Python と vSphere API による業務効率化

吉本 翼・石垣 佐・雨宮尚範・伊藤康広・太田芳博・藤原冨未治

名古屋大学全学技術センター　情報通信技術支援室

情報通信技術支援室では、サーバ仮想化ソフトウェア VMware vSphere（以下、vSphere）を用いて IT サービスの提供を行っている。vSphere の仮想マシン管理業務には Web GUI を使用しており、複数の仮想マシンに対して同一操作を繰り返すような作業が頻繁に発生する。そこで、Python を用いて vSphere API を操作する技術を習得し、業務効率化を目指すことにした。

Key Words : *Python, pyVmomi, vSphere, API, 業務効率化*

１．はじめに

情報通信技術支援室では、年次計画停電の際に、すべての仮想マシンのシャットダウン、パワーオン作業を vSphere の Web GUI で 1 台ずつ行っていた。しかし、仮想マシン数の増加に伴い、業務負荷が大きくなり、人的ミスが発生する可能性が高くなると懸念していた。そこで、作業を自動化するために様々なスクリプトを作成し、業務効率化を目指した。

２．検証用スクリプトの作成

プログラミング言語には、ライブラリが豊富で汎用性の高い Python を採用し、vSphere API を利用するためのライブラリには、pyVmomi を採用した。pyVmomi を用いることで、Python を使って仮想マシンに対する様々な操作を行うことができる。今回用いた Python のバージョンは 3.10、検証環境には ESXi 7.0 を利用した。作成したスクリプトの詳細を以下に示す。

（１）優先度記載スクリプト

シャットダウン、パワーオン作業では、各仮想マシンのサーバ種別に応じてシャットダウン、パワーオンの順番（優先度）を決めておく必要がある。各仮想マシンのサーバ種別に応じて定義した優先度を表―１に示す。

表―１　サーバ種別に応じて定義した優先度

サーバ種別	優先度
Webサーバ	10
転送専用のMailサーバ	20
Mailサーバ	30
送信専用のMailサーバ	40
受信専用のMailサーバ	50
その他	60

表―１に基づき、各仮想マシンの優先度を決定して、Microsoft Excel ですべての仮想マシンの優先度一覧表を作成した。優先度一覧表の例を図―１に示す。

	A	B
1	優先度	仮想マシン
2	10	web_server01
3	10	web_server02
4	20	transfer_mail_server
5	30	mail_server01
6	30	mail_server02
7	30	mail_server03
8	40	smtp_mail_server
9	50	pop_mail_server
10	60	etc

図―１　優先度一覧表の例

決定した各仮想マシンの優先度を、各仮想マシンの注釈（メモ）に記載するスクリプト（以下、優先度記載スクリプト）を作成した。本スクリプトには ReconfigVM_Task メソッドを利用した。ReconfigVM_Task メソッドとは、vSphere の VirtualMachine マネージドオブジェクトで利用できるメソッドであり、仮想マシンオブジェクトの様々なプロパティを編集することができる。本スクリプトの抜粋を図―２に示す。

```
vml=pchelper.get_all_obj(content,[vim.VirtualMachine]) …①
wb=load_workbook("vm_list.xlsx") …②
ws=wb["priority_sheet"]
priority_dict={10:"priority[10]",20:"priority[20]",\
               30:"priority[30]",40:"priority[40]",\
               50:"priority[50]",60:"priority[60]"}
for i in range(ws.min_row+1,ws.max_row+1): …③
  for vm in vml: …④
    if ws.cell(row=i,column=2).value==vm.name: …⑤
      if ws.cell(row=i,column=1).value in priority_dict:
        memo=vim.vm.ConfigSpec(annotation= \
        priority_dict[ws.cell(row=i,column=1).value]) …⑥
        vm.ReconfigVM_Task(memo) …⑦
```

図―２　優先度記載スクリプト（抜粋）

すべての仮想マシンのオブジェクトをリスト型変数の vml に代入して（①）、優先度一覧表を読み込む（②）。次に優先度一覧表の処理対象範囲を決定し（③）、vml から仮想マシンのオブジェクトを 1 つ抽出して vm に代入する（④）。優先度一覧表に記載された仮想マシンの名前が、vm の仮想マシンの名前と一致する場合（⑤）、その仮想マシンに対応する優先度を priority[10]～priority[60]の形式で、文字列型変数の memo に代入して（⑥）、仮想マシンのメモに memo の内容を書き込む（⑦）。

