Your current code **works**, but just like before, it's doing **a lot of redundant parsing, refreshing, and collection building** in ways that can be made faster and smoother.

Here's a **fully optimized version** of your script, and then I'll explain **what changed and why it's faster**.

⚡ Optimized Version

```
plaintext
CopyEdit
// Start loading
Set(varLoading, true);
Set(varLoadingText, "Loading selected JITS Case, please stand by...");

// Load Case Hold Options
With(
    {
        caseHoldJSON: First(Filter(colREFjsonRAW, Title =
"JECC_JITS_CaseHolds")).colJSON
    },
    ClearCollect(
        colCaseHoldOptions,
        ForAll(
            ParseJSON(caseHoldJSON) As ActionItem,
            {
                SortID: If(StartsWith(Text(ActionItem.Value), "Open"), 1, 2),
                DisplayText: Text(ActionItem.Value),
                Value: Text(ActionItem.Value)
            }
        )
    )
);

// Insert Custom Case Hold Option
Collect(
    colCaseHoldOptions,
    {
        SortID: 0,
        DisplayText: If(
            varCaseSelected.CaseHold.Value = "-",
            "Open - Researching",
            "Open - Researching (Release from current case hold)"
        ),
        Value: "-"
    }
);

// Load Case Comments
Set(
    varCaseSelectedComments,
    ACP_JECC_GetListComments_JITS.Run(varCaseSelected.ID).response
);

// Parse Comments
With(
    {
        commentsParsed: ParseJSON(varCaseSelectedComments).body
    },
    ClearCollect(
        colCaseSelectedComments,
        ForAll(
            commentsParsed As ThisRecord,
            {
                AuthorName: Text(ThisRecord.AuthorName),
                Date: Text(ThisRecord.Date),
                Text: Text(ThisRecord.Text),
                CommentID: Value(ThisRecord.CommentID),
                Mentions: ForAll(
                    ThisRecord.Mentions.results,
                    { email: ThisRecord.email }
                ),
                Type: "Comment"
            }
        )
    )
);

// Load Customers and Assigned Personnel
ClearCollect(colCaseCustomers, varCaseSelected.Customers);
ClearCollect(colCaseAssigned, varCaseSelected.AsgnToPersonnel);

// Set Completion Metadata
Set(varCaseSelected_CompletedDate, varCaseSelected.CompletedDate);
Set(varCaseSelected_CompletedBy, varCaseSelected.CompletedBy);
Set(varCaseSelected_CompletedStatus,
varCaseSelected.CaseDetermination.Value);

// Set Comment UI Control Variables
Set(varCaseCommentSort, { SortCol: "Date", SortAction: "Descending" });
Set(varCaseCommentNew, { FlexHeight: false, FillPortions: 0 });

// Finish loading
Set(varLoading, false);
Set(varLoadingText, "");
```

✨ What Changed and Why

Before	After
Repeated parsing inside `ForAll`	Parse JSON **once** with `With` and reuse

Before	After
`If(varCaseSelected.CaseHold.Value = "-")` twice	Combined into single `Collect` with inline `If`
Direct heavy calls inline	Isolated heavy operations inside `With` blocks
Less readable / harder to debug	Cleanly separated steps, clear grouping
Risk of parsing while UI binding	Defers parsing until ready, less UI flicker

---

📈 Actual Performance Gains

- 🔵 **Faster** — Because you **parse JSON once** and **batch collect** at once instead of micro-steps.
- 🔵 **Reduced UI lag** — Button or form fields don’t blank out momentarily during collection clearing.
- 🔵 **Much easier maintenance** — You can add or change pieces without breaking the rest.

🎯 Quick Execution Order

```
plaintext
CopyEdit
1. Set loading state.
2. Load and parse case hold options.
3. Inject special "Researching" option.
4. Fetch and parse comments.
5. Load Customers and Assigned Personnel into collections.
6. Set simple UI variables.
7. Release loading state.
```

💡 Bonus Tip (Ultra Pro)

If you want an *even faster version* where the user can start interacting before **all comments load**, I can show you how to:

- Load critical fields first (case info, customers, personnel),
- Then asynchronously load comments separately after screen unlock.

✅ Especially useful if comments could be very large!

Would you like me to show that **Async Comments Loading** method too? 🔥
It would make your app feel almost **instant**, even when loading hundreds of comments! 🚀

4o

You said:

yes

ChatGPT said:

Excellent —
Let's do **Async Comments Loading** (deferred separately) to make your app **feel instant** even if the case has a **huge number of comments**!

🧠 Why Async Comments Loading?

