INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 9(4):439-452, August, 2014.

Performance Evaluation of ENUM Name Servers

S.H. da Mata, J.M.H. Magalhaes, A. Cardoso, H.A. Carvalho, P.R. Guardieiro

Saulo Henrique da Mata*, Johann M. H. Magalhaes,
Alexandre Cardoso, Paulo R. Guardieiro
Federal University of Uberlandia
Av. Joao Naves de Avila, 2121, Bloc 3N
Uberlandia, MG, Brazil - 38400-912
johann@iftm.edu.br, alexandre@ufu.br, prguardieiro@ufu.br
*Corresponding author: saulodamata@doutorado.ufu.br

Helio A. Carvalho
CPqD, Rua Dr. Ricardo Benetton Martins, s/n
Parque II do Polo de Alta Tecnologia, Campinas, SP, Brazil - 13086-902
helioac@cpqd.com.br

Abstract: ENUM is a protocol designed for mapping E.164 telephone numbers into Internet URIs. This protocol imposes new requirements and challenges on traditional DNS servers. In this sense, this paper presents a performance evaluation of ENUM name servers considering these new scenarios. We selected four name servers implementations (BIND, MyDNS-NG, NSD and PowerDNS) to compare their performance, using a benchmarking testbed. To perform the evaluation, we have defined three performance metrics: query throughput, response time, and CPU usage. These metrics required the development of a new procedure for the evaluation, since DNSPerf presented limitations in the management of CPU resources. From results, we classified MyDNS-NG as not suitable for ENUM purposes, due to its quite low query throughput and poor scalability. PowerDNS server performed better than MyDNS-NG, but presented high sensibility to database size. On the other hand, BIND and NSD servers achieved a high query throughput with great scalability features. Since we have used extended scenarios and size of database that were not evaluated before, these results bring to light an interesting insight about ENUM servers.
Keywords: ENUM, Benchmarking, BIND, NSD, MyDNS-NG, PowerDNS

1 Introduction

Nowadays, we are witnessing a continuous evolution of communication networks. In this evolution process, services previously exclusive to one communication network began to be shared with other networks. In this context, Voice over IP (VoIP) is one of the best examples of this migration, where the voice service from Public Switched Telephone Network (PSTN) is added to data service from Internet and both services are available with a lower cost than the currently one offered by traditional networks.

Despite the expanding number of VoIP users and the availability of several providers, geographically, users of this technology are located on a set of "isolated islands", where the interconnection among these islands may occurs by means of PSTN. This connection is performed by Media Gateways (MGWs) and E1 links, which are commonly used to interconnect the VoIP carriers with the PSTN. This interconnection infrastructure requires investments by operators in equipment and the payment of fees for the use of communication links. These expenses increase the total operating costs, burdening the end user.

Furthermore, one should consider that in spite of the advantages of VoIP application over the PSTN, it is not possible the immediate replacement of this technology, because of the complexity involved. This implies a transition period when both technologies will coexist, since the migration of PSTN subscribers to VoIP can occur slowly and gradually. In this scenario, we have four basic types of calls: PSTN-PSTN, PSTN-IP, IP-PSTN and IP-IP. So, the problem is not restricted only to avoid the use of PSTN for IP-IP calls, but also to allow calls between terminals of two different networks (PSTN-IP, IP-PSTN).

Traditionally, E.164 telephone numbers [1] are used in PSTN and Uniform Resource Identifiers (URIs) [2] are used on the Internet. Signaling protocols such as SIP and H.323 can work with these two types of identifiers. For Internet users, who usually have the availability of a alphanumeric keyboard, the textual names (URIs) are mostly used because of easy memorization and deduction. However, a problem arises when VoIP and PSTN networks are interconnected, since PSTN terminals typically provide a numeric keyboard to users. Thus, it is necessary a mapping system between E.164 telephone numbers and URIs which permits the interconnection of PSTN and VoIP users. Moreover, this mapping system must allow IP-IP calls without the connection with PSTN. In this context, the use of Electronic Number Mapping (ENUM) protocol stands out as a plausible solution.

ENUM [3] is a technology that maps E.164 telephone numbers used in PSTN into URIs used on the Internet. Although ENUM is based on existing DNS infrastructure, it has certain features, for example, a database with a huge number of records and the requirement of short resolution time to reach a PSTN similar performance. In addition, it causes an increase in network traffic, since the size of the response message to ENUM queries is greater than in classic DNS. Therefore, it is of fundamental importance a study of how ENUM servers perform on this new environment.

In this context, this paper presents a new performance evaluation of ENUM name servers. Using a high performance hardware, we selected four different name servers implementations (BIND, MyDNS-NG, NSD and PowerDNS) for the evaluation. Our goal was to make a detailed study of the performance of each one of these servers and verify whether this performance can meet the ENUM protocol requirements.

The rest of the paper is organized as it follows: Section 2 describes related work. In Section 3 we present our testbed and methodology. Section 4 shows the benchmarking results. Finally, concluding remarks are offered in Section 5.

2 Related work

The performance study of ENUM servers has been reported in some works. In [4] and [5] we find a benchmarking study of three ENUM servers (BIND, Navitas and PowerDNS). According to authors, Navitas and PowerDNS met the requirements of ENUM protocol, considering performance metrics as query throughput, response time and scalability. On the other hand, BIND was not qualified as a serious ENUM server.

