Proof of compliance for the relative operator on the proportional distribution of unused share in an ordering fairshare system

Gonzalo P. Rodrigo - gonzalo@cs.umu.se

1 Introduction

Decentralized prioritization is a technique to influence job scheduling order in grid fairshare scheduling without centralized control. The technique uses an algorithm that measures individual user distances between quota allocations and historical resource usage (intended and current system state) to establish a semantic for prioritization.

As described in [?], the intended target and current are represented in trees, a natural map of the organizations that represent. To calculate the distance between the target and the state, priority operators are applied to each corresponding to generate a priority tree. Which later will be manipulated to calculate the final priority.

The priority operates are mathematical functions defined by the following constraints (Extracted from [?]):

1. An operator is a function with two input variables such that:

$$t \in [0,1], s \in [0,1] \Rightarrow F(t,s) \in [-1,1] \tag{1}$$

$$\begin{aligned} F(t,s) = 0 &\iff t = s \\ F(t,s) > 0 &\iff t > s \\ F(t,s) < 0 &\iff t < s \end{aligned} \tag{2}$$

where t represents a target value, s a (normalized) state value and $t = s$ is the ideal balance axis transecting the operator value space.

2. The function is strictly increasing on target and strictly decreasing on state:

$$\begin{aligned} \forall t_j, t_i, s_j, s_i, t, s \in &(0,1], \\ F(t_j, u) > F(t_i, u) &\iff \quad t_j > t_i \\ F(t_j, s_i) > F(t_j, s_j) &\iff \quad s_j < s_i \\ F(t_j, s) = F(t_i, s) &\iff \quad t_j = t_i \\ F(t, s_i) = F(t, s_j) &\iff \quad s_j = s_i \end{aligned} \tag{3}$$

3. Operator functions are idempotent and deterministic.

In this document study one of the properties presented in [**?**]: Redistribute unused resource capacity among users in the same subgroup proportionally to their allocations. In particular we will prove that, under conditions of infinite output resolution, the Relative operator complies with this property.

2 Redistribute unused resource capacity among users in the same subgroup proportionally to their allocations

2.1 Desired behavior

In this hierarchical prioritization system, the target functions that govern the access to the resources are shaped as trees. Any group of child nodes to the same parent node conform a subgroup. Each node in a subgroup receives a share (proportion) of how much of the share of the parent node corresponds to that node. For the state of the system is similar: each node calculates a normalized state in comparison to the total state of all the nodes of the subgroup.

Under ideal circumstances, when all the nodes are applying to use the resource, the subgroup nodes state will converge to the target values defined for each one of them. However, if one or more of the nodes are not using the resources then the system will converge to a different state. Which one? It will depend on the operator. But the desired behavior is that the unused share is divided among the nodes which apply for the resource. Let's look at an example: 3 nodes with shares $\{t_1 = 0.5, t_2 = 0.2, t_3 = 0.3\}$. If all the users use the resources, their state will converge to those values. However, if user 1 stops submitting jobs, two possible targets can be considered (Among others):

1. $\{t'_1 = 0, t'_2 = 0.45, t'_3 = 0.55\}$ where the unused share is divide equally, 0,25 for both.

2. $\{t'_1 = 0, t'_2 = 0.4, t'_3 = 0.6\}$ where the unused share was divided proportionally to their target t_i: 0,2 for 0,2 and 0,3 for 0,3.

The desired behavior is the second one. And this t'_i are the virtual target: the new target to which the system should converge when one or more users are not applying for a resource.

2.2 Terminology

The presented virtual target: a new set of t'_i where the inactive users have a value of 0 and the active ones a new value with a proportional increase. It can be expressed as:

$$
\begin{aligned}
i, \in \mathbb{T}, K = \sum_{i \in \mathbb{T}} t_i \\
t'_i = \frac{t_i}{\sum_{i \in \mathbb{T}} t_i} = \frac{t_i}{K}, K < 1, t'_i \geq t_i
\end{aligned}
\tag{4}
$$

being $\mathbb{T} = \{\text{set of indexes of the users submitting jobs}\}, t_i$ the target of user i and t'_i the virtual target of the user i after adding the proportional part of the unused share.

