levels on the client side and easily creates single-point dependencies, even causing Slashdot effects [2]; the alternative is to achieve linkage between client and cloud business data through data synchronization technology, aggregating the business data that users want to share to the cloud in real time for access by process engines and participants. The technical challenge lies in keeping the data in the cloud relatively "fresh."

3. **Dynamic Scalability of Data Management**: Workflow systems provide a stateful service, and management of business process states depends on workflow data management. Frequent state transitions cause intensive read/write operations on workflow data. Traditional workflow systems generally use relational databases to manage workflow data. This centralized data management approach can meet the needs of process management within a single enterprise or organization. However, cloud-based workflow systems must provide services across the entire Internet, and traditional process data management methods can no longer adapt to requirements for high scalability and high-concurrency read/write operations. Although data management software for cloud computing environments such as BigTable [12], Amazon S3, and HBase [13] have emerged, these systems provide relatively simple data query services, primarily key-

value based. Whether they can meet the data management requirements of process systems remains to be further studied.

3.2 Scalable and Reliable Process Operation Support

SaaS systems are characterized by large user bases, continuous user growth, and periodic, "clustered" user access patterns. These features impose high requirements on process application runtime support. The system must be able to provide more processing capacity when user requests are concentrated to avoid significant increases in response time, and reduce processing capacity when user requests decrease to save operational costs. Furthermore, in critical business applications, when business process engines encounter exceptions or crashes during process execution, corresponding exception handling mechanisms should be provided, such as migrating process instances from failed engines to healthy ones for continued execution, ensuring smooth business operations. Therefore, as a sustainably operable process SaaS system, it must consider the continuous growth of system users and data scale during operation and must provide strong guarantees for business continuity. Consequently, a scalable and reliable process runtime environment is required.

4.1 Objectives

Based on the above analysis, the design philosophy of VINCA ProFlow is to address the current situation of backward IT infrastructure in small and medium-sized organizations by providing them with an on-demand rentable process service (BPM-as-a-Service). The specific objectives are:

1. **Support Data-Isolated Tenancy Model**: While tenants share the same process service instance, each can enjoy exclusive access to an independent data space, ensuring effective data isolation and privacy protection among tenants.

2. **Scalable Data Management**: When VINCA ProFlow' s existing hardware resources are insufficient to store process data or support high-concurrency process data access, new servers can be added to share the storage and data access load. This process should be transparent and smooth to the process engine, requiring no code modifications or stopping of running process instances.

3. **Support Data Integration Between Client Business Systems and Cloud Applications**: To enable business processes hosted in the cloud for small and medium-sized organizations to access the "freshest" business data from the organizations' local systems, any minor changes in local business data must be synchronized to the cloud business data space in real time for use during business process execution.

4. **Provide Reliable and Scalable Process Runtime Environment**: Reliability means that when an engine server fails, the system can mi-

grate process instances from the failed server to healthy servers within a short time to ensure business continuity. Scalability means that when user concurrent access is intensive, engine servers can be dynamically added to horizontally extend system processing capacity. When user access decreases, engine servers can be gradually reduced to lower operational costs.

4.2 Key Technologies

[FIGURE:1]

To achieve these objectives, this paper builds upon previous work in two areas: First, we introduced a multi-tenant shared-instance, data-space-exclusive SaaS tenancy model into the VINCA ProFlow architecture, achieving isolated storage of organizational data, process model data, process runtime data, and personalized configuration data among tenants, preventing tenant data from being peeked at or stolen. Second, by introducing mechanisms for virtual machine management, engine cluster management and scheduling, and data management, we provide a dynamically scalable and reliable business process execution environment capable of handling continuous and rapid changes in user requests and data, as shown in Figure 1. Due to space limitations, this paper only reports on the most relevant key technologies.

