March 2021

A Homotopy Type Method for the Purposes of Cloud Resource Management

Lei Liu

A Homotopy Type Method for the Purposes of Cloud Resource Management

Abstract

This disclosure describes techniques to manage cloud resources by categorical generalization and by homotopy type deductions and constructions. Per the techniques, a foundation, a framework, and an internal-homotopy are developed for cloud resource management. Cloud resource management is developed by giving direct access to the induction principal without application of small object argument. An inner anodyne with a division algorithm transforms infinite resources to finite usage measures. An additive and a multiplicative structure implements the inner anodyne. The techniques leverage topoi, similar to an alternative universe, such that homotopy can be applied to cloud resource management. Inner fibration is developed by eviction procedure. A block-based eviction procedure constructs cardinals to solve sizing issues with a sufficient supply of universes. Minimal models implement the framework where evictions construct terminals. Both inner anodyne of structures and inner fibration of evictions are stable. Examples are provided of the framework of cloud resource management frame as minimal models with stability.

Keywords

- Homotopy
- Cloud resource-usage
- Categorical generalization
- Grothendieck universe
- Grothendieck construction
- Grothendieck topoi
- Fibration
- Anodyne morphism

- Resource eviction
- Cache eviction

1 Background

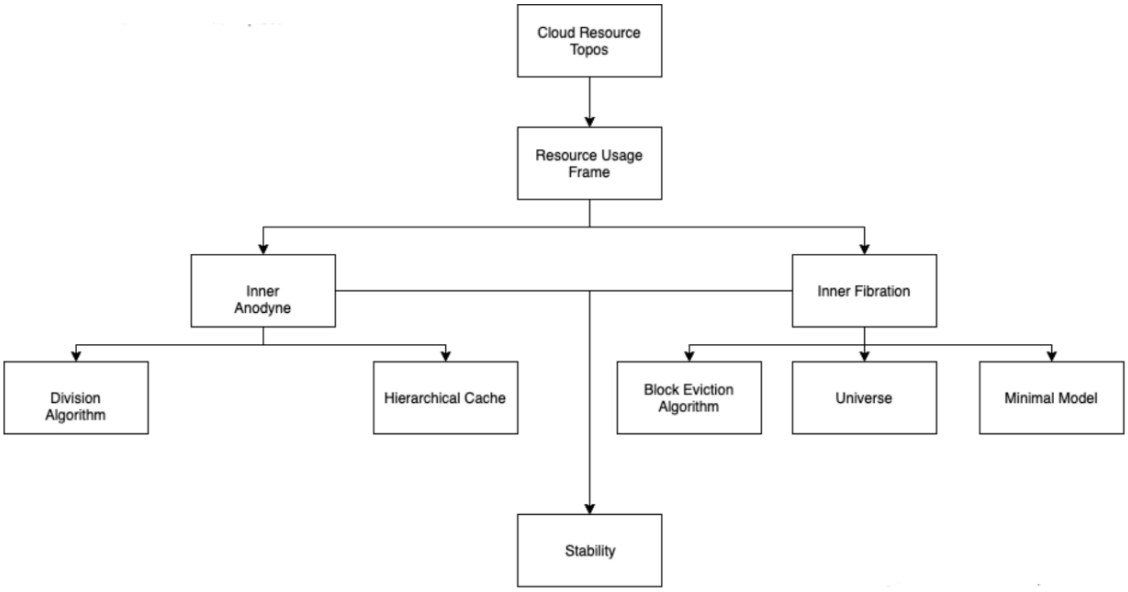


Figure 1: Cloud Resource Management Topoi Framework

1.1 Measure Resource Utilization

An important difference between cloud resources and classical computing theory is that resource usage provides a synthetic description of services in the following sense. Cloud resources are infinite and consequently synthetic in the style of Euclid: one starts from some basic notions (points and lines), constructions (a line connecting any two points), and axioms (all right angles are equal), and deduces consequences logically. This is in contrast with analytic computing resources, where notions such as points and lines are represented concretely using cartesian coordinates in $\mathbb{R}^n$—lines are sets of points—and the basic constructions and axioms are derived from this representation. While classical computing theory is analytic (spaces and paths are made of points), cloud computing resource types are synthetic: points, paths, and paths between paths are basic, indivisible, primitive notions. Hence, classical computing analytics such as throughput and Ops are not sufficient to measure cloud resource utilization.