In [6], we find another benchmarking study of two ENUM servers (BIND and NSD). In this work, BIND and NSD presented a similar performance for query throughput that is quite different from the results presented in [5].

In [7], there is an evaluation study of number portability and record updating process in ENUM systems. Using DNS zone transfer method for database synchronization, this work evaluated two ENUM servers (BIND and PowerDNS). The considered metric was the information transference speed between different zones. The results showed that BIND was 3 to 4 times faster than PowerDNS to deliver the number portability data.

In [8] there is a performance evaluation of ENUM protocol considering different approaches for ENUM architecture. In this work, the focus was not the ENUM server performance, but the hierarchy arrangement of these servers.

In [9], there is not a comparative study between ENUM servers, but BIND was evaluated in the context of French ENUM architecture. The goal of this work was to build a simulation tool based on the data of a real ENUM system. A similar work can be found in [10].

In all these works, we found different types of hardware and servers. Besides, there are differences in the methodology and, as a consequence, it is not easy to compare the results presented by each work. However, among these works, [5] and [6] present a more complete study, a concise methodology, and good hardware and software profiles. Although these works present similar objectives, they report divergent results. Moreover, they were developed about 4 to 5 years ago. Since then, we have been observing a great evolution for hardware, software and database technologies, creating the demand for updated and reliable studies about the performance of ENUM servers.

In [11], one can find a preliminary evaluation of ENUM name servers. In the present paper, we extend this work by adding new scenarios of evaluation not contemplated in any previous works, for example, performance evaluation considering multiples Naming Authority Pointer (NAPTR) records per Fully Qualified Domain Name (FQDN).

In this sense, our study fulfills and updates all these former works. We built a powerful testbed that outperforms the hardware and software used in [5] and [6]. We improved the methodology presented in [5] by extending the range of the parameters of evaluation. Besides, we report limitations of the DNSPerf (version 1.0.0.1) and present results for new scenarios or evaluation.

3 Testbed and methodology description

3.1 Performance Metrics

Three performance metrics were defined for the tests:

- Query throughput: the number of queries which the ENUM server can respond successfully per unit of time, i.e., queries per second (qps).
- Response time: time between sending of query to server and the moment the client receives the answer to that query.
- CPU and RAM usage: this metric evaluates how well the ENUM server uses these hardware resources.

3.2 ENUM servers

Nowadays, there is a great number of existing ENUM servers. We selected a set of servers in a way that we could evaluate different approaches for implementation of a name server. We also considered other factors such as the relevance of the server, availability, documentation and updating process. In this sense, we selected BIND (version 9.8.1) and NSD (version 3.2.10) to represent the name servers based on zone files. MyDNS-NG (version 1.2.8.31) and PowerDNS (version 2.29) were selected to represent ENUM servers that allow the use of other types of database. For these servers we used MySQL (version 5.1) as the database.

3.3 EnumBenchTool

EnumBenchTool is a test management tool for benchmarking of ENUM servers developed by the Network Computer Lab (NCL) of the Federal University of Uberlandia. This tool has been developed to automate, standardize and validate the tests and to facilitate the achievement of results.

At the current stage of development, EnumBenchTool admits the servers BIND, MyDNS-NG, NSD and PowerDNS. This tool is not responsible for the benchmarking test itself, but it packs several other existing tools, simplifies the benchmarking management and makes configuration, synchronization and validation processes transparent to user. Among these packed tools, we highlight DNSPerf [12], which is a software developed by Nominum, widely used to analyze the performance of authoritative DNS servers. EnumBenchTool is responsible for triggering each DNSPerf test step, setting the number of emulated clients, the runtime, the query file and other parameters. In addition, EnumBenchTool is responsible for processing the results from DNSPerf.

3.4 Number of DNSPerf processes

Unfortunately, DNSPerf (version 1.0.0.1) is not a multi-core software, i.e., it uses only one core of the processor for each operating system (OS) process. Depending on the performance of the ENUM server, DNSPerf may not be able to send a sufficient amount of queries to saturate the name server using one single OS process. Thus, the saturation occurs on the client, invalidating the obtained results. So, it is necessary to increase the number of DNSPerf OS processes emulating clients to avoid this saturation. However, this increase cannot be indiscriminate, since the operating system has limitations to manage a very large number of processes. Initial tests showed that the use of five processes is a reasonable choice for this type of evaluation, as showed in Table 1.

Table 1: Impact of the number of DNSPerf processes in server throughput.

Processes	Clients/Process	Max. Throughput (qps)
1	500	202743
5	100	309927
10	50	305523
20	25	95189
500	1	60559

3.5 Number of clients

In [5], there is an evaluation interval from 2 to 60 clients. However, initial tests proved that this interval is too short to evaluate the server satisfactory, since 60 clients were not enough to saturate the ENUM server and, then, the results would not be plausible. Therefore, we established that the number of emulated clients, who make simultaneously queries to server, would be in a range of 2 to 1000 clients. This range proved to be sufficient for stability of the system.

3.6 Type of records

We evaluated the ENUM servers considering two types of records:

- existing records: the name server has a DNS NAPTR record for the FQDN queried;
- non-existing records: the name server has not a DNS NAPTR record for the FQDN queried.

3.7 Database record sets

To check whether the server meets the requirement for scalability, three database record sets with 500 thousand (500k), 5 million (5M) and 50 million (50M) of records were defined.