1. **Multi-Tenant, Scalable Data Management Technology**

To achieve the first three objectives mentioned in Section 4.1, we focused on developing dynamic scalability technology for multi-tenant data management during VINCA ProFlow' s development. Progress has been made in three aspects:

- **Multi-Tenant Data Isolation Based on Virtual Data Services**: In a multi-tenant architecture, ensuring data isolation presents two difficulties. On one hand, there is access permission isolation—tenants cannot access each other' s data without authorization to prevent data privacy leakage. On the other hand, there is quality-of-service isolation for data access—when one tenant submits a slow query, other tenants' data access requests should not be delayed or denied because the database server is too busy. Virtualization technology is an effective method for resource isolation. VINCA ProFlow uses Xen 9 to encapsulate operating systems with relational databases and key-value data storage systems into virtual machine images. On each virtual machine instance, multiple data service processes run on different ports, with each tenant exclusively occupying its own data service process. When creating a tenant, VINCA ProFlow can automatically create the virtual data services needed for its applications to store process data, organizational information, and personalized configuration data, achieving data isolation among tenants. Additionally, graphical monitoring dashboards allow viewing each tenant' s quota usage in the system.

- **Client-Cloud Business Data Linkage Based on Data Sharding**: When integrating client business systems with cloud systems, the most critical aspect is achieving linkage between local and cloud data. In practical applications, business data changes often exhibit locality characteristics—updated data on the client side differs only partially from its cloud replica. To this end, we propose a data sharding-based client-cloud business data linkage mechanism aimed at quickly identifying data differences and reducing data traffic to improve the real-time nature of business data linkage. The approach deploys agent programs on the client side that implement event-driven data change monitoring mechanisms by registering file system hooks and monitoring database logs. The agent divides business data into fragments of appropriate granularity based on data type and calculates a characteristic value for each fragment. For plain text document data, the unit is a certain number of lines; for binary document data, the unit is a fixed-length byte window; for data in relational database management systems, the unit is a fixed-size tuple window. Data characteristic values are calculated using a combination of weak and strong checksum algorithms. When client-side data is modified, the client agent transmits the characteristic values of business data fragments to the cloud. The cloud identifies missing fragments by comparison and returns their characteristic values to the client agent. The client agent then compresses and sends the missing data fragments to the cloud, where they are decompressed and merged into the appropriate locations.
- **Feature-Based Scalable Management of Process Data**: Traditional business process systems generally use relational databases to manage process data, but this approach has scalability limitations. Classic scalability solutions for relational databases mainly include two types: read-write separation [17] and data sharding [18]. Read-write separation suits read-heavy applications, but data latency becomes severe in environments with frequent updates. While data sharding improves scalability, it sacrifices transaction guarantees and complex query capabilities of relational databases. In recent years, distributed key-value storage systems emerging in Internet computing (such as BigTable [12], Dynamo, Cassandra [19], etc.) have made special optimizations for key-based read/write performance and can partition and replicate data across multiple servers through peer-to-peer (P2P) protocols, enhancing horizontal scalability. We consider combining the advantages of different data management tools for process data management. Through preliminary experiments, we observed that process data exhibits different access characteristics, as shown in Table 2 . Metadata has more read operations than write operations, with relatively complex queries—for example, retrieving process definitions or querying users under an organization. Instance data has balanced read/write operations that are both frequent, with simple queries primarily based on single-key lookups—for example, changing the status of a process instance based on its ID or updating a process variable' s value based on its ID.

Log data has more write operations than read operations, but queries are relatively complex and often statistical—for example, finding the 10 most time-consuming activities within a certain period.

Based on the analysis and summary of these characteristics, VINCA ProFlow proposes a feature-based scalable management mechanism for process data. The idea is to adopt different storage methods for different types of process data according to their characteristics, integrating relational databases and distributed key-value storage systems to solve scalability issues within workflow data. Relational databases are used for metadata storage, while the distributed key-value storage system Cassandra is used for storing runtime process instance data, leveraging the scalability of distributed key-value storage to enhance workflow runtime scalability. Additionally, an abstract Process Data Access Layer (PDAL) shields the workflow engine from underlying data model differences and internally completes process data adaptation and routing, allowing the engine' s data access code to remain unchanged as the number of nodes scales.