3 Demonstration for relative operator

3.1 Method

The base of the proof is quite simple:

1. We have to sets of policies $\{t_i\}$ (original target).
2. if an operator receives $\{t_i\}$ as an input and all users apply for the resources, the system will converge to $\{t_i\}$.
3. If some users don't apply for resources, the system should converge to $\{t'_i\}$ (recalculated target). This is the desired behavior.
4. If an operator would receive with $\{t'_i\}$ and all users in that new target apply for the resources, the system will converge to $\{t'_i\}$.
5. Given any two active users and their two original target values (t_i, t_j) and their corresponding virtual ones, (t'_i, t'_j). If we can prove that for any input state, an operator will produce the same ordering when the target is (t_i, t_j) and (t'_i, t'_j), then every individual decision will be the same in both cases and the system will be brought to (t'_i, t'_j).
6. As a consequence, the operator will comply with the property.

We have to sets of policies $\{t_i\}$ (original target), $\{t'_i\}$ (recalculated target). A priority operator which would operate with $\{t'_i\}$ would bring the system to that target.

This can be expressed as:

$$
\begin{aligned}
i, j \in \mathbb{T}, K = \sum_{i \in \mathbb{T}} t_i \\
t'_i = \frac{t_i}{\sum_{i \in \mathbb{T}} t_i} = \frac{t_i}{K}, K < 1, t'_i \geq t_i \\
\forall t'_i, t'_j, s_i, s_j \in [0, 1] \\
1. \quad F(t'_j, s_j) > F(t'_i, s_i) \Rightarrow F(t_j, s_j) > F(t_i, s_i) \\
2. \quad F(t'_j, s_j) = F(t'_i, s_i) \Rightarrow F(t_j, s_j) = F(t_i, s_i)
\end{aligned}
\tag{5}
$$

being $\mathbb{T} = \{\text{set of indexes of the users submitting jobs}\}, t_i$ the target of user i, s_i the normalized state of user i and t'_i the virtual target of the user i after adding the proportional part of the unused share.

If we prove 1 and 2, F will bring the system to the virtual target.

3.2 The relative operator

$$F(p,u) = \begin{cases} \dfrac{t-s}{t} & s < t \\ 0 & s = t \\ -\dfrac{s-t}{u} & s > t \end{cases} \tag{6}$$

3.3 Development

This proof would very easy if the operators would be defined with the same function for $t > s$ and $t < s$, as it just would mean a proportional increase in the output of the operator, keeping the ordering among users. However, it is not the case, and the functions have different behaviors over and under the balance point so it has to be analyzed.

3.3.1 Cases

We have to study the ordering cases for all the possible relationships of t'_i, t'_j, t_i, t_j and s_i, s_j when ordering two users. For each case we will evaluate if they are real (If the relation ship between the values and the output of the operator are possible) and check if the ordering is the same.

Here we can observe all the possible cases.We already purged those which are not possible. For example, under the condition of $F(t'_j, s_j) = F(t'_i, s_i)$, the case of $t'_j > s_j \wedge t'_i < s_i$, cannot happen, as the priority signs have to be the same in both sides, this and other cases have been eliminated. We will test one every remaining case.

$$\begin{array}{cc} 1. & F(t'_j, s_j) > F(t'_i, s_i) \Rightarrow F(t_j, s_j) > F(t_i, s_i) \\ 1.1 & t'_j > s_j \wedge t'_i > s_i \\ 1.2 & t'_j > s_j \wedge t'_i < s_i \\ 1.3 & t'_j < s_j \wedge t'_i < s_i \\ 2. & F(t'_j, s_j) = F(t'_i, s_i) \Rightarrow F(t_j, s_j) = F(t_i, s_i) \\ 2.1 & t'_j > s_j \wedge t'_i > s_i \\ 2.2 & t'_j < s_j \wedge t'_i < s_i \end{array} \tag{7}$$

3.3.2 Case 1.1

In this case we will stay all the possible cases that comply with:

$$\begin{aligned} F(t'_j, s_j) > F(t'_i, s_i) \Rightarrow F(t_j, s_j) > F(t_i, s_i) \\ t'_j > s_j \wedge t'_i > s_i \end{aligned} \tag{8}$$

Meaning that the version of the functions used by the virtual targets will be always the positive one.