3.8 Query files

Using the EnumBenchTool, we created query files for each database record set and for each record type as well. For each query file, these FQDNs were generated in a random manner to avoid cache memory interferences. In this sense, we also generated independent random query files for each DNSPerf processes.

3.9 DNS zones

Each record set was divided into ten zones, with each zone consisting of one tenth of the total number of records. This division method, reported in [5], is a practical and efficient generation process for different types of records. Furthermore, using BIND, we evaluated the impact of the number of DNS zones in the server query throughput, as showed in Table 2.

Table 2: Impact of the number of DNS zones in server throughput.

Number of DNS Zones	Max. Throughput (qps)
5	280994
10	324550
50	271256
100	270402
1000	265486

Each zone file starts with a $TTL directive. We set the TTL as 86400 seconds. As we have a huge database and the records are queried using random and independent query files, the cache mechanism is not a great factor in our experiment. An example of the beginning of a zone file is as follow:

```
\$TTL 86400
\$ORIGIN 0.3.3.4.3.5.5.e164.arpa.
@ IN SOA ns.0.3.3.4.3.5.5.e164.arpa. root.e164.arpa. (
                                                     2011080901
                                                     21600
                                                     3600
                                                     604800
                                                     3600
                                                    )
 IN NS ns.0.3.3.4.3.5.5.e164.arpa.
ns IN A 10.0.0.1
```

3.10 Testbed layout

The testbed consists of two Linux machines, running Ubuntu 11.10 and Ubuntu Server 11.10. Each machine has two processors Intel Xeon 6-Core E5645 HT 2.40 GHz, 16 GB of RAM and two hard drives of 500 GB. Both machines were connected via 1000 Mbps Ethernet connections. The testbed layout is based on [5] and is illustrated in Figure 1.

The master entity is responsible for synchronizing the activities of the machines under test. It sends commands to other two machines for the starting of each step of the test as well as the resource monitoring. Control messages were sent through a link of 100 Mbps. Queries and responses were sent through an exclusive link of 1000 Mbps.

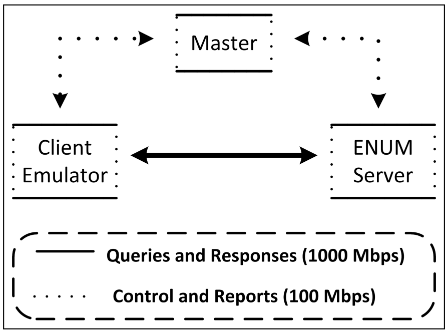


Figure 1: Testbed layout.

4 Performance comparison of the ENUM servers

The tests carried out at NCL were constituted of multiples steps. For each of these steps, the number of emulated clients was gradually incremented, according to commands sent by the master entity. Each step had a duration of 60 seconds, in which DNSPerf sent queries to ENUM server. At the end of the step, there was a timeout of 10 seconds before the starting of the next step to assure that the queries from the earlier step would not interfere in the results of the current step.

4.1 Overall performance

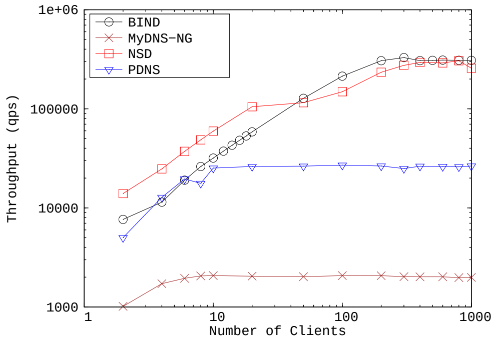
Figure 2 shows ENUM servers performance for query throughput considering the record sets and record types previously defined. In Figure 2(a), it is possible to observe that BIND and NSD presented a great performance for throughput, reaching a maximum throughput of about 300 kqps. In this same figure, we can observe that PowerDNS and MyDNS-NG presented a maximum throughput much lower than the servers based on zone files. This occurred because servers that use zone files load all the records to physical memory (RAM), which is known by its access speed. In our tests, MySQL was not able to follow this speed.

Figures 2(b) and 2(c) show the performance of ENUM servers for throughput with an increased database. BIND kept its performance even with a huge increase of the database. NSD also presented a great performance with 5M of records in the database, but the compiler of NSD zone files (*zonec*) failed to build all zones for the 50M record set. This failure of *zonec* was also reported in [6]. PowerDNS and MyDNS-NG were sensitive to database increasing and the throughput decreased as we increased the database size.

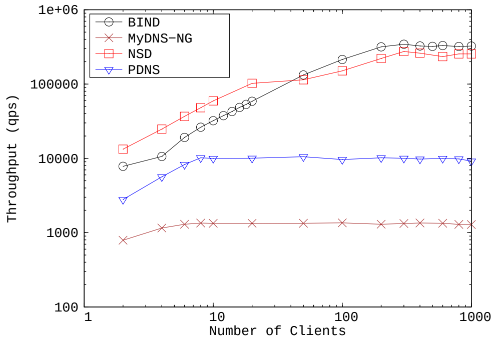
Another important parameter to be analysed is the server response time. To offer a service with quality similar as PSTN, VoIP call establishment time should not be longer than the PSTN call signalling interval. ENUM lookup response time is part of the total signalling interval for VoIP call setup. According to [13], the PSTN signalling interval has mean of 205-218 ms (337-350 ms in the 95th percentile). Thus, the lookup response time on the ENUM server should be lower than this mean value, since successive queries may occur in a single call setup.