2. **Engine Cluster Management and Scheduling Technology**

To achieve the scalability proposed in Objective 4 of Section 4.1, VINCA ProFlow adopts a distributed multi-engine architecture. Underlying data consistency is provided by the process data access layer. The meta-engine is responsible for scheduling and managing multiple process execution engines. The number of process execution engines initially ranges from 1-2. When system load is high, administrators can add relatively idle process execution engines to the cluster through the system management console to share the system load. When no sufficient process execution engines are available, administrators can activate instances of virtual machine images with process execution engines installed to achieve rapid scaling. The engine cluster scheduling principle is: when the meta-engine receives a request from the front-end workflow portal, it selects a process execution engine to execute the requested process through the first-level scheduler' s decision based on the application. The meta-engine records the response time of each process request and periodically adjusts the weight strategy for each process. Furthermore, since processes contain not only local service calls but also remote service calls, and sometimes a remote service call activity can be provided by multiple nodes, each process execution engine has an independent second-level scheduler for scheduling service call activities during application execution.

- **Weight Presetting Based on Process Similarity**: To reduce the time required for weight convergence of weight strategies corresponding to newly deployed processes, we propose a weight presetting algorithm based on process similarity. If no processes are currently deployed in the system, the initial weight distribution for the newly deployed process is set equally among all engines. If the system has deployed a certain number of processes and their weight strategies have converged to a stable state after some runtime, the initial weight strategy for the newly deployed process can be calculated based on the weight values of similar processes.

- **Dynamic Weighted Round-Robin Based on Process Characteristics**: Common scheduling algorithms include simple round-robin, static weighted round-robin, request-number-based weighted least-connections-first, and dynamic weighted round-robin. Combining VINCA ProFlow' s business characteristics, we propose a dynamic weighted round-robin method based on process characteristics. Traditional dynamic weighted round-robin uses a unified weight strategy for all requests, whereas VINCA ProFlow' s algorithm uses different weight strategies for different business processes. Resource consumption varies across different process executions –some processes are compute-intensive, others are I/O-intensive, and the computing power, I/O throughput, and operating systems of nodes hosting process execution engines differ. Therefore, appropriate weight strategies need to be maintained for each process type.
- **Two-Level Scheduling Mechanism**: Since services invoked by processes include both local services and services on other nodes, and multiple nodes may have replicas of the same functional service, the system must select which node should execute the service when encountered during process execution. Due to differences in hardware/software resources and communication speeds across nodes, service call response times vary. Therefore, to optimize service call response times, VINCA ProFlow designed a two-level scheduling mechanism: in addition to the meta-engine' s first-level scheduler for applications, each engine maintains a second-level scheduler for service calls. This scheduler distributes requests for remote service call activities to nodes deploying service replicas according to certain weights.

3. **Engine Failure Detection and Process Instance Migration Technology**

To achieve a reliable process execution environment, several important issues must be addressed: accurately and promptly detecting failed engines, and rapidly migrating and recovering unfinished processes from failed engines; sharing process execution information among engines so that when one engine fails, other engines can obtain information about unfinished processes; and selecting appropriate process recovery starting points to avoid re-executing completed parts.

To address these issues, we researched process engine failure detection and instance migration technology, and performed three tasks based on engine clusters and meta-engine management:

- **Extended the process activity runtime model in the WMFC (Workflow Management Coalition) specification to describe process activity execution at a finer granularity**;
- **Implemented process information sharing in the engine cluster using distributed caching methods. When an engine fails, other engines can read information about unfinished processes from**

the cache server and recover them;

- **Used multicast technology to detect engine failures. When the meta-engine detects a failed engine, it promptly migrates unfinished process instances from the failed engine to healthy engines for continued execution**.

5.1 Scalability Evaluation

To verify VINCA ProFlow' s scalability, we used tools such as JMeter and Apache Benchmark to simulate high-concurrency process request scenarios and evaluated the system' s throughput (number of process requests processed per unit time) under different deployment modes.

