Capacity of Distributed SDC Resources

Pengchao Han (Imperial), Shiqiang Wang (IBM US), Kin K. Leung (Imperial),
Kevin Chan (ARL), and Don Towsley (UMass)

***Abstract*—In the Software Defined Coalition (SDC), computation, communications and memory resources are distributed across multiple domains and can be used to execute analytics. It is challenging to characterize the capacity of such distributed resources because of the randomness of required resources by tasks. In this work, we derive analytical formulas for the upper bound of capacity of distributed systems with multiple resources. Resource allocation methods including random assignment, power of d choices assignment and the least occupancy first assignment are considered to help analyze and approximate the capacity of distributed SDC resources. The capacity results are useful for describing the remaining capacity of distributed resources in SDC or distributed computing systems in general. They can also be used to assist scheduling and admission decisions of distributed analytics to various resources in the systems. Numerical study is included to validate of the capacity upper bounds.**

I. Introduction

In the Software Defined Coalition (SDC slice), computation, communications and memory resources are distributed across multiple domains and can be used to execute analytics. The resource capacity of one SDC slice is necessary information for other SDC slices to realize cooperated resource scheduling and admission control. However, it is challenging to characterize the resource capacity of SDC slice due to the random resource requirements of tasks and distributed nature of resources. In this work, we derive analytical formulas for the upper bound of capacity of distributed SDC resources. The moment generating function of resources is utilized to characterize the random resource requirements. Moreover, three task assignment approaches, namely random assignment, power of d choices assignment and the least occupancy first assignment are taken into account to assist the derivation of resource capacities. The capacity results are useful for describing the remaining capacity of distributed resources in SDC or distributed computing systems in general. They can also be used to assist scheduling and admission decisions of distributed analytics to various resources in the systems.

This work is organized as follows. Section II describes system models and the problem. The resource capacities based on three task assignment approaches are formulated respectively in Section III, followed by numerical evaluations in Section IV. Finally, Section V concludes this work.

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

This paper includes preliminary results for internal discussion at the Annual Fall Meeting of DAIS ITA in 2018. A paper related to this work has been published subsequently [3].

II. System Models and Problem Description

A. System and Task Models

We consider a system (SDC slice) with multiple and distributed resources running multiple tasks. There are N servers, each has K types of resources (i.e., computation, communications and memory) with capacity $c_{n,k}$ for server n resource k. Assume all servers are identical, being equipped with same capacity for all resources. The resource requirements are normalized such that $c_{n,k} = 1, \forall n, k$. Moreover, there are M tasks. The requirement of task m for resource k is a random variable $X_{m,k}$ with $X_{m,k} \in [0, 1]$. We use μ_k and σ_k^2 to denote the mean and variance of requirements of all tasks for resource k and let $\mu^{\max} = \max_k \{\mu_k\}$ and $\sigma^{\max} = \max_k \{\sigma_k\}$.

B. Problem Description

We emphasize on how to characterize capacity to enable efficient use of distributed SDC resources belonging to different coalition members. In our consideration, each task should be assigned to only one server in an SDC slice, which will allocate resources to the task for service provisioning. Note that different task assignment approaches bring about different performance related to the number of tasks the SDC slice can serve. Therefore, we aim to analyze the maximum number of tasks with multiple and random resource requirements can be served using different task assignment approaches in the SDC slice.

Three task assignment approaches are considered. The random assignment (RAND) randomly allocates M tasks to N servers without any additional communication among different servers. Second, the power of d choices assignment (PODC) randomly chooses $d \geq 2$ servers for each task and assigns the task to the least-occupied one among the d servers. For multiple resources in SDC slice, the server with the minimum maximum occupancy of all K resources will be chosen among the d selected servers. Resource information will be exchanged between these d servers and the task holder to determine the least-occupied server. It has been proved that having just two choices yields a large reduction on the maximum load over having one choice, while each additional choice beyond two decreases the maximum load by just a constant factor [4]. Therefore, PODT is an effective way to achieve load balancing while generating little resource monitoring overheads, especially when $d = 2$. Third, the least occupancy first assignment (LOFA) assigns each task to one server while minimizing the maximum occupancy among all resources on all servers. LOFA requires all resource information of servers, resulting in the best load balancing but the highest communication overheads due to resource monitoring and information exchanges.