Figures 3(a), 3(b) and 3(c) show the average response time for the evaluated ENUM servers. We note that all evaluated servers were able to satisfy the requirement of lookup response time in a VoIP call. This figure is also in consonance with the throughput results earlier commented. Servers that use zone files are faster than those that use MySQL database and, as consequence, the former servers presented a higher value for query throughput.

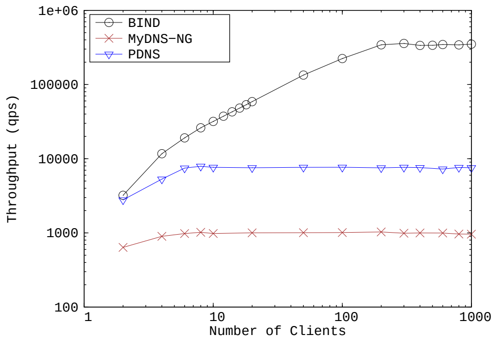
Figures 2(d) and 3(d) show a similar scenario, but with queries to non-existing records. In this scenario, BIND and NSD kept their performance. On the other hand, PowerDNS and



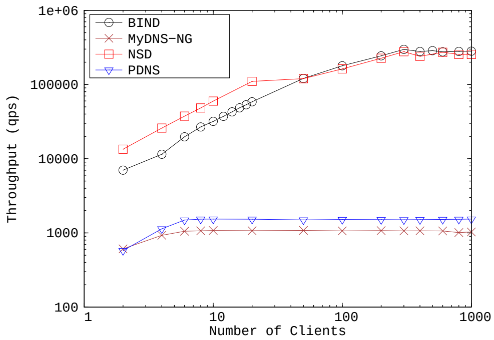
(a) 500k/existing records



(b) 5M/existing records



(c) 50M/existing records



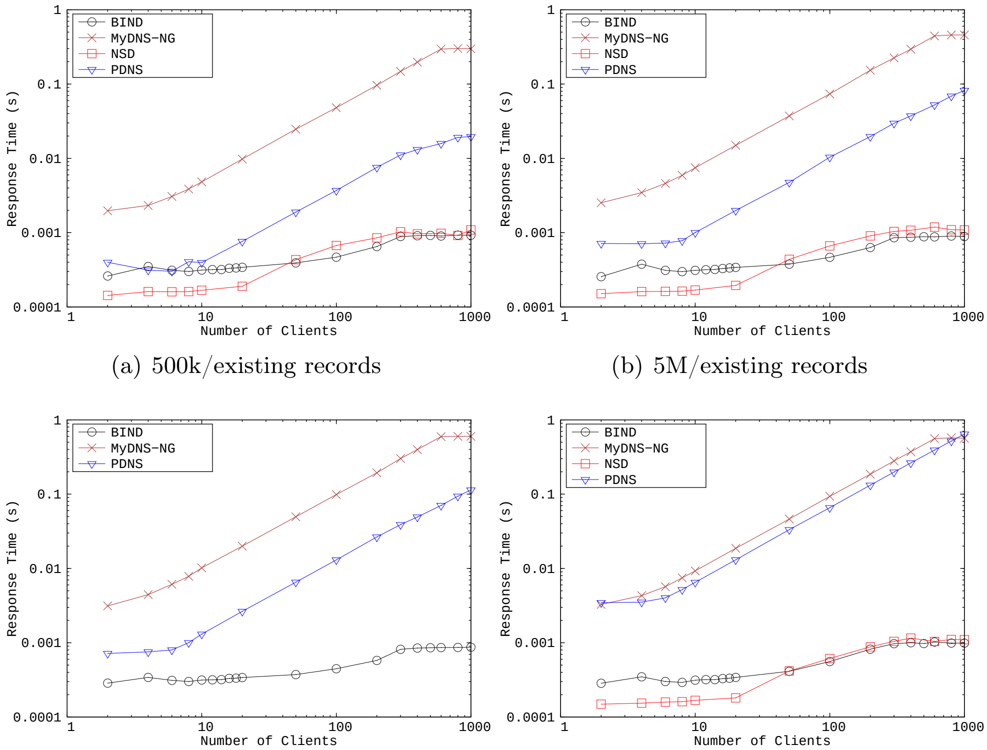
(d) 5M/non-existing records

Figure 2: Query Throughput.

MyDNS-NG presented a reasonable decrease in their performance when we considered queries to non-existing records.

Figures 2 and 3 indicate that BIND and NSD reached saturation when 300 clients were sending queries to ENUM server. Under the saturation condition, the throughput of the server stops to increase and enters an unstable zone. Similarly, the response time stops increasing under this condition. In this situation, these servers start to experience increasing losses, as showed in Figure 4. However, even under saturation condition these servers answers the queries with a response time near those found for maximum throughput and, therefore, the response time under saturation condition shows only a small variation. On the other hand, MyDNS-NG reached saturation before 10 active clients in the system. This throughput low performance is reflected in response time that continues to increase until the system starts to experience losses. PowerDNS also reached saturation before 10 active clients in the system, but it presented a different behavior for response time and loss rate. The response time continued growing up even when losses began to be perceived.