However, even if the virtual targets keep that relationship with don't know of the former target did also. That's why we have to evaluate all the possibilities.

$$\begin{array}{ll} 1.1 & t'_j > s_j \wedge t'_i > s_i \\ 1.1.a & t_j > s_j \wedge t_i > s_i \\ 1.1.b & t_j > s_j \wedge t_i < s_i \\ 1.1.c & t_j < s_j \wedge t_i < s_i \\ 1.1.c & t_j < s_j \wedge t_i > s_i \end{array} \tag{9}$$

3.3.3 Case 1.1.a

Implication to prove:

$$F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j > s_j \wedge t_i > s_i \Rightarrow F(t_j, s_j) > F(t_i, s_i) \tag{10}$$

We will start with $F(t'_j, s_j) > F(t'_i, s_i)$ transforming it until we arrive to $F(t_j, s_j) > F(t_i, s_i)$

$$\begin{aligned} F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j > s_j \wedge t_i > s_i \\ F(t'_j, s_j) > F(t'_i, s_i) \\ \frac{t'_j - s_j}{t'_j} > \frac{t'_i - s_i}{t'_i}, 1 - \frac{s_j}{t'_j} > 1 - \frac{s_i}{t'_i} \\ -\frac{s_j \cdot K}{t_j} > -\frac{s_i \cdot K}{t_i}, \frac{s_j}{t_j} < \frac{s_i}{t_i} \\ -\frac{s_j}{t_j} > -\frac{s_i}{t_i}, 1 - \frac{s_j}{t_j} > 1 - \frac{s_i}{t_i}, \frac{t_j - s_j}{t_j} > \frac{t_i - s_i}{t_i} \\ \frac{t_j - s_j}{t_j} > \frac{t_i - s_i}{t_i} \wedge \wedge t_j > s_j \wedge t_i > s_i \Rightarrow \\ F(t_j, s_j) > F(t_i, s_i) \end{aligned} \tag{11}$$

3.3.4 Case 1.1.b

Implication to prove:

$$F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j > s_j \wedge t_i < s_i \Rightarrow F(t_j, s_j) > F(t_i, s_i) \tag{12}$$

This is simpler we just need to evaluate the signs of the Operator with the final target values.

$$\begin{aligned} F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j > s_j \wedge t_i > s_i \\ t_j > s_j \wedge t_i > s_i \Rightarrow F(t_j, s_j) > 0 \wedge F(t_i, s_i) < 0 \Rightarrow F(t_j, s_j) > F(t_i, s_i) \end{aligned} \tag{13}$$

3.3.5 Case 1.1.c

Implication to prove:

$$F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j < s_j \wedge t_i < s_i \Rightarrow F(t_j, s_j) > F(t_i, s_i) \tag{14}$$

We will start with $F(t'_j, s_j) > F(t'_i, s_i)$ transforming it until we arrive to $F(t_j, s_j) > F(t_i, s_i)$. It is more interesting as we will have to arrive to the negative version of the operator.

$$\begin{aligned} F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j < s_j \wedge t_i < s_i \\ F(t'_j, s_j) > F(t'_i, s_i), \\ \frac{t'_j - s_j}{t'_j} > \frac{t'_i - s_i}{t'_i}, 1 - \frac{s_j}{t'_j} > 1 - \frac{s_i}{t'_i} \\ -\frac{s_j \cdot K}{t_j} > -\frac{s_i \cdot K}{t_i}, \frac{s_j}{t_j} < \frac{s_i}{t_i} \\ \frac{t_j}{s_j} > \frac{t_i}{s_i}, \frac{t_j}{s_j} - 1 > \frac{t_i}{s_i} - 1, \frac{t_j - s_j}{s_j} > \frac{t_i - s_j}{s_i} \\ \frac{t_j - s_j}{s_j} > \frac{t_i - s_j}{s_i} \wedge t_j < s_j \wedge t_i < s_i \Rightarrow \\ F(t_j, s_j) > F(t_i, s_i) \end{aligned} \tag{15}$$