Based on these three task assignment approaches, the resource capacity is defined as the maximum number of tasks M that the system can serve simultaneously, so that the overload probability is no more than a small value ε:

$$\Pr\left\{\bigcup_{k=1}^{K}\left(\bigcup_{n=1}^{N}\left[\rho_{n,k}\geq 1\right]\right)\right\}\leq\varepsilon \tag{1}$$

where $\rho_{n,k}=\sum_{m=1}^{M}X_{m,k}I_{m,n}$ denotes the occupancy of resource k on server n and $I_{m,n}$ equals 1 if task m is assigned to server n, and 0 otherwise.

III. Resource Capacities

In this Section, the resource capacities is analyzed and the upper bounds is formulated for different task assignment approaches. It is worth mentioning that our analytical bounds are sufficient conditions for resources capacities with the given overload probability ε. Namely, if the number of tasks is smaller than the bounds, the overload probability in (1) can be guaranteed, but the reverse may not be true. Applying Boole's inequality to (1), we have

$$\begin{aligned}&\Pr\left\{\bigcup_{k=1}^{K}\left(\bigcup_{n=1}^{N}\left[\rho_{n,k}\geq 1\right]\right)\right\}\\&\leq\sum_{n=1}^{N}\sum_{k=1}^{K}\Pr\left(\rho_{n,k}\geq 1\right).\end{aligned} \tag{2}$$

Assume resource requirements of all tasks for each resource are independent and identical distributed with the same probability distribution and different distributions apply to different resources. We use $F_k(\theta)=E\left(e^{\theta X_{m,k}}\right)$ with parameter $\theta>0$ to denote the moment generating function of $X_{m,k}$. Using Chernoff's bound, we have

$$\Pr\left(\rho_{n,k}\geq 1\right)\leq\frac{E\left(e^{\theta\rho_{n,k}}\right)}{e^{\theta}}\ \text{for}\ \theta>0. \tag{3}$$

Moreover, we can get an upper bound for (2) as

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\Pr\left(\rho_{n,k}\geq 1\right)\leq NK\Pr\left(\rho_{n}^{\max}\geq 1\right), \tag{4}$$

where $\rho_{n}^{\max}=\max_k\left\{\rho_{n,k}\right\}$. By combining (3) and (4), we have

$$E\left(e^{\theta\rho_{n}^{\max}}\right)\leq\frac{\varepsilon e^{\theta}}{NK},\theta>0, \tag{5}$$

which is a sufficient condition for the overload probability in (1). For RAND and PODC, we will calculate the maximum moment generating function of occupancy of servers to achieve resource capacities while satisfying the overload probability in (5). While for LOFA, we will directly analyze (1) for resource capacity derivation.

A. Random Assignment

For any server n, by considering $\rho_{n,k}=\sum_{m=0}^{M}X_{m,k}I_{m,n}$, we have

$$\begin{aligned}&E\left(e^{\theta\rho_{n,k}}\right)\\&=E\left[E_I\left(e^{\theta\sum_{m=0}^{M}X_{m,k}I_{m,n}}|X_{1,k},\cdots,X_{M,k}\right)\right]\\&=E\left[\prod_{m=0}^{M}E_I\left(e^{\theta X_{m,k}I_{m,n}}|X_{m,k}\right)\right],\end{aligned}$$

where $E_I[\cdot]$ depicts the mean of $I_{m,n}$. For any task m, there exist $\Pr(I_{m,n}=1)=1/N$ and $\Pr(I_{m,n}=0)=1-1/N$ in RAND. Thus, $E_I\left(e^{\theta X_{m,k}I_{m,n}}|X_{m,k}\right)=1+\left(e^{\theta X_{m,k}}-1\right)/N$. Therefore,

$$\begin{aligned}E\left(e^{\theta\rho_{n,k}}\right)&=E\left[\prod_{m=0}^{M}\left(1+\frac{\left(e^{\theta X_{m,k}}-1\right)}{N}\right)\right]\\&=\prod_{m=0}^{M}\left(1+\left(F_k(\theta)-1\right)/N\right).\end{aligned}$$

