

基于浮动利率的 PaaS 平台计费模型研究^①

肖哲彬^{②*} 魏高峰^{**} 张兆心^{③*}

(* 哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

(** 威海广播电视台 威海 264200)

摘 要 研究了 PaaS 平台计费问题。针对传统 PaaS 计费模型在计费策略上忽视平台的资源利用率,导致集群资源效益不高的问题,对基于实际用量的计费方法进行了改进,提出了一种基于浮动利率的计费模型。浮动利率与业务应用的实际资源利用率相关,并受平台资源成本和用户效用约束,在用量结算基础上计算动态利率,从而保证单位资源的效益。实验结果表明,该计费模型能从供需的角度优化平台资源利用率,在保证平台盈利的同时提高总体资源的效益水平。

关键词 PaaS, 容器, 计费模型, 资源效益, 浮动利率

0 引 言

计费系统是实现云服务资源经济价值的必要手段。为了提高服务质量,云服务供应商(service provider, SP)不能简单考虑资源成本与经济效益,更应该关注资源利用率与消费者使用行为。在缺少合适计费模型的条件下,研究如何提高资源效益难以取得实际效果。因此,研究合理的计费模型是一项重要课题。

目前对基于容器的平台即服务(platform as a service, PaaS)平台计费模型研究较少,以下分析了相关的云平台计费策略研究。国防科学技术大学的陈红^[1]等人分析了当前云服务的计费现状和主流平台的计费机制。东北大学的刘国奇^[2]等人研究了与服务等级(service level agreement, SLA)挂钩的基础设施即服务(infrastructure as service, IaaS)动态资源计费策略。湖南大学的马亿昨^[3]等人提出了一种基于用户反馈机制的计费算法,采用插件的形式定制服务,在传统计费的模式上加入了用户反馈

因素,解决了传统计费模型的模式单一、灵活性差等问题。Yuan^[4]等人提出了一种基于租赁实体的计费模型,它降级了供应商成本的同时提高了平台的资源利用率。郭宛^[5]等人提出了一种将服务协议与投资回报率相结合的计费模型,并在 IaaS 平台上很好地平衡了供需关系。以上文献都基于 IaaS 平台研究计费模型,由于 PaaS 平台在总量和实例层使用度量方面与 IaaS 不同,因而 PaaS 收费和计量是根据实际使用确定的,PaaS 计量因子包括传入和传出的网络带宽、每小时的 CPU 使用时间、存储数据量、高可用性、每月服务收费等^[6]。与传统 PaaS 平台不同的是,容器应用需要预先指定资源配额,所以基于容器的 PaaS 平台有着自生的特殊性,在 PaaS 的服务模式下,有着类似 IaaS 的资源申请方式。

1 基于 Docker 的 PaaS 服务平台

在互联网高速发展的环境下,为了及时高效地完成互联网信息获取、探测以及网络安全检测方面的实验,构建了基于 Docker(一个开源的应用容器

① 国家自然科学基金(61100189, 61370215, 61370211, 61402137)和国家科技支撑计划(2012BAH45B01)资助项目。

② 男,1993 年生,硕士;研究方向:云计算与容器技术;E-mail: xzbzwtsjn@163.com

③ 通信作者,E-mail: heart@hit.edu.cn

(收稿日期:2017-06-01)

引擎)的 PaaS 服务平台。基于 Docker 的 PaaS 服务平台是一种开放式的、面向互联网的、可供第三方使用的基础服务平台。平台通过对资源与任务的调配管理,以及根据任务的实际需求和资源的实际情况进行动态聚合与协同工作,提供按需、可定制、易于管理和扩展的公共计算服务能力。该平台分布于全国 31 个省、直辖市的各大运营商,可满足国家中心及分中心对域名安全检测、协议探测、穿透验证以及综合信息获取等方面的功能需求。

本文以基于 Docker 的 PaaS 服务平台为例,进行计费模型的研究。平台基于 ApacheMesos 实现,运行 Chronos 和 Marathon 两种 Framework,分别负责周期任务与服务部署的管理。平台提供命令行和 Web 界面工具支持,系统管理员可以通过上述工具配置、启动、停止业务实例,上传、移除部署的应用,监控资源等。基于 Docker 的 PaaS 服务平台架构如图 1 所示。