3.3.6 Case 1.1.d

Implication to prove:

$$F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j < s_j \wedge t_i > s_i \Rightarrow F(t_j, s_j) > F(t_i, s_i) \tag{16}$$

This case is interesting because is one in which the precondition is not possible. Let's elaborate the left part of the implication.

$$\begin{aligned} F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j < s_j \wedge t_i > s_i \\ \text{From the case 1.1.a we know that:} \\ F(t'_j, s_j) > F(t'_i, s_i) \Rightarrow \frac{s_j}{t_j} < \frac{s_i}{t_i} \\ \text{However: } (t_j < s_j \Rightarrow \frac{s_j}{t_j} > 0 \wedge t_i > s_i \Rightarrow \frac{s_i}{t_i} < 0) \Rightarrow \frac{s_j}{t_j} > \frac{s_i}{t_i} \\ \text{If we then combine all the predicates:} \\ F(t'_j, s_j) > F(t'_i, s_i) \wedge t_j < s_j \wedge t_i > s_i \Rightarrow \frac{s_j}{t_j} < \frac{s_i}{t_i} \wedge \frac{s_j}{t_j} > \frac{s_i}{t_i} \end{aligned} \tag{17}$$

Situation that it is not possible. This means, that with such t', the conditions of $t_j < s_j \wedge t_i > s_i$ are not possible and there is no need to elaborate this case.

3.3.7 Case 1.2

In this case we will stay all the possible cases that comply with:

$$\begin{aligned} F(t'_j, s_j) > F(t'_i, s_i) \Rightarrow F(t_j, s_j) > F(t_i, s_i) \\ t'_j > s_j \wedge t'_i < s_i \end{aligned} \tag{18}$$

Meaning that the version of the functions used by the virtual targets will be always the positive for t'_j and the negative for t'_i.

However, even if the virtual targets keep that relationship with don't know of the former target did also. That's why we have to evaluate all the possibilities.

$$\begin{array}{ll} 1.2.a & t_j > s_j \wedge t_i < s_i \\ 1.2.b & t_j < s_j \wedge t_i < s_i \end{array} \tag{19}$$

3.3.8 Case 1.2.a

Implication to prove:

$$F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i < s_i \wedge t_j > s_j \wedge t_i < s_i \Rightarrow F(t_j, s_j) > F(t_i, s_i) \tag{20}$$

This is a simple case as we need part of the precondition to infer the right part.

$$\begin{aligned} t_j > s_j \Rightarrow F(t_j, s_j) > 0 \wedge t_i < s_i \Rightarrow F(t_i, s_i) < 0 \\ t_j > s_j \wedge t_i < s_i \Rightarrow F(t_j, s_j) > F(t_i, s_i) \end{aligned} \tag{21}$$

3.3.9 Case 1.2.b

Implication to prove:

$$F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i < s_i \wedge t_j < s_j \wedge t_i < s_i \Rightarrow F(t_j, s_j) > F(t_i, s_i) \tag{22}$$

We will start with $F(t'_j, s_j) > F(t'_i, s_i)$ but we will establish a property around K what will lead us to the right side of the implication.