Let $F^{\max}(\theta)=\max_k\left\{F_k(\theta)\right\}$. Combing the above with (5), we get

$$\prod_{m=0}^{M}\left(1+\left(F^{\max}(\theta)-1\right)/N\right)\leq\frac{\varepsilon e^{\theta}}{NK}.$$

Taking the logarithm on both sides of the above equation, it yields

$$M\log\left(1+\frac{1}{N}\left(F^{\max}(\theta)-1\right)\right)\leq\log\frac{\varepsilon}{NK}+\theta.$$

This gives

$$M\log\left(1+\tfrac{1}{N}\left(F^{\max}(\theta)-1\right)\right)^{N}/N\leq\log\tfrac{\varepsilon}{NK}+\theta. \tag{6}$$

For $F^{\max}(\theta)-1>0$ and positive N, it is true that $\left(1+\left(F^{\max}(\theta)-1\right)/N\right)^{N}\leq\exp\left(F^{\max}(\theta)-1\right)$ [2]. Thus, by approximating the left side of (6) using the above equation, we have

$$M\log e^{F^{\max}(\theta)-1}/N\leq\log\tfrac{\varepsilon}{NK}+\theta,$$

and finally

$$M\leq\frac{N\left(\log\left(\frac{\varepsilon}{NK}\right)+\theta\right)}{F^{\max}(\theta)-1}. \tag{7}$$

Eq. (7) specifies the resource capacity of multi-resource distributed systems that captures the maximum moment generating function of resource requirements over all resources $F^{\max}(\theta)$.

In a special case, we take Gaussian distribution for example to show how to achieve numerical resource capacity. In Gaussian distribution, moment generating function of resource k is $F_k(\theta)=\exp\left(\mu_k\theta+\sigma_k^2\theta^2/2\right)$ and we have $F^{\max}(\theta)=\exp\left(\mu^{\max}\theta+\left(\sigma^{\max}\theta\right)^2/2\right)$. Applying $F^{\max}(\theta)$ to (7), we have

$$M\leq\frac{N\left(\log\frac{\varepsilon}{NK}+\theta\right)}{\exp\left(\mu^{\max}\theta+\left(\sigma^{\max}\theta\right)^2/2\right)-1}. \tag{8}$$

By analyzing the derivative of the right-hand side of (8) with respect to θ, we find the right-hand bound increases first when θ goes higher and reaches the maximum point at θ^*, after which a decreasing trend is showed. Therefore, we can achieve the maximum M by taking θ as θ^*:

$$\theta^*=\frac{-(\sigma^{\max}\log\frac{\varepsilon}{NK}+\mu^{\max})+\sqrt{\Delta}}{2\left(\sigma^{\max}\right)^2}. \tag{9}$$

where $\Delta=(\sigma^{\max}\log\frac{\varepsilon}{NK}+\mu^{\max})^2-4\left(\sigma^{\max}\right)^2(\mu^{\max}\log\frac{\varepsilon}{NK}-1)$.

B. Power of d Choices Assignment

In PODC, it has been proved that the maximum number of tasks on any server is $M/N + \log\log N/\log d$ with high probability for $M \gg N$ [1]. It is reasonable to assume $M \gg N$ as we assume each server can serve multiple tasks simultaneously. Thus,

$$E\left(e^{\theta\rho_n^{\max}}\right) = \left[F^{\max}(\theta)\right]^{(M/N+\log\log N/\log d)}.$$

Applying the above equation to (5), we have

$$\left[F^{\max}(\theta)\right]^{(M/N+\log\log N/\log d)} \leq \frac{\varepsilon e^{\theta}}{NK}, \theta > 0.$$

It yields

$$M \leq N\left(\frac{\log\frac{\varepsilon}{NK}+\theta}{\log F^{\max}(\theta)} - \frac{\log\log N}{\log d}\right). \tag{10}$$