（２）シャットダウンスクリプト

仮想マシンをシャットダウンするためのスクリプト（以下、シャットダウンスクリプト）を作成した。シャットダウンスクリプトでは、ShutdownGuest メソッドを利用した。ShutdownGuest メソッドとは vSphere の VirtualMachine マネージドオブジェクトで利用できるメソッドであり、仮想マシンのシャットダウンを行うことができる。本スクリプトの抜粋を図―３に示す。

```
priority="priority[10]" …①
vml=pchelper.get_all_obj(content,[vim.VirtualMachine])
for vm in vml:
  if vm.summary.runtime.powerState== \
     vim.VirtualMachinePowerState.poweredOn: …②
    if priority==vm.summary.config.annotation: …③
      vm.ShutdownGuest() …④
      while vm.summary.runtime.powerState!= \
      vim.VirtualMachinePowerState.poweredOff: …⑤
        time.sleep(1)
        continue
      time=datetime.datetime.now()
      print(time.strftime("%H:%M:%S")+" "+vm.name) …⑥
```

図―３　シャットダウンスクリプト（抜粋）

文字列型変数の priority にシャットダウン対象の優先度を代入する（①）。次に、起動している仮想マシンを抽出し（②）、その中に priority と同じ優先度が設定された仮想マシンが存在する場合（③）、該当する仮想マシンのシャットダウンを行う（④）。シャットダウンを行った仮想マシンの電源状態を参照し（⑤）、仮想マシンが停止したことを確認し次第、その時刻とシャットダウンを行った仮想マシンの名前を表示する（⑥）。なお、⑤では、vSphere への負荷を軽減するために、1 秒のスクリプト停止時間を設け、電源状態の評価回数を減らしている。

（３）パワーオンスクリプト

仮想マシンをパワーオンするためのスクリプト（以下、パワーオンスクリプト）を作成した。パワーオンスクリプトでは、PowerOnVM_Task メソッドを利用した。PowerOnVM_Task メソッドとは vSphere の VirtualMachine マネージドオブジェクトで利用できるメソッドであり、仮想マシンのパワーオンを行うことができる。本スクリプトの抜粋を図―４に示す。

```
priority="priority[10]" …①
vml=pchelper.get_all_obj(content,[vim.VirtualMachine])
for vm in vml:
  if vm.summary.runtime.powerState== \
     vim.VirtualMachinePowerState.poweredOff: …②
    if priority==vm.summary.config.annotation: …③
      vm.PowerOnVM_Task() …④
      while vm.summary.runtime.powerState!= \
      vim.VirtualMachinePowerState.poweredOn: …⑤
        time.sleep(1)
        continue
      time=datetime.datetime.now()
      print(time.strftime("%H:%M:%S")+" "+vm.name) …⑥
```

図―４　パワーオンスクリプト（抜粋）

文字列型変数の priority にパワーオン対象の優先度を代入する（①）。次に、停止している仮想マシンを抽出し（②）、その中に priority と同じ優先度が設定された仮想マシンが存在する場合（③）、該当する仮想マシンのパワーオンを行う（④）。パワーオンを行った仮想マシンの電源状態を 1 秒ごとに参照し（⑤）、仮想マシンが起動したことを確認し次第、その時刻とパワーオンを行った仮想マシンの名前を表示する（⑥）。

３．スクリプトの実行について

前章で説明したスクリプトを年次計画停電の際に本番環境で実行した結果について以下に示す。

（１）事前作業

事前作業として、優先度記載スクリプトを実行して各仮想マシンのメモに優先度が書き込まれたことを確認した。

（２）シャットダウン作業

シャットダウンスクリプトを実行して、仮想マシンのシャットダウンを行った。仮想マシンのシャットダウンにかかった時間を表―２に示す。

表―２　仮想マシンのシャットダウンにかかった時間

優先度	台数	所要時間	所要時間 (1 台平均)	サーバ種別
10	51 台	13 分 40 秒	16 秒	Web サーバ
20	8 台	1 分 19 秒	10 秒	転送専用 Mail サーバ
30	19 台	12 分 50 秒	40 秒	Mail サーバ
40	15 台	2 分 24 秒	10 秒	送信専用 Mail サーバ
50	14 台	2 分 13 秒	10 秒	受信専用 Mail サーバ
60	53 台	16 分 40 秒	19 秒	その他
合計	160 台	49 分 6 秒	－	－