$$\begin{aligned}
F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i < s_i \wedge t_j < s_j \wedge t_i < s_i \\
t'_j > s_j \Rightarrow F(t'_j, s_j) > 0, \frac{t'_j - s_j}{s_j} > 0, \frac{t_j}{s_j \cdot K} - 1 > 0 \\
\frac{t_j}{s_j \cdot K} > 1, \frac{t_j}{s_j} > K \\
t'_i < s_i \Rightarrow F(t'_i, s_i) < 0, \frac{t'_i - s_j}{t'_i} < 0, 1 - \frac{s_i \cdot K}{t_i} < 0, \\
1 < \frac{s_i \cdot K}{t_i}, \frac{t_i}{s_i} < K \\
\text{let's put both together:} \\
\frac{t_j}{s_j} > K \wedge \frac{t_i}{s_i} < K \Rightarrow \frac{t_i}{s_i} < K < \frac{t_j}{s_j} \Rightarrow \frac{t_i}{s_i} < \frac{t_j}{s_j} \\
\text{and as seen in previous equations:} \\
\frac{t_i}{s_i} < \frac{t_j}{s_j} \wedge t_j < s_j \wedge t_i < s_i \Rightarrow \\
F(t_j, s_j) > F(t_i, s_i)
\end{aligned} \tag{23}$$

3.3.10 Case 1.3

In this case we will stay all the possible cases that comply with:

$$\begin{aligned}
F(t'_j, s_j) > F(t'_i, s_i) \Rightarrow F(t_j, s_j) > F(t_i, s_i) \\
t'_j < s_j \wedge t'_i < s_i
\end{aligned} \tag{24}$$

Meaning that the version of the functions used by the virtual targets will be always the negative for both targets. As a consequence, the original targets will be also negative.

However, even if the virtual targets keep that relationship with don't know of the former target did also. That's why we have to evaluate all the possibilities.

$$1.3.a \quad t_j < s_j \wedge t_i < s_i \tag{25}$$

3.3.11 Case 1.3.a

Implication to prove:

$$F(t'_j, s_j) > F(t'_i, s_i) \wedge t'_j < s_j \wedge t'_i < s_i \wedge t_j < s_j \wedge t_i < s_i \Rightarrow F(t_j, s_j) > F(t_i, s_i) \tag{26}$$

We will start with $F(t'_j, s_j) > F(t'_i, s_i)$ transforming it until we arrive to $F(t_j, s_j) > F(t_i, s_i)$. It is more interesting as we will have to arrive to the negative version of the operator.

$$\begin{aligned} F(t'_j,s_j) > F(t'_i,s_i) \wedge t'_j < s_j \wedge t'_i < s_i \wedge t_j < s_j \wedge t_i < s_i \\ F(t'_j,s_j) > F(t'_i,s_i), \\ \frac{t'_j - s_j}{s_j} > \frac{t'_i - s_i}{s_i}, \frac{t'_j}{s_j} - 1 > \frac{t'_i}{s_i} - 1, \\ \frac{t_j}{s_j \cdot K} - 1 > \frac{t_i}{s_i \cdot K} - 1, \frac{t_j}{s_j \cdot K} > \frac{t_i}{s_i \cdot K}, \frac{t_j}{s_j} > \frac{t_i}{s_i} \\ \frac{t_j}{s_j} - 1 > \frac{t_i}{s_i} - 1, \frac{t_j - s_j}{s_j} > \frac{t_i - s_i}{s_i} \\ \frac{t_j - s_j}{s_j} > \frac{t_i - s_i}{s_i} \wedge t_j < s_j \wedge t_i < s_i \Rightarrow F(t_j,s_j) > F(t_i,s_i) \end{aligned} \tag{27}$$

3.3.12 Case 2.1

In this case we will stay all the possible cases that comply with:

$$\begin{aligned} F(t'_j,s_j) = F(t'_i,s_i) \Rightarrow F(t_j,s_j) = F(t_i,s_i) \\ t'_j > s_j \wedge t'_i > s_i \end{aligned} \tag{28}$$

Meaning that the version of the functions used by the virtual targets will be always the positive one.