1.2 Sizing Issues

In ordinary computing theory, one frequently encounters data in which the collection of objects is too large to form computation. Generally speaking, this does not create any difficulties so long as anything computationally disallowed is avoided (such as considering the "category of all categories" as data). The same issues arise in the setting of higher cloud computing theory and are in some sense even more of a nuisance. In ordinary computing theory, one generally allows a category $\mathbf{C}$ to have a proper class of objects but still requires $\mathrm{Hom}_{\mathbf{C}}(X, Y)$ to be a set for fixed objects $X, Y \in \mathbf{C}$. The formalism of ∞-categories of cloud resources treats objects and morphisms on the same footing (they are both simplices of a simplicial set), and it is somewhat unnatural (though certainly possible) to directly impose the analogous condition. Hence, throttling of request rates does not resolve sizing issues for cloud resource utilization.

2 Description

2.1 Universal Cover

In cloud computing, where $\mathbb{S}^1$ is regarded as a topological space, the winding map $w : R \to \mathbb{S}^1$ looks like a helix projecting down onto the circle. This map w sends each point on the helix to the point on the circle that it is sitting above. It is a fibration, and the fiber over each point is isomorphic to the integers. Lifting the path that goes counter-clockwise around the loop on the bottom, results in an elevation by one level in the helix, incrementing the integer in the fiber. Similarly, going clockwise around the loop on the bottom corresponds to going down one level in the helix, decrementing this count. This fibration is called the universal cover of the circle, illustrated in Figure 2.

Theorem 1. $\pi_1(\mathbb{S}^1) = \mathbb{Z}$

Corollary 1.1. *Cloud resource usage is homotopic equivalent to* $\mathbb{Z}$*.*

Example 2.1. *An implementation of an algebraic system* $\mathbb{Z}$ *to measure cloud resource utilization.*

Example 2.2. *An implementation of integer computing as resource utilization counters.*

2.2 Division Algorithm

Lemma 2. *In finite algebraic system* $\mathbb{Z}_n$*, for given integers* k *and* n *with* $n > 0$*, the division algorithm for* $\mathbb{Z}$ *determines integers* q *and* r *uniquely such that* $k = qn + r$*, where* $0 \leq t \leq n$*.*

Corollary 2.1. *For* n *fixed, the division algorithm can be used to map each integer* k *onto its remainder with modulo* n *as:*$\rho : \mathbb{Z} \to \mathbb{Z}_n$*. These remainders modulo* n *can be added and multiplied.*

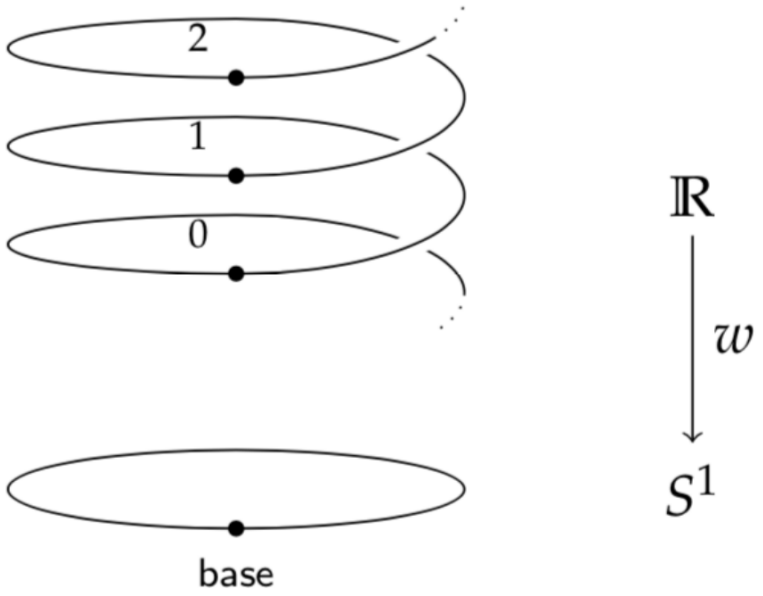


Figure 2: Universal Cover

Theorem 3. *If $\rho : \mathbb{Z} \to \mathbb{Z}_n$ is the function assigning to each integer its remainder modulo n, there is exactly one binary operation $\bigoplus$ on $\mathbb{Z}_n = \{0, \cdots, n-1\}$ such that $\rho(k+m) = (\rho k) \bigoplus (\rho m)$ for all integers k and m. This operation $\bigoplus$ is commutative, associative, and has zero as unit. To each $r \in \mathbb{Z}_n$ there is an $\rho' \in \mathbb{Z}_n$ with $r \bigoplus r' = 0$.*