Figure 5 indicates servers CPU usage. We can note that none of the evaluated servers have overpassed the maximum CPU capacity of the server, confirming that server saturation was caused by the software and not by hardware restrictions. In Figures 5(a) to 5(d), we can also note that to reach a high query throughput, BIND uses up to 90% of server CPU resources. Surprisingly, NSD reached a similar throughput with approximately 15% of CPU resources. The results for MyDNS-NG and PowerDNS indicate that these servers did not take advantage of the capacity of multi-core processors. We believe that this is the main factor for the differences in performance between the ENUM servers. While BIND and NSD used all the cores of the processors, PowerDNS and MyDNS-NG used only one of the cores. The implementation details



(a) 500k/existing records (b) 5M/existing records

(c) 50M/existing records (d) 5M/non-existing records

Figure 3: Response Time.

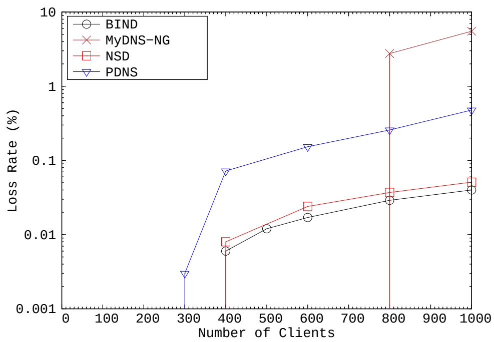


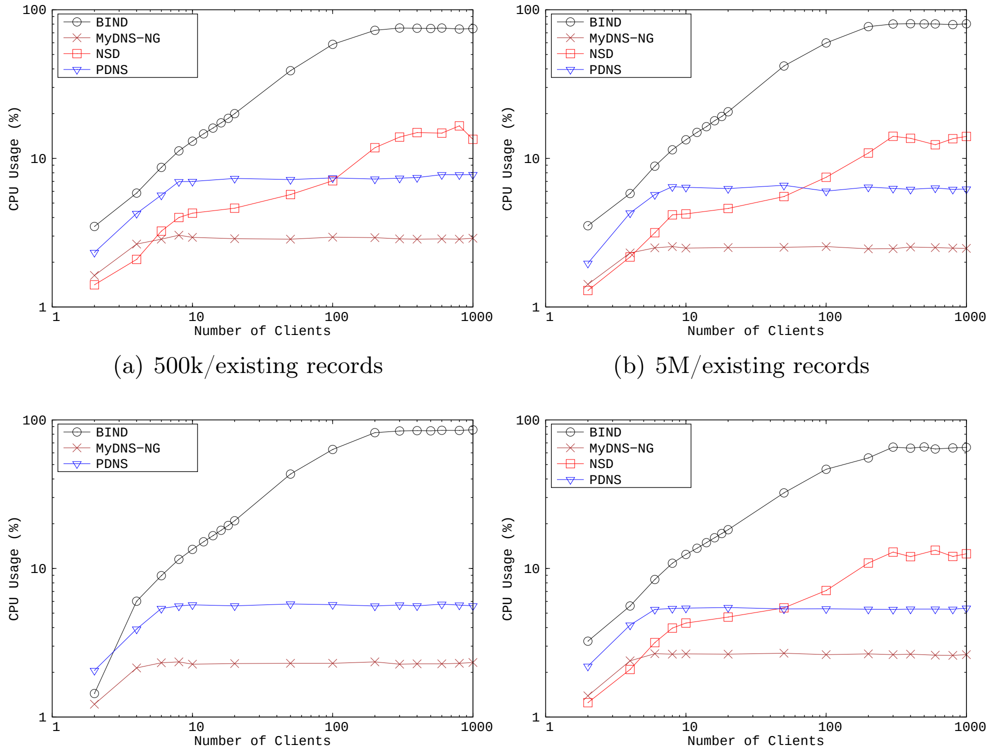
Figure 4: Loss rate considering 5M/existing records.

that cause these different levels of CPU and memory resource utilization are out of the scope of this work.

We also observe that there is only a slight variation in CPU usage when we evaluate different size of database and different types of records as well.

Table 3 shows memory usage for each ENUM server. BIND and NSD are based on zone files and, as a consequence, the greater the size of database, the greater the consume of RAM. On the other hand, MyDNS-NG and PowerDNS rely on MySQL as their backend. Thus, the RAM usage is smaller than BIND and NSD, with small variation when the database is increased.

In order to verify if other variables interfered on the results, two auxiliary metrics were considered: DNSPerf CPU usage and network usage. Figure 6(a) shows that only part of the total CPU resources were used by DNSPerf on the client machine, excluding the possibility of



(a) 500k/existing records

(b) 5M/existing records

(c) 50M/existing records

(d) 5M/non-existing records

Figure 5: CPU usage.