However, even if the virtual targets keep that relationship with don't know of the former target did also. That's why we have to evaluate all the possibilities.

$$\begin{array}{ll} 2.1 & t'_j > s_j \wedge t'_i > s_i \\ 2.1.a & t_j > s_j \wedge t_i > s_i \\ 2.1.b & t_j > s_j \wedge t_i < s_i \\ 2.1.c & t_j < s_j \wedge t_i < s_i \\ 2.1.c & t_j < s_j \wedge t_i > s_i \end{array} \tag{29}$$

3.3.13 Case 2.1.a

Implication to prove:

$$F(t'_j,s_j) = F(t'_i,s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j > s_j \wedge t_i > s_i \Rightarrow F(t_j,s_j) = F(t_i,s_i) \tag{30}$$

We will start with $F(t'_j, s_j) = F(t'_i, s_i)$ transforming it until we arrive to $F(t_j, s_j) = F(t_i, s_i)$

$$\begin{aligned} F(t'_j, s_j) = F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j > s_j \wedge t_i > s_i \\ F(t'_j, s_j) = F(t'_i, s_i) \\ \frac{t'_j - s_j}{t'_j} = \frac{t'_i - s_i}{t'_i}, 1 - \frac{s_j}{t'_j} = 1 - \frac{s_i}{t'_i} \\ -\frac{s_j \cdot K}{t_j} = -\frac{s_i \cdot K}{t_i}, \frac{s_j}{t_j} = \frac{s_i}{t_i} \\ -\frac{s_j}{t_j} = -\frac{s_i}{t_i}, 1 - \frac{s_j}{t_j} = 1 - \frac{s_i}{t_i}, \frac{t_j - s_j}{t_j} = \frac{t_i - s_i}{t_i} \\ \frac{t_j - s_j}{t_j} = \frac{t_i - s_i}{t_i} \wedge t_j = s_j \wedge t_i = s_i \Rightarrow \\ F(t_j, s_j) = F(t_i, s_i) \end{aligned} \tag{31}$$

3.3.14 Case 2.1.b

Implication to prove:

$$F(t'_j, s_j) = F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j > s_j \wedge t_i < s_i \Rightarrow F(t_j, s_j) = F(t_i, s_i) \tag{32}$$

This case it is not taken into account as the precondition is not possible.

$$\begin{aligned} \text{From 2.1.a we obtain that:} \\ F(t'_j, s_j) = F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \Rightarrow \frac{s_j}{t_j} = \frac{s_i}{t_i} \\ t_j > s_j \wedge \frac{s_j}{t_j} < 1, t_i < s_i \wedge \frac{s_i}{t_i} > 1 \\ < 1 = \quad > 1 \text{ is not possible} \end{aligned} \tag{33}$$

3.3.15 Case 2.1.c

Implication to prove:

$$F(t'_j, s_j) = F(t'_i, s_i) \wedge t'_j > s_j \wedge t'_i > s_i \wedge t_j < s_j \wedge t_i < s_i \Rightarrow F(t_j, s_j) = F(t_i, s_i) \tag{34}$$

We will start with $F(t'_j, s_j) = F(t'_i, s_i)$ transforming it until we arrive to $F(t_j, s_j) = F(t_i, s_i)$. It is more interesting as we will have to arrive to the negative version of the operator.

$$\begin{aligned}F(t'_j,s_j)=F(t'_i,s_i)\wedge t'_j>s_j\wedge t'_i>s_i\wedge t_j<s_j\wedge t_i<s_i\\ F(t'_j,s_j)=F(t'_i,s_i),\\ \frac{t'_j-s_j}{t'_j}=\frac{t'_i-s_i}{t'_i},1-\frac{s_j}{t'_j}=1-\frac{s_i}{t'_i}\\ -\frac{s_j\cdot K}{t_j}=-\frac{s_i\cdot K}{t_i},\frac{s_j}{t_j}=\frac{s_i}{t_i}\\ \frac{t_j}{s_j}=\frac{t_i}{s_i},\frac{t_j}{s_j}-1=\frac{t_i}{s_i}-1,\frac{t_j-s_j}{s_j}=\frac{t_i-s_j}{s_i}\\ \frac{t_j-s_j}{s_j}=\frac{t_i-s_j}{s_i}\wedge t_j<s_j\wedge t_i<s_i\Rightarrow\\ F(t_j,s_j)=F(t_i,s_i)\end{aligned}\tag{35}$$