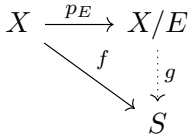
Theorem 4. *There is exactly one binary operation $\bigotimes$ on $\mathbb{Z}_n$ with $\rho(km) = (\rho k) \bigotimes (\rho m)$ for all integers k and m. This operation is commutative, associative, distributive over $\bigoplus$, and has 1 as unit.*

In this construction of the finite algebraic system $\mathbb{Z}_n$, the fact that $\rho : \mathbb{Z} \to \mathbb{Z}_n$ is a surjection satisfying the above theorems for addition and multiplication is important, while the choice of the remainder $\{0, 1, \cdots, n-1\}$ as the elements of $\mathbb{Z}_n$ is incidental. Alternatively, one frequently replaces each remainder r by the set of all integers with the remainder r. This set is residue class and the coset of an integer k with modulo n. Let $\mathbb{Z}/n$ denote the set of all these cosets. The following diagram commutes:

$$\begin{array}{ccc} \mathbb{Z} & \xrightarrow{p} & \mathbb{Z}/n \\ & \searrow^{\rho} & \downarrow \theta \\ & & \mathbb{Z}_n \end{array}$$

Theorem 5. *If E is an equivalence relation on a set X, the projection $p_E : X \to X/E$ is a surjection with equivalence kernel E.*

Theorem 6. *If E is an equivalence relation on a set X, let $f : X \to S$ be any function such that xEy implies $fx = fy$. Then there is exactly one function $g : X/E \to S$ for which $f = g \circ p_E$. If f is a surjection and $fx = fy$ implies xEy, then g is a bijection.*



Lemma 7. *For the set T of all triangles t in the plane, let a binary relation S be defined by tSt' if and only if t and t' are similar. The quotient set T/S may be regarded as the set of all possible shapes of triangles.*

Lemma 8. *For points (x, y) in the real coordinate plane $\mathbb{R}^2$, define $(x, y)E(x', y')$ to mean that $x - x'$ and $y - y'$ are both integers. E is an equivalence relation and the quotient set $\mathbb{R}^2/E$ may be described as the set of points on a torus.*

Example 2.3. *A division algorithm to transform infinite resources to finite resource management measures.*

Example 2.4. *A measurement of cloud resource implemented by the division algorithm.*

Example 2.5. *An implementation of the cache structure as the equivalence kernel.*

Example 2.6. *An implementation the quotient structure as the resource keys.*

Example 2.7. *An implementation of the modulo, the residue class, and the coset structure as the resource elements.*

Example 2.8. *An implementation of the addition and multiplication of remainders modulo as hierarchical layers of resource structures.*

```
Class Utilization (key, action, path, time, status)
Class ResourceKey (key, useddate)
Class ResourceValueKey (action, path, time, status)
Cache[ResourceKey, Map[ResourceValueKey, LongAdder]]
```

Example 2.9. *Where addition is union of resource cosets and multiplication is the hierarchy of resources, the resource structure is important while the resoruce elements are incidental.*

2.3 Topoi Generalization

A frame is a category of open subsets in a space generalized from topological space: a locale which is defined to be anything that has a collection of open subsets that behaves essentially like the open subsets of a topological space.

Definition 1. *A frame $\mathcal{O}$ is a poset that has all small coproducts called joins $\bigvee$ and all finite limits called meets $\wedge$ and which satisfies the infinite distributive law: $x \wedge (\bigvee_i y_i) \leq \bigvee_i (x \wedge y_i)$ for all x, $\{y_i\}_i$ in A.*

Remark. *Join means the depth in the size of a Set but not proper class. Hence, there is an arbitrary co-limit. The finite limit for meets means width should be finite. Infinite distributive law expresses that a frame has universal colimits: they are stable under pullback. In a poset pullbacks and products coincide.*

Definition 2. *A frame is naturally equipped with the structure of a site: a family of morphism $\{U_i \to U\}$ is covering precisely if U is the union of the U_i as $\bigvee_i U_i = U$.*

Definition 3. *Let $\mathcal{U}$ be a poset. $\mathcal{U}$ is said to be a locale if the following conditions are satisfied: (1) Every subset $\{U_\alpha\}$ of elements of $\mathcal{U}$ has a least upper bound $\bigcup_\alpha U_\alpha$ in $\mathcal{U}$. (2) The formation of least upper bounds commutes with meets in the sense that $(U_\alpha \cap V) = (\bigcup U_\alpha) \cap V$. (Here $(U \cap V)$ denotes the greatest lower bound of U and V , which exists by virtue of assumption.*