The test case used in this paper consisted of 100,000 process requests with a concurrency level of 1,000. The tested process was a human-free process including sequential, branching, and looping structures, with script activities in the process to avoid performance interference from external systems introduced by Web Service calls. The system deployment modes were: (1) one engine paired with one data node; (2) keeping one data node unchanged while dynamically increasing the number of process engines; (3) dynamically increasing both process engines and data nodes simultaneously.

Test results show that in deployment mode (1), due to the physical performance limitations of a single process engine and single data node, throughput becomes difficult to improve after reaching a certain level. In deployment mode (2), system throughput can be further improved by increasing the number of process engines, but is ultimately still limited by the data node' s processing capacity. In deployment mode (3), throughput capacity can be continuously extended by simultaneously increasing both engine nodes and data nodes.

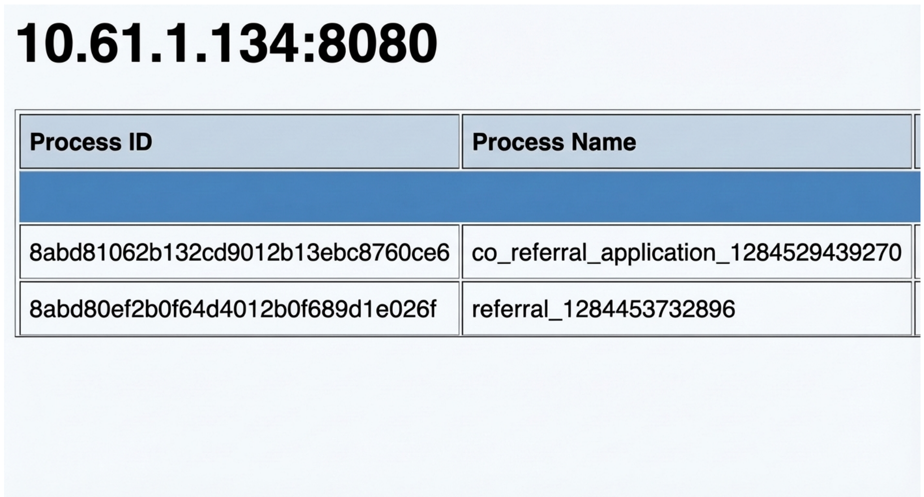
5.2 Reliability Evaluation

To test VINCA ProFlow' s reliability, we first started two engines, then shut down one engine to observe whether process instances running on the shut-down engine were successfully migrated to the other engine and recovered for execution. As shown in Figure 2

, before testing, the system had two healthy engines: the engine with IP 10.61.1.134 had 5 executing processes, and the engine with IP 10.61.0.111 had 1 executing process.

During testing, we forcibly shut down the engine with IP 10.61.0.111. As shown in Figure 3 [FIGURE:3], the 1 process from the shut-down engine was migrated to the engine with IP 10.61.1.134 for continued execution.

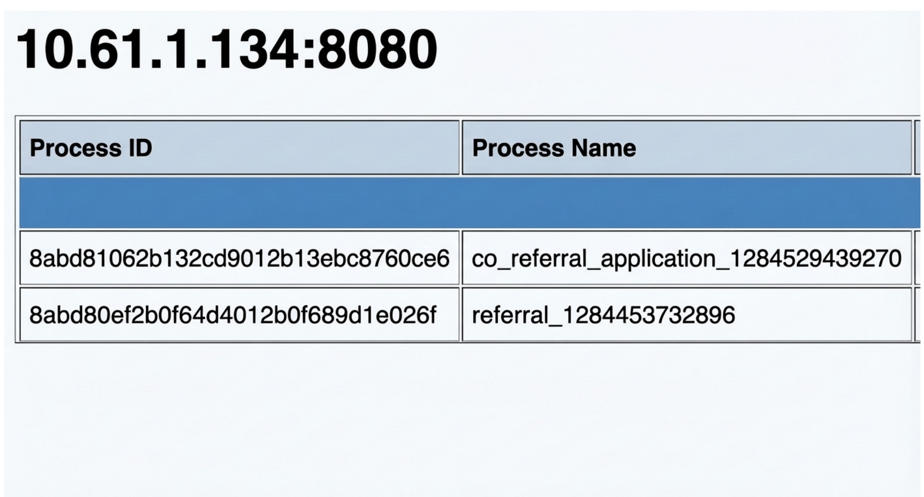
[FIGURE:3]