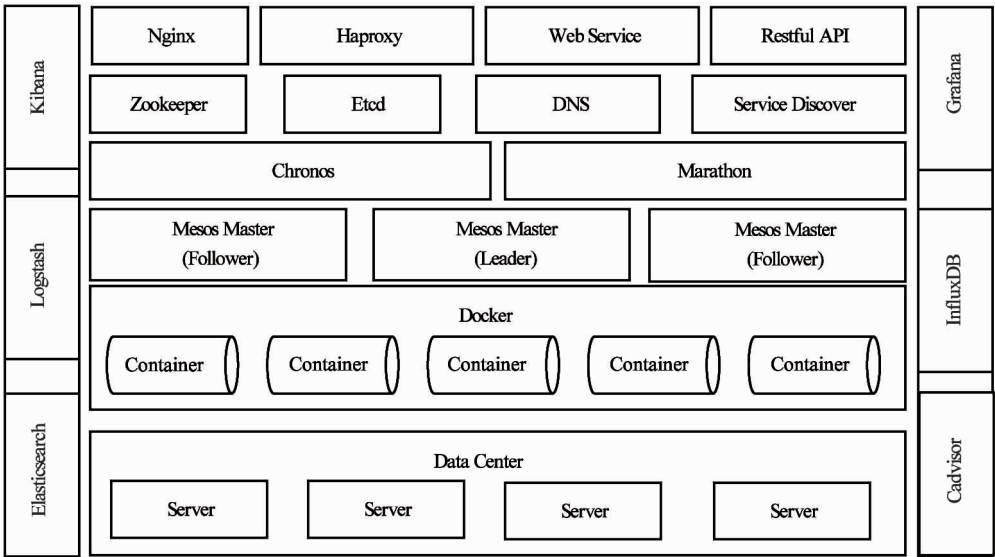


图 1 PaaS 平台架构图

Mesos 将整平台资源资源池化,以资源邀约的方式为 Framework 分配资源^[7]。每个应用通过 Chronos 或 Marathon 申请资源配置,Mesos 从资源池中提取所需的资源分配给应用,应用下发到资源节点上以 Docker 实例的形式运行^[8]。计费因子包括 CPU、内存、网络流量、blockIO、系统服务请求次数等^[9]。

在这种服务模式与资源分配方式下,消费者往往不明确任务所需的资源量,如果申请过多资源将增加自己的成本,并且造成资源浪费,申请较少资源则容易导致业务部署失败。实际上,大多数用户偏向于申请较多资源以避免部署失败,因此平台上总存在部分资源因业务的弹性需求而利用率较低。所以传统 PaaS 的计费方式在容器 PaaS 平台模式下是不可行的。除平台自身资源调节方式外,本文提出

一种基于浮动利率的计费模型,从计费的角度平衡供需关系,优化资源分配,提高资源效益。

2 基于浮动利率的计费模型

由市场经济规律可知,价格上涨将导致商品需求量下降^[10],计费模型的出发点是通过市场调节供需的方式减少资源浪费来提高平台有限资源的利用率。在 PaaS 平台上,所有消费者以接口的形式共享平台资源池,且 PaaS 的资源隔离粒度没有 IaaS 精细,所以平台必须根据盈利目标评估合理的业务应用资源利用率。

平台可提供包括 CPU 资源、内存资源、网络资源在内的 n 项资源。设各项资源的利用率评估权重为 $w = (w_1, w_2, \dots, w_n)$, 单位资源的价格为 $p =$

$(p_1, p_2, \dots, p_n)$ 。设某用户为应用 j 申请的资源配置为 $a = (a_1^j, a_2^j, \dots, a_n^j)$, 应用 j 的实际资源用量为 $u = (u_1^j, u_2^j, \dots, u_n^j)$ 。用户对作业的满意度可以用效用函数表示, 文献[11]解释了效用函数, 用户的效用函数可表示为

$$Utility(x) = \begin{cases} w_s \log_2 x & \alpha = 1 \\ w_s (1 - \alpha)^{-1} x^{1-\alpha} & \alpha > 1 \end{cases} \quad (1)$$