Remark. *For every topological space X, the collection $\mathcal{U}(X)$ of open subsets of X forms a locale. Conversely, if $\mathcal{U}$ is a locale, then there is a natural topology on the collection of prime filters of $\mathcal{U}$ which enables the extraction of a topological space from $\mathcal{U}$. These two constructions are adjoint to one another, and in good cases they are actually inverse equivalences. More precisely, the adjunction gives rise to an equivalence between the category of spatial locales and the category of sober topological spaces. In general, a locale can be regarded as a sort of generalized topological space in which one may speak of open sets but one does not generally have a sufficient supply of points.*

Remark. *In classical topos theory, there are functorial constructions for passing back and forth between topoi and locales. Given a locale U (such as the locale $\mathcal{U}(X)$ of open subsets of a topological space X), one may define a topos X of sheaves (of sets) on $\mathcal{U}$. The original locale $\mathcal{U}$ may then be recovered as the partially ordered set of subobjects of the final object of X. In fact, for any topos X, the partially ordered set $\mathcal{U}$ of subobjects of the final object forms a locale. In general, X cannot be recovered as the category of sheaves on $\mathcal{U}$; this is true if and only if X is a localic topos: that is, if and only if X is generated under colimits by the collection of subobjects of the final object 1_X. A generalization of this picture enables the passing between m-topoi and n-topoi for any $m \leq n$.*

Remark. *The notion of frame is a generalization of the notion of category of open subsets of a topological space.*

Remark. *A frame is a lattice which has not only finite joins but all small joins. As the distributive law certainly holds when the joins in question are finite, it is a distributive lattice.*

Corollary 8.1. *The notion of frame or rather its formal dual, the notion of locale-is the special case of the notion of* $(n, 1)$*-toposes for* $n = 0 : (0, 1)$*-toposes.*

Remark. *The notion of* $(0, 1)$*-toposes is essentially equivalent to that of Heyting algebra;*

Theorem 9. *Given the existence of finite limits and arbitrary colimits, the infinite distributive law expresses that a frame has universal colimits: they are stable under pullback.*

Remark. *In a poset, pullbacks and products coincide.*

Corollary 9.1. *A frame is automatically a Heyting algebra.*

Corollary 9.2. *A frame is naturally equipped with the structure of a site: a family of morphism* $\{U_i \to U\}$ *is covering precisely if* U *is the union of the* U_i*:* $\bigvee_i U_i = U$*.*

Corollary 9.3. *A complete decidable linear order is a frame.*

Example 2.10. *An implementation of the cloud resource usage frame by constructing the inner anodyne of resource structures.*

2.4 Small Object Argument

Corollary 9.4. *Any function* $f : X \to S$ *can be written as a composite* $f = g \circ p$ *such that* $X \xrightarrow{p} X/E_f \xrightarrow{g} S$ *with* g *injective and* p *the projection to a quotient set; namely, to the quotient set of* X *by the equivalence kernel* E_f *of* f*.*

The small object argument shows that every map $X \xrightarrow{p} Y \xrightarrow{q} Z$ of simplicial sets admits a factorization, where p is anodyne (left anodyne, right anodyne, inner anodyne, a cofibration) and q is a Kan fibration (left fibration, right fibration, inner fibration, trivial fibration).

Example 2.11. *An implementation of constructed inner anodyne as division algorithmic resource structures*

Example 2.12. *An implementation of constructed inner fibration as a resource eviction algorithm.*

2.5 Inner Fibration

The notion of inner fibration of simplicial sets is one of the notion of fibrations of quasi-categories. When the notion of $(\infty, 1)$-category is incarnated in terms of the notion of quasi-category, an inner fibration is a morphism of simplicial sets $C \to D$ such that each fiber is a quasi-category and such that over each morphism $f : d_1 \to d_2$ of D, C may be thought of as the cograph of a profunctor—cograph of an $(\infty, 1)$-profunctor $C_{d_1} \nrightarrow C_{d_2}$.