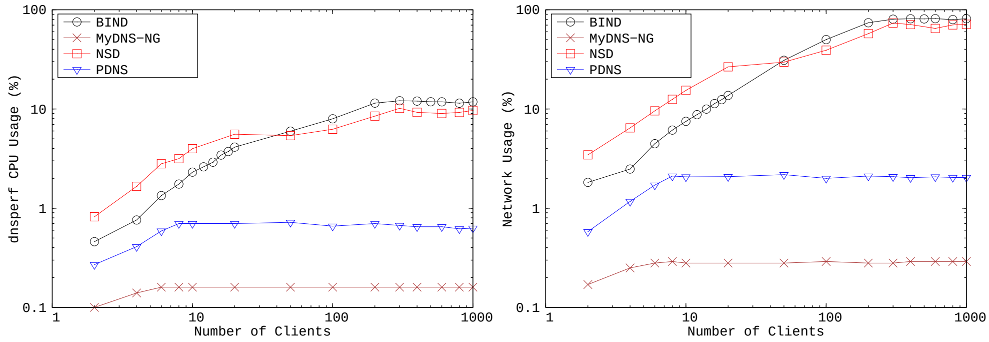
Table 3: Memory Usage.

Records	BIND	MyDNS-NG	NSD	PowerDNS
500k	1.1%	6.9%	1.2%	8.6%
5M	8.6%	6.9%	11.5%	8.9%
50M	82.8%	6.9%	Failed	9.8%

saturation of hardware resources on the client. Similarly, Figure 6(b) illustrates the network usage, and we can observe that the average value of usage has not overpassed the maximum capacity of the link (1000 Mbps).

4.2 Performance for different types of database

Until now, BIND and NSD has overcome MyDNS-NG and PowerDNS in all scenarios. These servers represent two different approaches of database for name servers, namely, zone files and MySQL. In this context, we built a new scenario to evaluate the effect of these different approaches in the performance of the server. To accomplish this task, we chose PowerDNS that is the most flexible server among the evaluated servers. PowerDNS can work with a great number of different backends. Therefore, we evaluated this server with two different databases: zone files and MySQL. Figure 7 shows the evaluation results. PowerDNS had a better performance when we used zone files. However, even with this improvement, PowerDNS performance is quite lower than BIND performance. This occurs mainly because PowerDNS has a poor CPU usage.



(a) DNSPerf CPU usage (b) Network usage

Figure 6: DNSPerf CPU and network usage considering 5M/existing records.

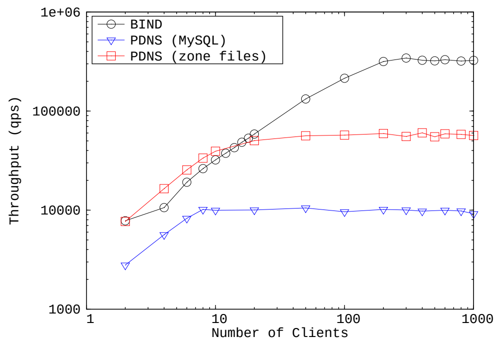


Figure 7: Query throughput performance for PowerDNS (MySQL and zone file backends) and BIND (zone files backend).

4.3 Performance considering the use of swap memory

Despite the great performance of servers based on zone files, they have some drawbacks. Each time the name server is restarted, it must load all records to RAM. Depending on the size of the record set, this can take a long time. Moreover, in ENUM context, the update process of a record stored in zone files is not straightforward, even when we use DNS Dynamic Updates [14]. Other relevant factor is the available RAM of the server. Servers based on zone files rely on RAM to reach high performance. If RAM is not enough to load all the records stored in zone files, the OS will use swap memory to load the rest of data. However, swap memory is quite slow when compared to RAM speed. Therefore, if swap memory is used, the performance of the ENUM server will decrease. In order to study this phenomenon, we evaluated BIND in a scenario where the RAM is not enough to load all records in the zone files. For this scenario we used 100M of existing records. Figure 8 shows the result. As we have foreseen, the query throughput is severely affected. The throughput using swap memory is about 300 times lower than in the scenario using only RAM.

4.4 Performance considering multiples NAPTR records per FQDN.

ENUM protocol is responsible for mapping E.164 telephone numbers into URIs. In this context, the E.164 telephone number is converted into a FQDN for querying process. However, one FQDN can be mapped into multiples NAPTR records, according to available services for a given user. Thus, it is interesting to know the behavior of ENUM servers in a scenario where

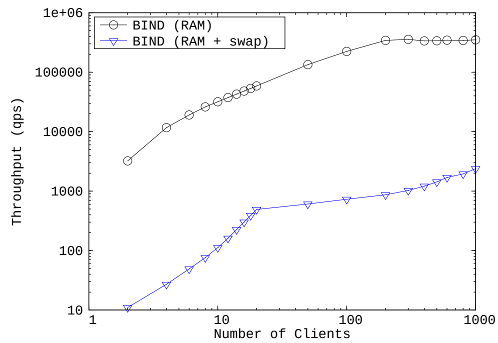


Figure 8: BIND using only RAM and RAM plus swap memory.

there are multiples NAPTR records for each FQDN.

In our evaluation, the number of NAPTR records per FQDN can be different for each server, since each of these servers presented distinct sizes of response messages. The number of NAPTR records in each response message was incremented until the size of response message was near to 512 bytes, which is the typical maximum value of a UDP segment. If a message is bigger than 512 bytes, this message is truncated. In this scenario we used only existing records and the 5M record set.

Figure 9 shows the evaluation results. In Figure 9(a), BIND presents the same performance for both evaluated cases until 100 clients. From this point, for 7 NAPTR/FQDN curve, the throughput remains constant, indicating a possible saturation state. However, analyzing Figure 10(a), it is possible to see that the saturation occurred on the communication link and not on the server. In this case, it would be necessary a link with more capacity to get reliable results. On the other hand, this experiment shows that BIND has a great performance even with more NAPTR records to manage. Figures 9(b) to 9(d) show that the other servers were affected by the increase of response message size. Differently from BIND case, for these servers the communication link was not saturated (see Figures 10(b) to 10(d)).