10.61.1.134:8080

Process ID	Process Name
8abd81062b132cd9012b13ebc8760ce6	co_referral_application_1284529439270
8abd80ef2b0f64d4012b0f689d1e026f	referral_1284453732896

Figure 1: Figure 2



10.61.1.134:8080

Process ID	Process Name
8abd81062b132cd9012b13ebc8760ce6	co_referral_application_1284529439270
8abd80ef2b0f64d4012b0f689d1e026f	referral_1284453732896

Figure 2: Figure 2

6. Conclusion

As a component of VINCA TARC, VINCA ProFlow aims to provide a rentable SaaS business process system. This paper first analyzes the necessity of providing process service capabilities in VINCA TARC based on the contradiction between SMEs' demand for business process management and their investment capacity. It then discusses domestic and international related work progress and analyzes key issues that need to be addressed when delivering business process systems in SaaS mode. Finally, it discusses the objectives, key technologies, and experimental evaluation of VINCA ProFlow, a SaaS process system independently developed by the Institute of Computing Technology, Chinese Academy of Sciences.

References

[1] Brian Hayes. Cloud Computing. Communications of the ACM, 2008, 51(7):9-11.
[2] Slashdot Effect. http://en.wikipedia.org/wiki/Slashdot_effect.
[3] IBM BPM BlueWorks. https://apps.lotuslive.com/bpmblueworks/.
[4] Fujitsu Cloud BPM: Fujitsu Global. http://www.fujitsu.com/global/services/software/interstage/solutions/bpm.html
[5] Christina Hoffa, Gaurang Mehta, Timothy Freeman. On the Use of Cloud Computing for Scientific Workflows. In IEEE Fourth International Conference on eScience (eScience '08), Indiana, USA, Dec. 7-12, 2008, pp.640-645.
[6] Tim Dörnemann, Ernst Juhnke, Bernd Freisleben. On-Demand Resource Provisioning for BPEL Workflows Using Amazon' s Elastic Compute Cloud. In Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid ' 09), Beijing, China, May 18-21, 2009, pp.140-147.
[7] Tim Dörnemann, Ernst Juhnke, Thomas Noll. Data Flow Driven Scheduling of BPEL Workflows Using Cloud Resources. In IEEE 3rd International Conference on Cloud Computing (CloudCom ' 10), Miami, USA, July 5-10, 2010, pp.196-203.
[8] Vinod Muthusamy, Hans-Arno Jacobsen. BPM in cloud architectures: Business process management with SLAs and events. In Business Process Management 2010 (BPM ' 10), NJ, USA.
[9] Tridgell, A. and Mackerras, P. The rsync algorithm. Technical Report TR-CS-96-05, Department of Computer Science, The Australian National University, Canberra, Australia, 1996.
[10] James Broberg, Rajkumar Buyya, Zahir Tari. Creating a ' Cloud Storage' Mashup for High Performance, Low Cost Content Delivery. In 6th International Conference on Service Oriented Computing (ICSOC ' 08), Sydney, Australia, Dec. 1-5, 2008, pp.178-183.
[11] Huajian Mao, Nong Xiao, Weisong Shi, Yutong Lu. Wukong: A Cloud-Oriented File Service for Mobile Internet Devices. In Proceedings of the 7th International Conference on Services Computing (SCC ' 10), Miami, USA, July