正常にすべての仮想マシンのシャットダウンが行われた。しかし、検証環境に比べて、シャットダウンが完了するまでに、多くの時間を要した仮想マシンが見られた。特に優先度 30（Mail サーバ）では、1 台平均 40 秒と最も時間がかかっており、シャットダウンスクリプトの実行結果を確認したところ、シャットダウンまでの所要時間が 10 秒の仮想マシンや、2 分 30 秒の仮想マシンが存在した。また、優先度 10（Web サーバ）では 1 台平均 16 秒、優先度 60（その他）では 1 台平均 19 秒と他の優先度の仮想マシンに比べて時間がかかった。

（３）パワーオン作業

パワーオンスクリプトを実行して、仮想マシンのパワーオンを行った。パワーオンの順序は、シャットダウン時の優先度とは逆になる。仮想マシンのパワーオンにかかった時間を表―３に示す。すべての優先度において 1 台平均 1 秒～2 秒でパワーオンが行われた。

表―３　仮想マシンのパワーオンにかかった時間

優先度	台数	所要時間	所要時間 (1 台平均)	サーバ種別
60	53 台	2 分 4 秒	2 秒	その他
50	14 台	15 秒	1 秒	受信専用 Mail サーバ
40	15 台	17 秒	1 秒	送信専用 Mail サーバ
30	19 台	20 秒	1 秒	Mail サーバ
20	8 台	18 秒	2 秒	転送専用 Mail サーバ
10	51 台	1 分 12 秒	1 秒	Web サーバ
合計	160 台	4 分 26 秒	-	-

４．問題点と対応策

本番環境でスクリプトを実行して分かった 3 つの問題点とその対応策を以下に述べる。

1 つ目の問題点は、シャットダウンスクリプトおよびパワーオンスクリプトにおいて、仮想マシンの名前順（アルファベット順）に処理が行われなかったため、進捗状況の把握が困難だったことである。対応策としては、アルファベット順に処理を行い、vSphere の Web GUI でも進捗状況を把握しやすくすることや、「5 台/10 台完了」のような形式で進捗情報をスクリプトの実行画面に表示することなどが挙げられる。

2 つ目の問題点は、検証環境に比べて、シャットダウンスクリプトですべての仮想マシンをシャットダウンするまでに、多くの時間を要したことである。対応策としては、vSphere に影響が出ない範囲で並列処理を行うことが挙げられる。

3 つ目の問題点は、シャットダウンに多くの時間を要した根本的な原因を特定できていないことである。優先度 10 と優先度 60 については、Web アプリケーションを構成するプロセスの停止に時間がかかることが原因だと思われる。しかし、優先度 30 については、稼働しているサービス（Mail）の種類が同じであるにも関わらず、シャットダウンに要する時間に差があった。各仮想マシンのメールユーザ数やディスク使用量などについて調査を行ったが、原因の特定はできなかった。対応策としては、次年度の計画停電時のシャットダウンスクリプト実行中にコンソールの監視を行い、原因の究明を行うことが挙げられる。

５．結論

Python を使って仮想マシンを操作するための多くの知見を得ることができた。また、運用している vSphere 内のすべての仮想マシンに対するシャットダウン、パワーオン作業を自動化することで、昨年度に比べて作業時間の短縮及び人的リソースの有効活用による業務効率化を実現することができた。

参考文献

1）山田 祥寛，『独習 Python』，翔泳社，(2020)

2）“Python 言語リファレンス”，https://docs.python.org/ja/3/reference/index.html（参照 2023/2/14）

3）“VMware vSphere API Reference documentation”，https://developer.vmware.com/ja/apis/1067/vsphere（参照 2023/2/14）

4）“vSphere Web Services SDK Programming Guide”，https://vdc-download.vmware.com/vmwb-repository/dcr-public/d4fd4125-8683-4388-9bf0-7b73c0e5cc34/e5b8ce6d-969f-4e48-af05-d572c08e7b47/vsphere-web-services-sdk-70-ga.pdf（参照 2023/2/14）