其中 x 表示获得的服务, w_s 表示用户愿意付出的费用, $Utility(x)$ 表示效用函数, 在 $\alpha = 1$ 时效用函数又称为对数函数^[11]。本文评估固定服务下计费对效用的影响, 故取 $\alpha = 1$ 为条件, 于是效用函数可写成

$$Utility(w_s) = \frac{k \log_2 j}{w_s} \quad (2)$$

其中 k 为常数因子, w_s 与 $Utility$ 成反比例, 当服务一定时, 用户花费越大效用越低。设用户的效用区间为 $[Utility_{\min}, Utility_j]$, 即对用户而言, 存在一个效用下界, 当效用值低于下界时, 用户对任务的花费是很不满意的, 对平台而言可能存在损失用户的风险。

定义 1: 应用的综合资源利用率(q): 应用综合资源利用率可以表示一个应用实例在其生命周期中对资源的综合使用情况, 评价方式如下式:

$$q = \sum_{i=1}^n \frac{u_i}{\alpha_i} w_i \quad (3)$$

定义 2: 平台综合利用率阈值(Q): 平台综合利用率阈值由投资回报率 (return on investment, ROI)^[12] 决定, 业务应用达到综合利用率阈值即表示平台在该业务应用上达到了盈利指标, 可以表示为

$$ROI = f(Q), Q \propto ROI \quad (4)$$

定义 3: 多用户潜在收益率(δ): 多用户潜在收益率表示同等资源分配给多个用户的情况下比全部分配给单一用户多盈利 δ 。

在传统 PaaS 的计费模式下 (按用量计费), 平台在应用 j 上获得的收益为 $\sum_{i=1}^n u_i^j \cdot p_i$, 在这种计费模式下, 平台在计费中没有考虑用户申请的资源量 $\sum_{i=1}^n a_i^j$, 如果作业 j 的资源利用率很低, 即存在很多闲置的资源, 就可能影响平台资源利用率而达

不到平台的投资回报率。事实上, 业务应用的资源利用率应该在某个合理的区间范围内, 过高可会导致业务应用崩溃, 过低则浪费资源。所以平台需要区分应用的资源利用率水平差异化计费。

定义 4: 浮动利率: 对于不同资源利用率的应用, 采取的计费方法如下:

$$Cost = \begin{cases} \sum_{i=1}^n u_i^j \cdot p_i & q \geq Q \\ (1 + (1 + \delta) \frac{Q - q}{q}) \sum_{i=1}^n u_i^j \cdot p_i & q < Q \end{cases} \quad (5)$$

其中 Q 指平台的综合资源利用率阈值。 q 为应用的实际资源利用率, q 由式(3)计算得出。对于综合资源利用率大于 Q 的应用仍采取按配置计费的方法, 对于资源利用率不足 Q 的应用采取浮动利率的方式进行计费, 式(5)中用量费用的系数 $(1 + (1 + \delta) \frac{Q - q}{q})$ 即为浮动利率, 浮动利率的大小与应用的实际资源利用率有关, 利用率越接近 Q , 利率越小, 反之利率越大。 $Cost$ 应满足的条件为

$$(1 + ROI) \sum_{i=1}^n a_i^j c_i \leq Cost \leq \frac{k \log_2 j}{Utility_{\min}} \quad (6)$$

式(6)中等式左边为申请资源的成本, 等式右边为用户效用函数的下界。现实意义是, $Cost$ 在满足平台的 ROI 指标的情况下, 应该参考用户的效用范围。如果计费结果无解, 则平台需重新评估低资源利用率风险与用户损失风险。基于浮动利率 (surge rate) 的计费算法如下所示:

```

Input
( $Q_p, (a_1^j, a_2^j, \dots, a_n^j), (p_1, p_2, \dots, p_n), (w_1, w_2, \dots, w_n), \varphi, \delta$ )
//输入资源利用率阈值  $Q_p$ , 申请的资源量  $a_n^j$ , 资源价格  $p_n$ , 审计权重  $w_n$ , ROI 值  $\varphi$ , 潜在利润  $\delta$ 。

Output

$p$  //计费结果

BEGIN
   $p \leftarrow null$ 
   $Set(metrics) \leftarrow container$  // 用量采集
  if ( $metrics.T > deadline$ ) // 任务超时
     $invalid\ task\ and\ exits$  // 任务违规, 启动其他处理方案, 算法退出
  else
    for  $i$  in  $Set(S)$  do
       $Q_j += \frac{metrics_i}{a_i} w_i$  // 计算任务的资源利用率