Considering Gaussian distributed resources as a special case for example, we have

$$M \leq \frac{N\left(\log\frac{\varepsilon}{NK}+\theta\right)}{\mu^{\max}\theta + \left(\sigma^{\max}\right)^2\theta^2/2} - \frac{N\log\log N}{\log d}. \tag{11}$$

The derivative of the right part of (11) with respect to θ is

$$\frac{\left(\sigma^{\max}\theta\right)^2/2 + \left(\sigma^{\max}\right)^2\theta\log\frac{\varepsilon}{NK} + \mu^{\max}\log\frac{\varepsilon}{NK}}{-\left(\mu^{\max}\theta + \left(\sigma^{\max}\theta\right)^2/2\right)^2/N}, \tag{12}$$

where the numerator is a quadratic equation with one unknown. Two real roots, one negative and another positive, exist for the quadratic equation, which again means the right part of (11) increases when θ goes higher and reaches the maximum point when θ equals to the larger root θ^*. Therefore, we can obtain the maximum M when

$$\theta^* = \frac{\sigma^{\max}\log\frac{\varepsilon}{NK} - \sqrt{\Delta'}}{-\sigma^{\max}}, \tag{13}$$

where $\Delta' = \left(\sigma^{\max}\log\frac{\varepsilon}{NK}\right)^2 - 2\mu^{\max}\log\frac{\varepsilon}{NK}$.

C. Least Occupancy First Assignment

In LOFA, each task is assigned to the server with the minimum maximum occupancy among all resources at the time of the assignment. Define the k-height of task m, $h_{m,k}$ by the occupancy of resource k on server n where task m is assigned up to the time when tasks 1 to m have been assigned, namely $h_{m,k} = \sum_{n=1}^{N}\sum_{i=1}^{m}\left(X_{i,k}I_{i,n}I_{m,n}\right)$. Moreover, the super height of task m is defined as the maximum k-height among all resources for m, i.e., $h_m^{\max} = \max_k\{h_{m,k}\}$. Assume the task with the highest super height in the system is the last task indexed by M. Obviously $\rho^{\max} = \max_{n,k}\{\rho_{n,k}\} = h_M^{\max}$. Furthermore, denote a_M the server that task M is assigned, i.e, $I_{M,a_M} = 1$. According to LOFA, at the time task M is going to be assigned, the maximum occupancy of servers in $\{n = 1, \cdots, N, n \neq a_M\}$ is at least $\rho^{\max} - \max_k\{X_{M,k}\}$. Otherwise, task M would not be assigned to a_M if there exists other severs with lower maximum occupancy than a_M. In addition, denote κ_n the type of resource that has the maximum occupancy on server n at the time that task M is going to be assigned. Note that κ_n may be different for different servers. Let b_n be the number of tasks on server n at the time task M is going to be assigned and obviously $\sum_{n=1}^{N} b_n = M-1$. Thus, we have

$$\sum_{n=1}^{N}\sum_{m=1}^{b_n} X_{m,\kappa_n} \geq N\left(\rho^{\max} - \max_k\{X_{M,k}\}\right). \tag{14}$$

That is,

$$\rho^{\max} \leq \frac{\sum_{n=1}^{N}\sum_{m=1}^{b_n} X_{m,\kappa_n}}{N} + \max_k\{X_{M,k}\}. \tag{15}$$

In addition, the overload probability in (1) can also be expressed as

$$\begin{aligned}\Pr\left\{\bigcup_{k=1}^{K}\left(\bigcup_{n=1}^{N}\left[\rho_{n,k}\geq 1\right]\right)\right\} &\leq \Pr\left(\rho^{\max}\geq 1\right)\\ &= \Pr\left(N\rho^{\max}\geq N\right) \leq \varepsilon.\end{aligned}$$

Using the Chernoff bound to the above equation, we have

$$\Pr\left(N\rho^{\max}\geq N\right) \leq \frac{E\left(e^{\theta N\rho^{\max}}\right)}{e^{\theta N}} \leq \varepsilon \text{ for } \theta > 0. \tag{16}$$