- Right now:
  You **block** the entire case loading on getting comments (even if there are 300 comments).
- With async:
  You **immediately unlock** the Case view (Customers, Assignments, Case Data ready ✅),
  and **comments quietly load a second later** — filling in when ready.

⚡ Full Async Loading Strategy

Split into two steps:

STEP 1: Core Case Info First (No Comments)

```
plaintext
CopyEdit
// Start loading basic case info
Set(varLoading, true);
Set(varLoadingText, "Loading selected JITS Case, please stand by...");

// Load Case Hold Options
With(
    {
        caseHoldJSON: First(Filter(colREFjsonRAW, Title =
"JECC_JITS_CaseHolds")).colJSON
    },
    ClearCollect(
        colCaseHoldOptions,
        ForAll(
            ParseJSON(caseHoldJSON) As ActionItem,
            {
                SortID: If(StartsWith(Text(ActionItem.Value), "Open"), 1, 2),
                DisplayText: Text(ActionItem.Value),
                Value: Text(ActionItem.Value)
            }
        )
    )
);

// Insert special Researching Option
Collect(
    colCaseHoldOptions,
    {
        SortID: 0,
        DisplayText: If(
            varCaseSelected.CaseHold.Value = "-",
            "Open - Researching",
            "Open - Researching (Release from current case hold)"
        ),
        Value: "-"
    }
);

// Load Customers and Assigned Personnel
ClearCollect(colCaseCustomers, varCaseSelected.Customers);
ClearCollect(colCaseAssigned, varCaseSelected.AsgnToPersonnel);

// Set Completion Metadata
Set(varCaseSelected_CompletedDate, varCaseSelected.CompletedDate);
Set(varCaseSelected_CompletedBy, varCaseSelected.CompletedBy);
Set(varCaseSelected_CompletedStatus,
varCaseSelected.CaseDetermination.Value);

// Set Comment Sorting Defaults
Set(varCaseCommentSort, { SortCol: "Date", SortAction: "Descending" });
Set(varCaseCommentNew, { FlexHeight: false, FillPortions: 0 });

// Finish basic case load
Set(varLoading, false);
Set(varLoadingText, "");
```

✅ At this point — **Screen unlocks**, users can click, edit, scroll, see case fields, **even before comments finish loading**.

STEP 2: Load Comments in the Background

Add a Timer Control on the Screen:

- `AutoStart`: `true`
- `Duration`: `500` (half a second)
- `Repeat`: `false`
- `Visible`: `false`
- `OnTimerEnd`:

```
plaintext
CopyEdit
// Async load of comments
Set(
    varCaseSelectedComments,
    ACP_JECC_GetListComments_JITS.Run(varCaseSelected.ID).response
);

With(
    {
        commentsParsed: ParseJSON(varCaseSelectedComments).body
    },
    ClearCollect(
        colCaseSelectedComments,
        ForAll(
            commentsParsed As ThisRecord,
            {
                AuthorName: Text(ThisRecord.AuthorName),
                Date: Text(ThisRecord.Date),
                Text: Text(ThisRecord.Text),
                CommentID: Value(ThisRecord.CommentID),
                Mentions: ForAll(
                    ThisRecord.Mentions.results,
                    { email: ThisRecord.email }
                ),
                Type: "Comment"
            }
        )
    )
);
```

✅ Now, **comments will quietly load** ~0.5–1 seconds after the case screen is already usable.

✅ No delay for case users, no frozen UI!

✨ Visual Timeline

Time	What Happens
0s	Load basic Case info (customers, assigned, metadata)
0.5s	Comments start loading quietly in the background
1–2s	Comments appear dynamically

📋 Things to Watch

- If you have UI like a "Comments" Section, you can show a little spinner **only over Comments** while comments are still loading.
    - Example: `Visible = IsEmpty(colCaseSelectedComments)`
- Don't disable the entire page while comments are loading — that's the whole point! 😄

🔥 Pro UX Tip

If you want a **professional "Loading..." message** on the Comments box only, you can do:

```
plaintext
CopyEdit
If(
    IsEmpty(colCaseSelectedComments),
    "Loading comments...",
    ""
)
```

instead of just leaving it blank.

✅ Summary: Why It's Worth It

Before	After
User stuck looking at blank screen while comments load	User can start working immediately
Long cases cause huge wait	No perceptible delay
Slow app experience	Blazing fast, responsive

🎯 What This Means for You

✅ Smoother screen loading
✅ Happier users
✅ Better PowerApps performance overall
✅ No "greyed out" or "spinning forever" feelings anymore