```

```
end for
for i in Set(metrics) do
    Pusage += Pi · metricsi //按用量计费
end for
if (Qj > Q)
    return Pusage //按用量计费
else
    β ← (1 + (1 + δ) ·  $\frac{Q_p - Q_i}{Q_p}$ )Pusage

    if (![(1 + φ) ∑i=1n αi ci,  $\frac{k}{Utility_{min}}$ ])
        //区间不存在,平台需要评估低资源利用风险与损失用户风险,再决定计费策略
        Return (Low resource utilization risk > User lost risk? β:  $\frac{k}{Utility_{min}}$ )
        //根据当前风险值返回计费结果
        While (β >  $\frac{k}{Utility_{min}}$  && User lost risk > Low resource utilization risk)
            δ = δ - μ //适当减小 δ
            β ← (1 + (1 + δ) ·  $\frac{Q_p - Q_i}{Q_p}$ )Pusage
        //重新计算浮动利率并计费
    end while
    return β
end if
end if
end
```

3 实验及结果分析

实验首先从消费者个体的角度出发,分析浮动利率对消费者不同资源申请行为的影响,然后从平台的角度出发,分析基于浮动利率的计费方法带来的平台收益。

实验所用环境为基于 Docker 的 PaaS 服务平台原型,由 6 台 x86 服务器组成,其中 3 台作为 Master 节点,3 台作为 Slave 节点,Master 节点部署各种调度服务的主控部分,Slave 节点部署监控程序。数据采集与存储通过 Cadvisor 与 InfluxDB 部署实现。为了更直观地展示实验结果,实验采用控制变量法,避免多种因素共同作用干扰实验结果准确性,由于内存、磁盘和网络等因素在实验中变化相对较为稳定,可以归一化处理,所以控制应用和平台的内存、磁盘、网络带宽等因素不变,以 CPU 因素来评价应用和平台资源使用率,各服务器的配置见表 1。

表 1 服务器配置表

服务器	CPU 核心数 (core)	内存配置 (GB)	带宽 (Mbps)	磁盘配置 (TB)
Master1	24	32	100	1
Master2	24	32	100	1
Master3	24	32	100	1
Slave1	24	32	100	1
Slave2	24	32	100	1
Slave3	24	32	100	1

3.1 用户计费实验

目前基于 Docker 的 PaaS 服务平台以爬虫类探测业务为主,所以计费实验以爬虫应用为对象。实验控制爬虫的内存和带宽,内存和带宽分别为 512MB,5Mbps,分别在 0.5 核,1 核,2 核,4 核,8 核,16 核的资源申请配额下进行实验。设该爬虫应用在按用量计费的计费模式下,结算费用为 h 元,为了便于计算,设 $h = 1$,用户的效用函数为: $Utility(x) = 1/x$, x 的区间为 $[1, 2]$,平台利用率阈值 Q_0 为 0.45,潜在利润 δ 为 0.5。实验中爬虫应用的运行时间和运行时间内的平均资源利用率以及本文计费模型下的结算费用如表 2 所示。

表 2 爬虫应用计量结果

核数	运行时长 (s)	应用资源利用率 (%)	费用 (元)
0.5	86	96.8	1
1	87	93.2	1
2	86	44.42	1.02
4	87	33.0	1.54
8	87	30.95	1.68
16	86	28.7	1.85

由实验结果可知,在任务正常调度执行的情况下,任务的执行时长与 CPU 核数无关。在 0.5 核与 1 核 CPU 配置下,资源利用率均高于设定的阈值,所以按实际用量计费,利率都为 1。而 CPU 配置大于 2 核时,综合资源利用率不足 0.45,产生的浮动利率大于 1 但都没有超过用户最低效用值。而在传统 PaaS 的计费方式下,费用只与实际用量相关,所以费用始终为 1 元。由此可知,在本文计算方法下,