We then calculate $E\left(e^{\theta N\rho^{\max}}\right)$ from (16) as

$$\begin{aligned}&E\left(e^{\theta N\rho^{\max}}\right)\\ &= E\left(e^{\theta\left(\sum_{n=1}^{N}\sum_{m=1}^{b_n}X_{m,\kappa_n} + N\max_k\{X_{M,k}\}\right)}\right)\\ &= E\left(e^{\theta\left(\sum_{n=1}^{N}\sum_{m=1}^{b_n}X_{m,\kappa_n}\right)}\right)E\left(e^{\theta N\max_k\{X_{M,k}\}}\right).\end{aligned} \tag{17}$$

The third line is due to the fact that $X_{m,k}$ is independent for different tasks. Moreover,

$$\begin{aligned}E\left(e^{\theta\left(\sum_{n=1}^{N}\sum_{m=1}^{b_n}X_{m,\kappa_n}\right)}\right) &\leq \left(F^{\max}(\theta)\right)^{\sum_{n=1}^{N}b_n}\\ &= F^{\max}(\theta)^{(M-1)},\end{aligned} \tag{18}$$

$$E\left(e^{\theta N\max_k\{X_{M,k}\}}\right) \leq F_{NX}^{\max}(\theta), \tag{19}$$

where $F_{NX}^{\max}(\theta)$ is the maximum moment generating function of $NX_{m,k}$ among all resources. Therefore, Eq. (17) can be expressed as

$$E\left(e^{\theta N\rho^{\max}}\right) \leq F^{\max}(\theta)^{(M-1)}F_{NX}^{\max}(\theta). \tag{20}$$

Combined with (16), we can obtain the resource capacity:

$$M \leq \frac{\log\varepsilon + \theta N - \log F_{NX}^{\max}(\theta)}{\log F^{\max}(\theta)} + 1. \tag{21}$$

For Gaussian distribution, the resource capacity is

$$M \leq \frac{\log\varepsilon + N\left(1-\mu^{\max}\right)\theta - \left(N\sigma^{\max}\right)^2\theta^2/2}{\mu^{\max}\theta + \left(\sigma^{\max}\right)^2\theta^2/2} + 1. \tag{22}$$

The derivative of the right part of (22) with respect to θ is $\left[-N\left(\sigma^{\max}\right)^2\left(1-\mu^{\max}+N\mu^{\max}\right)\theta^2/2 - \left(\sigma^{\max}\right)^2\log\varepsilon\theta - \mu^{\max}\log\varepsilon\right] / \left(\mu^{\max}\theta + \left(\sigma^{\max}\right)^2\theta^2/2\right)^2$, where the numerator is a quadratic equation with one unknown. Similar to the analysis of (12), M reaches its

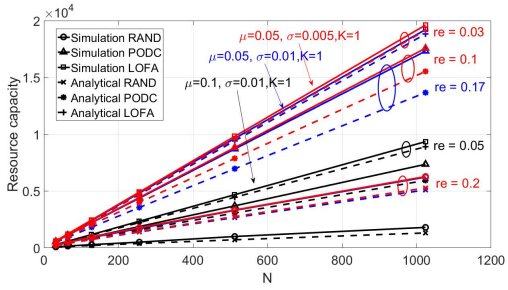

Fig. 1. Comparison of resource capacity with different μ and σ

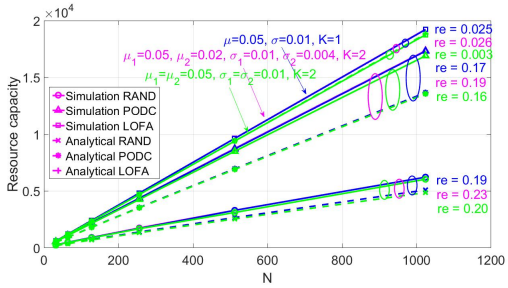

Fig. 2. Comparison of resource capacity with different K

TABLE I
RELATIVE ERROR WHEN $K = 1, \mu = 0.1, \sigma = 0.05$

N	2^5	2^6	2^7	2^8	2^9	2^{10}
RAND	0.3	0.37	0.35	0.32	0.31	0.32
PODC	0.5	0.51	0.55	0.59	0.63	0.67
LOFA	0.1	0.07	0.06	0.05	0.04	0.03