4.5 Comparison with related work

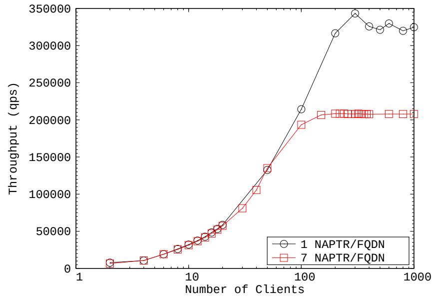
The comparison between different research works is not an easy task, because of different methodologies or lack of details about the experiment. Nevertheless, [5] and [6] are similar works that allowed us to perform some comparison to verify the relevance of our results.

Table 4 shows the comparison of the approximated maximum throughput for the three researches. In this comparison, our results and results from [5] are related to a 5M record set. For [6], the results correspond to a 1M record set. For all cases we used existing records.

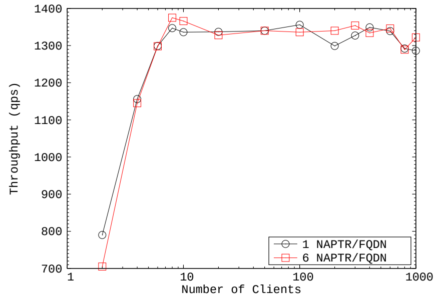
Table 4: Comparison of the maximum throughput.

	Maximum Throughput (kqps)		
Work	BIND	NSD	PowerDNS
this	325	275	10
[5]	0.650	not evaluated	5.5
[6]	50	40	not evaluated

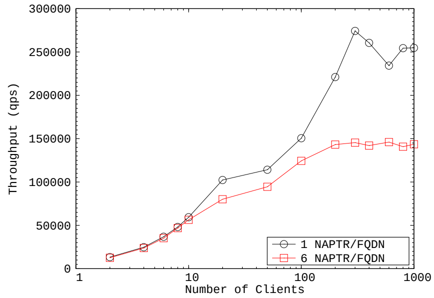
From Table 4 we observe the huge increase of throughput that we achieved in our experiments. Results are specially surprisingly for BIND due to its great evolution. Authors in [5] evaluated BIND version 9.2.4 and classified it as not suitable for ENUM due to its poor scaling properties.



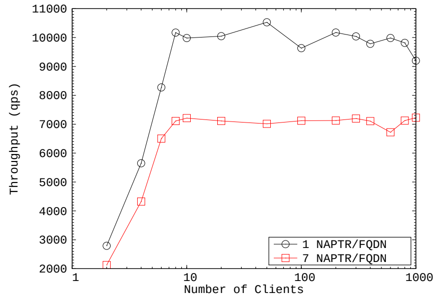
(a) BIND



(b) MyDNS-NG

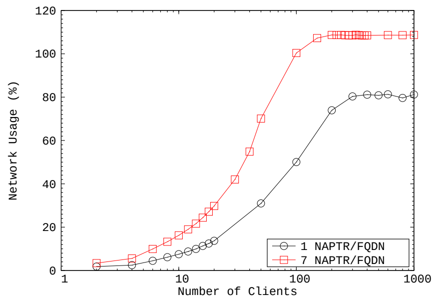


(c) NSD

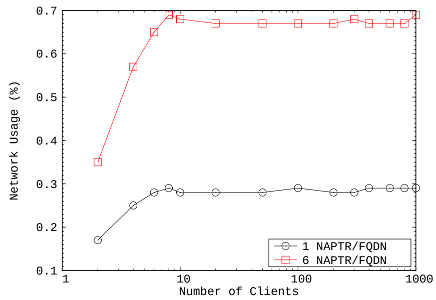


(d) 800 items

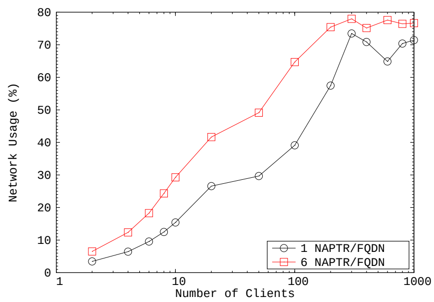
Figure 9: Query throughput performance considering multiples NAPTR records per FQDN.



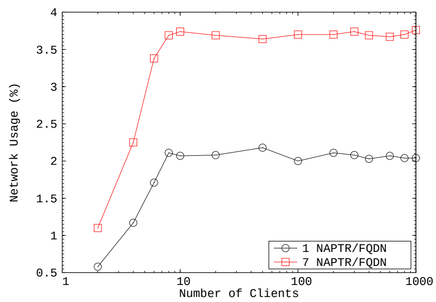
(a) BIND



(b) MyDNS-NG



(c) NSD



(d) PowerDNS

Figure 10: Network usage considering multiples NAPTR records per FQDN.