3.3.16 Case 2.1.d

Implication to prove:

$$F(t'_j,s_j)=F(t'_i,s_i)\wedge t'_j>s_j\wedge t'_i>s_i\wedge t_j<s_j\wedge t_i>s_i\Rightarrow F(t_j,s_j)=F(t_i,s_i)\tag{36}$$

This case it is not taken into account as the precondition is not possible.

$$\begin{aligned}\text{From 2.1.a we obtain that:}\\ F(t'_j,s_j)=F(t'_i,s_i)\wedge t'_j>s_j\wedge t'_i>s_i\Rightarrow\frac{s_j}{t_j}=\frac{s_i}{t_i}\\ t_j<s_j\wedge\frac{s_j}{t_j}>1,t_i>s_i\wedge\frac{s_i}{t_i}<1\\ >1=\ <1\text{ is not possible}\end{aligned}\tag{37}$$

3.3.17 Case 2.2

In this case we will stay all the possible cases that comply with:

$$\begin{aligned}F(t'_j,s_j)=F(t'_i,s_i)\Rightarrow F(t_j,s_j)=F(t_i,s_i)\\ t'_j<s_j\wedge t'_i<s_i\end{aligned}\tag{38}$$

Meaning that the version of the functions used by the virtual targets will be always the negative for both targets. As a consequence, the original targets will be also negative.

However, even if the virtual targets keep that relationship with don't know of the former target did also. That's why we have to evaluate all the possibilities.

3.3.18 Case 2.2.a

Implication to prove:

$$F(t'_j,s_j)=F(t'_i,s_i)\wedge t'_j<s_j\wedge t'_i<s_i\wedge t_j<s_j\wedge t_i<s_i\Rightarrow F(t_j,s_j)=F(t_i,s_i)\tag{39}$$

We will start with $F(t'_j, s_j) = F(t'_i, s_i)$ transforming it until we arrive to $F(t_j, s_j) = F(t_i, s_i)$. It is more interesting as we will have to arrive to the negative version of the operator.

$$\begin{aligned}
F(t'_j, s_j) = F(t'_i, s_i) \wedge t'_j < s_j \wedge t'_i < s_i \wedge t_j < s_j \wedge t_i < s_i \\
F(t'_j, s_j) = F(t'_i, s_i), \\
\frac{t'_j - s_j}{s_j} = \frac{t'_i - s_i}{s_i}, \frac{t'_j}{s_j} - 1 = \frac{t'_i}{s_i} - 1, \\
\frac{t_j}{s_j \cdot K} - 1 = \frac{t_i}{s_i \cdot K} - 1, \frac{t_j}{s_j \cdot K} = \frac{t_i}{s_i \cdot K}, \frac{t_j}{s_j} = \frac{t_i}{s_i} \\
\frac{t_j}{s_j} - 1 = \frac{t_i}{s_i} - 1, \frac{t_j - s_j}{s_j} = \frac{t_i - s_i}{s_i} \\
\frac{t_j - s_j}{s_j} = \frac{t_i - s_i}{s_i} \wedge t_j < s_j \wedge t_i < s_i \Rightarrow F(t_j, s_j) = F(t_i, s_i)
\end{aligned} \tag{40}$$

3.4 Finalization of the Proof

At this point all the cases have been checked, confirming that in all cases the property enunciated in this document holds for the Relative operator, so it proportionally distributes unused share in an ordering fairshare system

4 Extending these results to other operators

This is an ordering property of the priority operators. According to the equivalence property enunciated in [**?**], al the equivalent operators also comply with the property of proportional distribution of unused share in an ordering fairshare system.

5 Not applicable to

This demonstration is only applicable to the ordering cases in fairhsare, it doesn't apply to situations in which the magnitudes of the fairshare values are relevant by themselves.

References

[1] Östberg, P-O. and Espling, D., Elmroth, E.: Decentralized scalable fairshare scheduling. Future Generation Computer Systems - The International Journal of Grid Computing and eScience 29 (2013) 130–143

[2] Rodrigo, G.P., Östberg, P-O.and Elmroth, E.: Priority operators for fairshare scheduling. Not yet publisehd (2014)