业务应用的资源利用率越低,产生的费用越高,用户申请过多的资源不会有效减少任务的执行时间,反而因空闲资源过多影响平台其他应用部署而增加开销。

3.2 平台收益实验

实验 3.1 验证了用户在不同资源利用率下的部署应用开销情况,说明了浮动利率对用户个体的影响。本节实验在实验 3.1 的基础上验证平台在用户群体不同资源申请规模下的资源利用率和平台收益情况。仍以实验 3.1 中的爬虫为实验对象,实验环境与各参数设置同实验 3.1。实验设计三种爬虫,内存和带宽分别为 512MB 和 5MBps,CPU 资源申请配额分别为 4 核,2 核,0.5 核。三种配额的爬虫数量配比见表 3,模拟平台在日常资源申请时遇到的各种情况。对 5 次实验分别以本文的方法计费并与传统 PaaS 的计费方法作对比分析结果,资源配置表如表 3 所示。

表 3 实验配置表

爬虫资源	A	B	C
配额	(4 核-512MB)	(2 核-512MB)	(0.5 核-512MB)
实验 1	17	1	4
实验 2	13	8	8
实验 3	8	10	24
实验 4	4	4	96
实验 5	1	1	132

各组实验中平台采集的资源利用率变化如图 2 所示,两种计费模式下平台的收益如图 3 所示,在不同资源利用率下本文计费方式与传统计费方式的收益比如图 4 所示。在实验 1 中,平台资源申请状况不合理,导致平台运行的任务量较低,此时平台的 CPU 平均利用率为 4.45%,按本文计费方法可以得出计费结果为 $17 \times 1.54 + 1 \times 1.06 + 1 \times 4 = 31.2$ 元,而传统 PaaS 计费结果为 $22 \times 1 = 22$ 元,这种情况下,本文的计费结果明显高于传统 PaaS 的计费结果,来保证平台利益。从实验 2 到实验 5,高配置的爬虫数量逐渐减少,低配置的爬虫数量增多,平台的资源利用率逐渐提升,分别为 8.7%, 19.72%, 45.7% 和 61.39%,随着资源利用率的提升,平台收益明显提高,两种计费方式产生的费用也逐渐接近,

实验结果印证了本文计费模型在平台高利率情况下计费结果与传统 PaaS 相近,而在平台低利用率下,本文计费方式将对低资源利率的任务提高收费,从而计费结果高于传统 PaaS,保证资源的效益。基于浮动利率的计费模型可以在保证平台盈利的条件下提高资源的效益水平。