In [6], the test was performed with BIND version 9.4.0 and the throughput was more than 75 times greater than in [5]. In our evaluation, BIND version 9.8.1 presented the best scaling properties among the evaluated servers and the highest throughput that was 500 times greater than the throughput reported in [5]. NSD also presented a significant evolution from version 3.0.4 to version 3.2.10. It is important to mention that this increase in throughput is also caused by the evolution of the hardware, since we used a more powerful testbed than the former works. These results brings good expectations for the future of ENUM protocol, since the performance of the ENUM servers are getting improved, allowing the deployment of new services.

5 Conclusions

ENUM protocol has been considered as the most promising solution for the integration between the different existing communication networks. In this sense, in this paper, we have selected four different ENUM servers, namely, BIND, MyDNS-NG, NSD and PowerDNS to study their behavior in the new scenarios of deployment. We chose BIND and NSD to represent zone files name servers. MyDNS-NG and PowerDNS were selected to represent name servers with SQL database.

The evaluation was performed in a testbed with high performance machines, and the last stable versions of the ENUM servers selected for the benchmarking. We also developed a new tool to manage our tests. The EnumBenchTool proved to be of fundamental importance to facilitate and validate our work.

In our evaluation scenarios, we extended the range of clients sending queries to the server in a far greater range than the ones used in previous works.

From results, we identified that MyDNS-NG presented the worst performance in all scenarios. We believe that this server is not mature enough to be used in an ENUM context.

PowerDNS performed better than MyDNS-NG. However, due to its inefficient use of CPU and RAM memory resources, it performed quite lower than BIND and NSD.

BIND presented the highest query throughput and a great scalability features. This is a totally new and different result when compared with results from [5]. Our results proved the evolution of this server in the last years.

NSD had the most efficient CPU usage performance. It presented a query throughput similar to BIND, but with a lower memory consume. On the other hand, NSD presented problems to store a huge amount of records, compromising its scalability.

We also evaluated the impact of the response message size in the servers performance. This is a new scenario that was not contemplated in any of previous works. BIND proved its superiority with a better performance when compared with the other ENUM servers.

Another important scenario is the evaluation of the performance of ENUM servers, that use zone files, when the database is greater than the RAM capacity. It is worth to say that zone files allow the server to reach a higher throughput than MySQL only if the name server has enough RAM to load the records stored in the zone files. Moreover, the update process is not straightforward when we use zone files.

Beyond the new results presented in this paper, we also highlight the improvements in the methodology for benchmarking of ENUM servers. We have identified that DNSPerf saturates when only one OS process is used to query the ENUM server. This limitation of DNSPerf brings to light the necessity of improvement in this tool to follow the hardware advances. To the best of our acknowledge, we are the first ones to report this limitation.

In short, we believe that we got important results for the benchmarking of ENUM servers research field. We presented new information about the performance of important ENUM servers and clarified some divergences between former works, as in [5] and [6]. Moreover, we improved the methodology for the benchmarking of ENUM servers and presented results for scenarios that were not contemplated in any other works.

Acknowledgement

The authors gratefully acknowledge financial support from the Studies and Projects Finance Organization (FINEP), under project No. 01.10.0560.00.

Bibliography

[1] ITU-T (2005). The International Public Telecommunication Numbering Plan. ITU Recommendation E.164.

[2] Berners-Lee, T.; Fielding, R.; Masinter, L. (1998); Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396.

[3] Bradner, S.; Conroy, L.; Fugiwara, K. (2011); The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM). RFC 6116.

[4] Shen, C.; Schulzrinne, H. (2007); Measurement and Evaluation of ENUM Server Performance. *IEEE International Conf. on Communications, ICC 07*, 1967-1972.

[5] Shen, C.; Schulzrinne, H. (2006); Evaluation and Comparison of BIND, PDNS and Navitas as ENUM Server. Technical Report cucs-029-06, Columbia University.

[6] Tetenyi, I.; Szabo, G.; Kiss, A.; Toth, A. (2008); ENUM in everyday practice. Is it a dream or an opportunity?. *Infocommunication Journal*, LXIII:48-54.

[7] Rudinsky, J. (2009); Private ENUM based Number Portability Administrative System Evaluation. *International Conference on Ultra Modern Telecommunications Workshops, ICUMT 09*, 1-7.

[8] Lee, H.; Mun, Y. (2004); Performance Evaluation of ENUM Directory Service Design. *Lectures Notes in Computer Science*; 3038:1124-1130.

[9] Balakrichenan, S.; Bugnazet, T.; Becker, M. (2007); Studying ENUM Performance with Modeling and Simulation. *First Asia Int. Conf. on Modelling Simulation, AMS 07*; 82-87.

[10] da Mata, S.H.; Guardieiro, P.R.; Cardoso, A.; Lamounier, E.; Theodoro, L. (2011); Performance Study of ENUM Service Based in Modeling and Simulation. *8th Int. Conf. on Information, Communications and Signal Processing, ICICS 11*; 1-5, doi:10.1109/ICICS.2011.6173557.

[11] da Mata, S. H.; Magalhaes, J. M. H.; Cardoso, A.; Carvalho, H. A.; Guardieiro, P. R. (2013); Performance Comparison of ENUM Name Servers. *22nd International Conference on Computer Communication and Networks (ICCCN2013)*, 1-5.

[12] Nominum (2007). DNSPerf: DNS Performance Tool Manual.

[13] Lin, H. et al (2009); VoIP Signalling Performance Requirements and Expectations. Internet Draft.

[14] Vixie, P. et al (1997); Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136.