Corollary 9.5. *When $D = *$ is the point, an inner fibration $C \to *$ is precisely a quasi-category C.*

Lemma 10. *When $D = N(\Delta[1])$ is the nerve of the interval category, an inner fibration $C \to \Delta[1]$ may be thought of as the cograph of a profunctor—cograph of an $(\infty, 1)$-profunctor $C \nrightarrow D$.*

Corollary 10.1. *This $(\infty, 1)$-profunctor comes form an ordinary $(\infty, 1)$-functor $F : C \to D$ precisely if the inner fibration $K \to \Delta[1]$ is even a coCartesian fibration. And it comes from a functor $G : D \to C$ precisely if the fibration is even a Cartesian fibration.*

Remark. *This is the content of the $(\infty, 1)$-Grothendieck construction. And precisely if the inner fibration/cograph of an $(\infty, 1)$-profunctor $K \to \Delta[1]$ is both a Cartesian as well as a coCartesian fibration does it exhibit a pair of adjoint $(\infty, 1)$-functors.*

Remark. *An implementation of the inner fibration as cache eviction procedures.*

```
utilizationAccumulator: Cache[ResourceKey, Map[ResourceValueKey, LongAdder]]
= Resource.newBuilder()
  .maximumSize(ResourceConfig.size)
  .expireAfterWrite(ResourceConfig.writeExpirationTimeSec, TimeUnit.SECONDS)
  .removalListener(new RemovalListener[ResourceKey,
                           Map[ResourceValueKey, LongAdder]]() {
      override def onRemoval(key: ResourceKey,
         value: Map[ResourceValueKey, LongAdder], cause: RemovalCause):
         Unit = {
           service(key, value)(context)
         }
  })
  .build[ResourceKey, Map[ResourceValueKey, LongAdder]]()
```

2.6 Cardinality

There are several means of handling the technical difficulties inherent in working with large objects (in either classical or higher category theory):

1. One can employ some set-theoretic device that enables one to distinguish between large and small. Examples include:
   - Assuming the existence of a sufficient supply of (Grothendieck) universes.
   - Working in an axiomatic framework which allows both sets and classes (collections of sets which are possibly too large for themselves to be considered sets).
   - Working in a standard set-theoretic framework (such as Zermelo-Frankel) but incorporating a theory of classes through some ad hoc device. For example, one can define a class to be a collection of sets which is defined by some formula in the language of set theory.
2. One can work exclusively with small categories. and mirror the distinction between large and small by keeping careful track of relative sizes.
3. One can simply ignore the set-theoretic difficulties inherent in discussing large categories.

Needless to say, approach (2) yields the most refined information. However, it has the disadvantage of burdening our exposition with an additional layer of technicalities. On the other hand, approach (3) will sometimes be inadequate to make arguments which play off the distinction between a large category and a small subcategory which determines it. Consequently, the remainder of this document adopts approach (1). More specifically, assume that for every cardinal κ_0, there exists a strongly inaccessible cardinal $\kappa \geq \kappa_0$. Then let $\mathcal{U}(\kappa)$ denote the collection of all sets having rank $< \kappa$, so that $\mathcal{U}(\kappa)$ is a Grothendieck universe: in other words, $\mathcal{U}(\kappa)$ satisfies all of the usual axioms of set theory. Refer to a mathematical object as small if it belongs to $\mathcal{U}(\kappa)$ (or is isomorphic to such an object), and essentially small if it is equivalent (in whatever relevant sense) to a small object. Whenever it is convenient to do so, choose another strongly inaccessible cardinal $\kappa' > \kappa$ to obtain a larger Grothendieck universe $\mathcal{U}(\kappa')$ in which $\mathcal{U}(\kappa)$ becomes small. For example, an ∞-category $\mathcal{C}$ is essentially small if and only if it satisfies the following conditions:

- The set of isomorphism classes of objects in the homotopy category hC has cardinality $< \kappa$.
- For every morphism $f : X \to Y$ in $\mathcal{C}$ and every $i \geq 0$, the homotopy set $\pi_i(HomRC(X, Y), f)$ has cardinality $< \kappa$.

Block Based Eviction Algorithm

The existence of the strongly inaccessible cardinal κ cannot be proven from the standard axioms of set theory, and the assumption that κ exists cannot be proven consistent with the standard axioms for set theory. However, it should be clear that assuming the existence of κ is merely the most convenient of the devices mentioned above; none of the results proven in this book will depend on this assumption in an essential way.