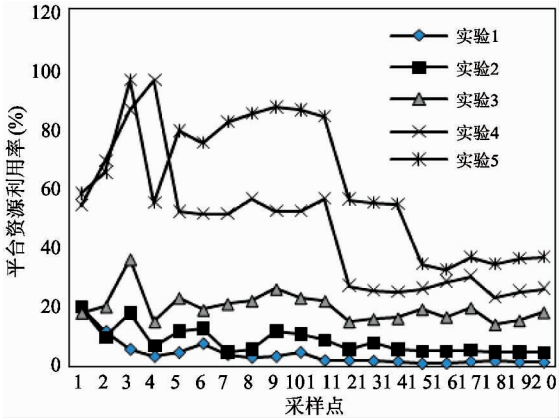


图 2 5 组实验 CPU 利用率比较图

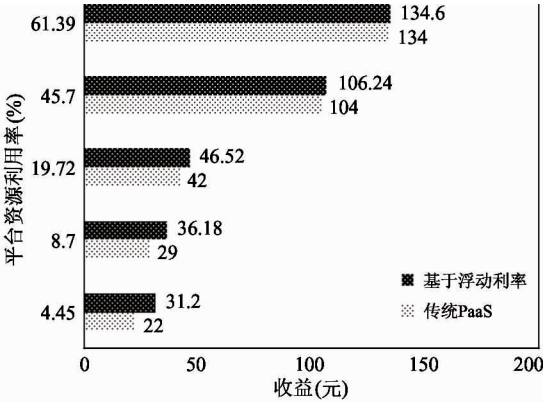


图 3 收益对比图

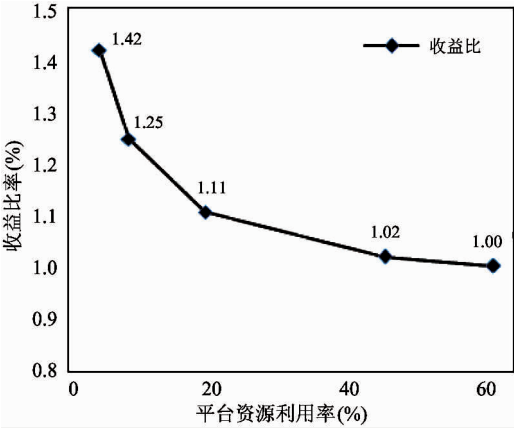


图 4 收益比率图

4 结 论

本文分析了现有 PaaS 计费模式存在的问题,并提出了一种基于浮动利率的计费模型。在计费模型上,借鉴传统 PaaS 的计费方法,并做出修改使其适用于容器 PaaS 环境。实验结果表明,基于浮动利率的计费模型能通过计费调节供需平衡,在保证平台盈利的情况下,提高资源的收益水平。然而,基于浮动利率的计费方法还只是一种尝试性的方法,目前仅考虑了爬虫业务应用,事实上不同类型的应用对资源偏好侧重点不同,在适用性方面可能仍存在一些需要改进的地方,这也是接下来本文要进行的工作。

参考文献

- [1] 陈红, 任怡, 刘晓建. 云计算平台下计费机制研究[J]. 计算机科学, 2011, 38(8):48-52
- [2] 刘国奇, 刘慧, 高宇, 等. 基于效用的云计算动态资源计费策略[J]. 吉林大学学报(工), 2013, 43(6): 1631-1637
- [3] 马亿昨, 池鹏, 陈磊, 等. 基于用户反馈机制的超级计算 CAE 云平台计费模型设计[J]. 中南大学学报自然科学版, 2015, 46(10):3718-3725
- [4] Yuan Q, Liu Z, Peng J, et al. A leasing instances based billing model for cloud computing[C]. In: Proceedings of the 6th International Conference on Advances in Grid and Pervasive Computing, Oulu, Finland, 2011. 33-41
- [5] 郭婉, 张晓. 一种基于 SLA 与 ROI 结合的 IaaS 动态计费模型[J]. 计算机研究与发展, 2014, 51(1):86-90
- [6] Meiers J. Cloud computing billing[EB/OL]. <http://www.ibm.com/developerworks/cloud/library/cl-cloud-metering>; IBM, 2011
- [7] Hindman B, Konwinski A, Zaharia M, et al. Mesos: a platform for fine-grained resource sharing in data center [C]. In: Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, Boston, USA, 2013. 429-483
- [8] Saha P, Govindaraju M, Marru S, et al. Integrating apache airavata with Docker, Marathon, and Mesos[J]. *Concurrency and Computation: Practice and Experience*, 2016, 28(7): 1952-1959
- [9] 任怡, 张菁, 陈红, 等. 云应用引擎的资源监控和计费机制研究[J]. 通信学报, 2012, 33(1):192-200
- [10] 琼·罗宾逊, 约翰·伊特韦尔. 现代经济学导论[M]. 北京:商务印书馆, 1982. 212-213
- [11] 罗杰, 张之明, 高志强, 等. 基于效益最大化的云虚拟机资源分配研究[J]. 计算机应用研究, 2016, 33(10):2963-2966
- [12] Information Systems Audit and Control Association. Calculation cloud ROI: from the customer perspective[EB/OL]. <http://www.isaca.org/Knowledge-Center/Research/ResearchDeliverables/Pages/Calculating-Cloud-ROI-From-the-Customer-Perspective.aspx>; Isaca, 2016

Research of billing model for PaaS platform based on surge rate

Xiao Zhebin*, Wei Gaofeng**, Zhang Zhaoxin*

(* School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

(** Weihai Radio and Television Station, Weihai 264200)

Abstract

The billing for PaaS platform is studied. In consideration of the fact that the cluster resource benefit of the traditional PaaS billing model is low because its billing strategy neglects the resource utilization, a new billing model based on surge rate is proposed through improving the billing method based on the actual amount used. The surge rate is related to the actual resource utilization of transaction and constrained by the platform resource cost and user utility, so it guarantees to improve the benefit of unit resource by calculating the dynamic rate based on actual usage. The experimental results show that the proposed billing model could optimize platform's resource utilization from the perspectives of supply and demand, and improve the benefit of platform resources under the premise of guaranteeing the profit.

Key words: PaaS (platform as a service), container, billing model, resource efficiency, surge rate