5-10, 2010.
[12] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh. Bigtable: A Distributed Storage System for Structured Data. ACM Transactions on Computer Systems (TOCS), 2008, 26(2):1-26.
[13] HBase. http://hbase.apache.org/.
[14] Jianwu Wang, Daniel Crawl, Ilkay Altintas. Kepler + Hadoop: A General Architecture Facilitating Data-Intensive Applications in Scientific Workflow Systems. In Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science (WORKS ' 09), Portland, Oregon, USA, November 15, 2009, pp.1-7.
[15] Chen Zhang, Hans De Sterck. CloudWF: A Computational Workflow System for Clouds Based on Hadoop. In CloudCom 2009, LNCS 5931, M.G. Jaatun, G. Zhao, and C. Rong (eds.), Springer-Verlag, 2009, pp.393-404.
[16] John Abraham, Pearl Brazier, Artem Chebotko, etc. Distributed Storage and Querying Techniques for a Semantic Web of Scientific Workflow Provenance. In Proceedings of the 7th International Conference on Services Computing (SCC ' 10), Miami, USA, July 5-10, 2010.
[17] Emmanuel Cecchet, George Candea, Anastasia Ailamaki. Middleware-based Database Replication: The Gaps Between Theory and Practice. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD' 08), Vancouver, Canada, June 9-12, 2008, pp.739-750.
[18] Pritchett D. BASE: An Acid Alternative [J]. ACM Queue. 2008, 6(3): 48-55.
[19] Avinash Lakshman, Prashant Malik. Cassandra: a decentralized structured storage system [J]. ACM SIGOPS Operating Systems Review (SIGOPS Oper. Syst. Rev.), 2010, 44(2):35-40.

Authors:

Li Houfu: Assistant Researcher, Software Integration and Service Computing Research Division, Institute of Computing Technology, Chinese Academy of Sciences
Sun Junyi: Ph.D. Candidate, Software Integration and Service Computing Research Division, Institute of Computing Technology, Chinese Academy of Sciences
Ma Qiang: Master' s Student, Software Integration and Service Computing Research Division, Institute of Computing Technology, Chinese Academy of Sciences
Wang Lei: Master' s Student, Software Integration and Service Computing Research Division, Institute of Computing Technology, Chinese Academy of Sciences
Zhao Zhuofeng: Senior Engineer, Software Integration and Service Computing Research Division, Institute of Computing Technology, Chinese Academy of Sciences
zhaozf@software.ict.ac.cn

Figures

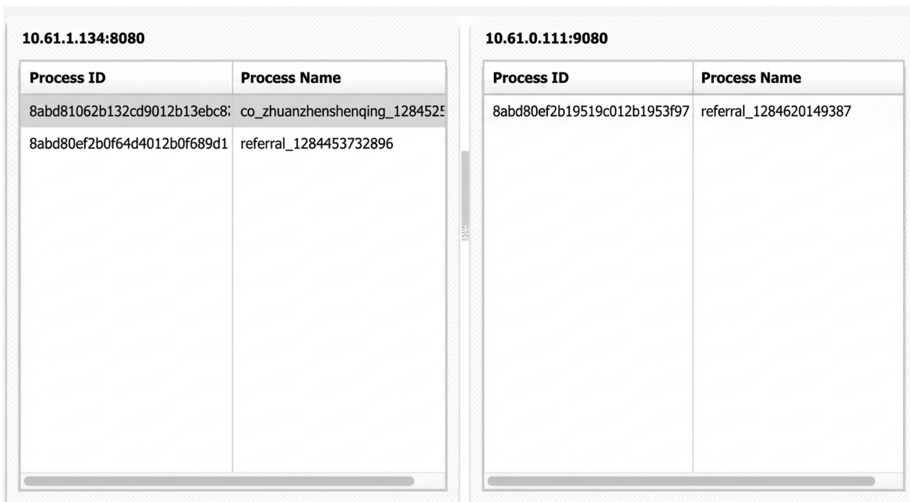


Figure 3: Figure 4

Source: ChinaXiv —Machine translation. Verify with original.