TABLE II
RELATIVE ERROR WHEN $K = 1, \mu = 0.05, \sigma = 0.01$

N	2^5	2^6	2^7	2^8	2^9	2^{10}
RAND	0.2	0.21	0.19	0.18	0.19	0.18
PODC	0.1	0.16	0.18	0.19	0.2	0.21
LOFA	0	0.03	0.03	0.03	0.02	0.02

TABLE III
RELATIVE ERROR WHEN $K = 1, \mu = 0.05, \sigma = 0.005$

N	2^5	2^6	2^7	2^8	2^9	2^{10}
RAND	0.2	0.2	0.21	0.19	0.17	0.16
PODC	0.1	0.08	0.09	0.1	0.11	0.12
LOFA	0	0.03	0.03	0.03	0.03	0.02

maximum value when

$$\theta^* = \frac{\sigma^{\max}\log\epsilon - \sqrt{\Delta''}}{-N\sigma^{\max}\left(1-\mu^{\max}+N\mu^{\max}\right)}, \tag{23}$$

where $\Delta'' = \left(\sigma^{\max}\log\varepsilon\right)^2 - 2N\left(1-\mu^{\max}+N\mu^{\max}\right)\mu^{\max}\log\varepsilon$.

IV. NUMERICAL RESULTS

We evaluate the tightness of above analytical bounds in (8), (11) and (22) for a wide range of number of servers. By setting $\varepsilon = 0.01, d = 2$, the comparison of resource capacity with different means and variances is shown in Fig. (1) where "re" stands for the average relative error over N values. It can be observed that all resource capacities increase linearly with an increasing number of servers. Moreover, it is obvious that LOFA achieves the highest resource capacity compared with other two approaches, between which the resource capacity of PODC is higher. Furthermore, the lower mean and variance lead to more tasks to be served through comparing performance curves with different colors in the figures. In addition, considering the relative error of analytical bounds compared with simulation results, LOFA performs best, followed by PODC and RAND respectively, which can also be verified in Tables (II) and (III). However, the analytical bound of PODC may not have acceptable relative error for the parameter settings under consideration when the mean and variance of required resources are high, as shown in Table (I).

The comparison of resource capacity for multiple resources is shown in Fig. (2). Multiple resources result in less capacity due to the randomness of required resources. However, the difference of resource capacity among different multiple resource scenarios are negligible as long as the maximum mean and variance stay same. Similar relative errors are also showed for different multiple resource scenarios with same maximum mean and variance. Overall, the difference between the derived bounds and the simulation results are acceptable for scheduling and admission decisions.

V. CONCLUSION

We focused on the capacity of distributed SDC resources considering different task assignment approaches. Analytical bounds have been derived as sufficient conditions to guarantee a given overload probability constraint in the SDC slice. The numerical results have verified the tightness of the derived bounds, which perform better when the mean and variance of required resources are reduced. Our capacity bounds for distributed resources results can be helpful for the scheduling and admission control of systems with inter-cooperated SDC slices with different degrees of resource-monitoring and communication overheads. In the future, splittable tasks will be emphasized by considering the incurred communication cost among distributed sub-tasks. The equivalent capacity of distributed systems with multiple resources supporting the splittable tasks will be analyzed to assist a more flexible resource scheduling and admission control.

REFERENCES

[1] Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. Balanced allocations. *SIAM J. Comput.*, 29(1):180–200, September 1999.

[2] Randy Cogill. Randomized load balancing with non-uniform task lengths. 2007.

[3] Pengchao Han, Shiqiang Wang, and Kin K Leung. Capacity analysis of distributed computing systems with multiple resource types. In *IEEE WCNC*. IEEE, 2020.

[4] Michael Mitzenmacher, Andrea W. Richa, and Ramesh Sitaraman. The power of two random choices: A survey of techniques and results. In *in Handbook of Randomized Computing*, pages 255–312, 2000.