Example 2.13. *An implementation of a sufficient supply of universes by infinite storage spaces.*

Example 2.14. *An implementation of the cardinal block-based cache eviction procedures.*

```
resourceValue.keys.grouped(Config.spannerWriteBatchSize).toList.map(k => {
 val updatePerMinuteUtilizationBatch = new
 BatchStatement(BatchStatement.Type.UNLOGGED)
 ......
 val updatePerMonthUsageBatch = new BatchStatement(
 BatchStatement.Type.UNLOGGED)

 k.map(valueKey => {
  minuteOfUsage = Date.from(minuteOfInstant)
  ......
  utilization = Date.from(yearOfInstant)
  updateUtilizationBatch.add(updateUtilizatin(valueKey.time).bind(
  Long.box(resourceValue(valueKey).sum), utiization.key))
  ......
})
```

2.7 Minimal Model

Lemma 11. *Let $p : X \to \Delta^n$ be an inner fibration. Then X is an ∞-category. Moreover, p is a minimal inner fibration if and only if X is a minimal ∞-category. This follows from the observation that for any pair of maps $f, f' : \Delta^m \to X$, a homotopy between f and f' is automatically compatible with the projection to Δ^n.*

Lemma 12. *If $p : X \to S$ is a minimal inner fibration and $T \to S$ is an arbitrary map of simplicial sets, then the induced map $X^T = X \times_S T \to T$ is a minimal inner fibration. Conversely, if $p : X \to S$ is an inner fibration and if $X \times_S \Delta^n \to \Delta^n$ is minimal for every map $\eta : \Delta^n \to S$,then p is minimal.*

Corollary 12.1. *For many purposes, the study of minimal inner fibrations reduces to the study of minimal ∞-categories.*

Minimal Block Based Eviction

Corollary 12.2. *An ∞-category C is minimal if the associated inner fibration $C \to *$ is minimal.*

Example 2.15. *A determination of the classification of spaces up to homotopy equivalence.*

Example 2.16. *A generalization of the theory of minimal models in which Kan fibrations are replaced by inner fibrations of cache evictions. Specifically, both block sizes are minimal.*

Example 2.17. *A generalization of the theory of minimal models in which cache evictions are effective to construct terminals.*

2.8 Stability Properties

Stability of Inner Fibrations

The class of left fibrations is stable under various important constructions, such as the formation of slice ∞-categories. left fibrations are stable under the formation of functor categories. The remainder of this document describes the study of the behavior of left fibrations under exponentiation. Our goal is to prove an assertion of the following form: if $p : X \to S$ is a left fibration of simplicial sets, then so is the induced map $X^K \to S^K$, for every simplicial set K.

Corollary 12.3. *Let $p : X \to S$ be a left fibration and let $i : A \to B$ be any cofibration of simplicial sets. Then the induced map $q : X^B \to X^A \times_{S^A} S^B$ is a left fibration. If i is left anodyne, then q is a trivial Kan fibration*

Lemma 13. *Let $p : C \to D$ be a Cartesian fibration of simplicial sets and let $q : K \to C$ be a diagram. Then (1) The induced map $p' : C/q \to D/pq$ is a Cartesian fibration. (2) An edge f of C/q is $p'-$Cartesian if and only if the image of f in C is p-Cartesian.*

Corollary 13.1. *The class of Cartesian fibrations is stable under the formation of over-categories and undercategories.*

Stability of Inner Anodyne

Lemma 14. *Let $p : X \to S$ be an inner fibration and let $i : A \to B$ be any cofibration of simplicial sets. Then the induced map $q : X^B \to X^A \times_{S^A} S^B$ is an inner fibration. If i is inner anodyne, then q is a trivial fibration. In particular, if X is an ∞-category, then so is X^B for any simplicial set B.*

Example 2.18. *An inner anodyne of caching structures and an inner fibration of evictions with stability.*

3 Conclusion

The techniques leverage homotopy theory to treat cloud resource usage as follows. Resource categories are generalized and homotopy types developed to treat cloud resource utilization. The cloud resource management frame is developed by the framework of minimal model with stability. The foundation of cloud-resource usage is developed. Direct

representation of cloud resource management is developed. Minimal models to generalize resource structures and evictions construct terminals are developed. The inner anodyne is characterized with a division algorithm to transform infinite resources to finite usage measures. An additive and a multiplicative resource structure is derived to implement the inner anodyne. The inner fibration is reinforced as implemented by resource eviction algorithm. The internal homotopy of cloud resource usage is developed based upon expirations. A block-based eviction procedure is developed to construct cardinals by solving the sizing issues with a sufficient supply of universes. Both inner anodyne of resource structures and inner fibration of evictions with stability are developed.